In-situ Soil Moisture Sensing: Measurement Scheduling and Estimation Using Sparse Sampling

XIAOPEI WU, Tsinghua University, The University of Michigan
QINGSI WANG, The University of Michigan
MINGYAN LIU, The University of Michigan

We consider the problem of monitoring soil moisture evolution using a wireless network of in-situ underground sensors. To reduce cost and prolong lifetime, it is highly desirable to rely on fewer measurements and estimate with higher accuracy the original signal (the temporal evolution of soil moisture). In this paper we explore the use of results from the theory of sparse sampling, including compressive sensing (CS) and matrix completion (MC), in this application context. We first consider the problem of reconstructing the soil moisture process at a single location using CS. Our physical constraint leads to very sparse measurement matrices, which makes finding a suitable representation basis very challenging: it needs to sufficient sparsify the underlying signal while at the same time be sufficiently incoherent with the measurement matrix, two common pre-conditions for CS techniques to work well. We construct a representation basis by exploiting unique features of soil moisture evolution, and show that this basis attains very good tradeoff between its ability to sparsify the signal and its incoherence with measurement matrices that are consistent with our physical constraints. We next consider the problem of jointly reconstructing soil moisture processes at multiple locations, assuming sparse measurements can be taken at each location. We show that the spatial soil moisture process enjoys the low-rank property, a prior for MC. Accordingly we introduce a spatio-temporal measurement matrix and apply the MC framework to reconstruct the soil moisture field. Extensive numerical evaluation is performed on both real, high-resolution soil moisture data and simulated data, and through comparison with a closed-loop scheduling approach. Our results demonstrate that for a single location, a uniform measurement scheduling followed by CS recovery results in a very nice tradeoff between estimation accuracy, sampling rate, flexibility and feasibility in implementation. When multiple locations are available, our results show that joint reconstruction using MC in general produces better estimation accuracy than using a single location alone, but it requires the use of independent and random measurement schedules across locations. We also show that these sparse sampling techniques can be augmented so as to be robust against sporadic data outliers/corruption caused by, e.g., intermittent sensor faults.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms: Design, Algorithms, Performance

ACM Reference Format:
Wu, X., Wang, Q., Liu, M., 2014. In-situ Soil Moisture Sensing: Measurement Scheduling and Estimation using Sparse Sampling. *ACM Trans. Sensor Netw.* X, X, Article 1 (May 2014), 29 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

This paper studies the efficient measurement scheduling and sensing of soil moisture. Soil moisture is a critical data type and measurement need in many scientific applications. For instance, it is used in all land surface models, water and energy balance

Author's addresses: Wu, X. , School of Software, Tsinghua University. Wang, Q. and Liu, M., Electrical Engineering and Computer Science, University of Michigan.

DOI: http://dx.doi.org/10.1145/0000000.0000000

models, weather prediction models, general circulation models, and ecosystem process simulation models [NASASP 2006]. It is also a key measurement need in precision farming and agricultural drought monitoring.

Soil moisture data has traditionally been collected using remote sensing techniques like radars and radiometers onboard satellites. Remote sensing covers large areas, but produces very coarse grained measurements, on the order of square kilometers. It was not until recently, with the advances in integrated wireless communication, sensing and processing technology, has in-situ sensing become a feasible option [Akyildiz et al. 2002; Ramanathan and Recht 2009]. In-situ moisture sensors can be densely deployed over a region of interest, at a resolution of one in every few square-feet, and thus can produce much finer grained measurements. To collect desired data at a single location, soil moisture sensor probes are typically placed vertically under the ground at different depths, up to 2 meters deep, with wires connecting them to a ground actuation and wireless transceiver module, see e.g., the SoilSCAPE project [Moghaddam et al. 2010b][1]. This wireless node actuates the moisture probes to take measurements and transfers the collected data wirelessly to a remote central location or base station for processing; an example of such a network is described in more detail in [Moghaddam et al. 2010b].

To gather sufficient information on the temporal and spatial variations and characteristics of soil moisture, it is highly desirable to deploy moisture probes (and the associated wireless nodes) at sufficiently high density, and to take measurements at sufficiently high frequency. A competing objective is to have the network function in a unmanned fashion for as long as possible, since such networks are typically deployed in an open (sometimes remote) field without immediate access to power or human intervention. This requires us to reduce the working times of the wireless nodes to conserve energy, even when renewable sources are used.

For a single wireless node, these competing interests imply that we need to make judicious decisions in measurement scheduling, i.e., when is the best time to take a measurement, so as to minimize the total amount of time the node needs to be active in actuating the moisture probes and in data transmission, while still satisfying the monitoring objective, i.e., achieving a desired level of accuracy in the estimated soil moisture evolution using the measurement data collected. In this paper we examine how compressive sensing and sparse sampling theory may be used to achieve these goals. Note that such measurement scheduling in general runs parallel to other energy efficient methods one may wish to adopt, including MAC[Ji et al. 2014] and routing[Wang et al. 2011]. It can also be jointly designed with a node's sleep schedules.

This problem belongs to the larger class of sensor scheduling problems. There are two general approaches. The first is a model-based approach that makes a measurement decision based on the knowledge of prior statistics of the underlying random process to be monitored, gained either through assumption or training, and is also sometimes referred to as the Bayesian approach; examples include [Athans 1972; Baras and Bensoussan 1989; Andersland and Teneketzis 1996; Evans and Krishnamurthy 2001; Krishnamurthy 2001; Deshpande et al. 2004; Li et al. 2009; Krause et al. 2008; Shuman et al. 2010], with [Shuman et al. 2010] focusing specifically on monitoring soil moisture.

The second is a model-free approach whereby measurement decisions do not rely on prior statistical models and are possibly also open-loop, i.e. independent of past observations and decisions. Sparse sampling, including compressive sensing and matrix

[1]Similar and alternative instrumentations have been used in other studies such as the Suelo project [Ramanathan and Recht 2009], which targets the monitoring of soil which includes but is not specifically designed for moisture data collection.

completion based measurement falls under this category. Recent advances in compressive sensing (CS) theory [Candés et al. 2006; Donoho 2006; Candés and Tao 2006; Jafarpour et al. 2009] allow one to represent compressible/sparse[2] signals with significantly fewer samples than required by the Nyquist sampling theorem. It is therefore particularly attractive in a resource constrained setting like ours. This technique has been used in data compression [Baron et al. 2006], channel coding [Candés and Tao 2005], analog signal sensing [Tian and Giannakis 2007], routing [Quer et al. 2009], sensor placement [Wu et al. 2011] and data collection [Luo et al. 2009], with varying degrees of success. To apply CS in our context, the idea would be to sample the soil moisture in time in some fashion (typically randomly) and use compressive sensing techniques to reconstruct or recover the entire process. Matrix completion (MC) extends CS to the processing of a two-dimensional signal (i.e. a matrix), and has similarly received much attention with applications to data gathering [Cheng et al. 2013], traffic estimation [Cheng et al. 2012], among others. In our context, MC is applicable when we have measurement data from multiple locations and we wish to reconstruct the soil moisture processes jointly at these locations. To apply MC, the moisture processes are organized into a matrix, each row representing the temporal process of a location and sparsely sampled; the estimation typically exploits potential spatial correlation across different locations.

While it is very tempting to use sparse sampling methods like those mentioned above in soil moisture sensing due to their potential to significantly reduce the number of measurements needed, doing so is not without challenges. For instance, there are two major issues in applying CS techniques to our problem. (1) It is not immediately clear how to find a good representation basis (Ψ) under which the soil moisture process may be sparsely represented. There is no systematic way of selecting such a matrix; it is usually done through trials and experience. (2) Under the CS framework the measurement scheduling is specified by a measurement matrix (Φ) (see e.g., the commonly used Gaussian matrix), which is often required to be dense, i.e., each measurement corresponds to a linear combination of multiple samples. However, in our problem the physical nature of the monitoring device is such that each measurement corresponds to one and only one sample of the underlying physical process. This makes our measurement matrix extremely sparse, a feature that often does not bode well for its success, see e.g., [Quer et al. 2009]. An additional consequence of a sparse measurement matrix Φ is that it becomes hard to make it sufficiently incoherent from Ψ, a typical condition required for the success of CS techniques. Similarly, there are two difficulties in using MC in our problem. (1) Given a field with spatially distributed sensors, it may be hard to determine whether the resulting signal matrix has the low-rank property, a typical condition for MC. This can be verified offline when training data is available, but a challenge otherwise. (2) Literature [Candés and Recht 2009] shows that good signal recovery quality is achieved (optimality guaranteed) if measurements are taken randomly. In practice however, random measurements leads to random sleep scheduling of the sensors which in general adds to the complexity in coordinating communication and is not preferred in terms of energy efficiency.

In this paper we consider both CS and MC techniques and examine their applicability and performance in soil moisture sensing, for the case when only a single location is involved and when multiple locations are used, respectively. Our main results and contributions are summarized as follows. (1) We identify a representation basis Ψ, under which the soil moisture process at a single location can be nearly sparsely represented. (2) This matrix is shown to work extremely well with a number of measurement ma-

[2]A signal is compressible/sparse if its coefficients in some fixed basis (e.g., wavelet, Fourier) have relatively few nonzero entries.

trices that are consistent with our physical constraint, including those induced by random and uniform scheduling methods. (3) For soil moisture processes across multiple locations, we show that the resulting signal matrix has a low-rank property when the node density is sufficiently high. This leads to very good recovery accuracy when processes at multiple locations are jointly reconstructed. (4) We compare the CS-based approach to a closed-loop approach to further examine its performance; this comparison also sheds light on the inherent limit of the latter.

All our methods are tested using two sets of data: a real soil moisture data set collected from a botanical garden, and a simulated data set calibrated using data collected from a farm. We note that the methodology presented in this paper is more generally applicable beyond these two data sets. Firstly, this method works well on data from other soil types because it exploits the dynamics driven by rainfall events, which is common across all soil types. Secondly, the combination of representation basis we proposed and the recovery algorithm used can potentially work well with other signal types that are relatively smooth in nature as we note later in Section 4.2. This however must be borne out by further experiments which is out of the scope of the current paper but may be pursued in future studies. Finally, we believe the general methodology followed in this paper, i.e., the selection of a sampling method, a representation basis and a recovery algorithm, as well as verifying the sparsity and incoherence, is applicable to a broad range of similar studies that may wish to employ compressive sensing techniques. While the specific selections of these elements may vary from application to application, and is typically done through experience and trial-and-error, this study nevertheless represents as systematic as possible a scientific method that may be used by other studies.

The remainder of this paper is organized as follows. In Section 2 we describe the problem as well as the data sets we use for our study. Section 3 gives a brief overview of the CS and MC literature most related to the problem under consideration. In Sections 4 and 5 we present the design of the matrices in order to apply CS and MC techniques, respectively. Numerical results are presented in Section 6. We give the solutions on how to address the corrupted measurements in Section 7. Comparison with a non-CS (closed-loop) approach is discussed in Section 8 and Section 9 concludes the paper.

2. THE SCHEDULING PROBLEM AND SOIL MOISTURE DATA

Consider a field of interest with L locations at which soil moisture sensors are deployed; these locations are denoted by $V = \{v_1, \cdots, v_L\}$. For any location $v_i \in V$, typically multiple soil moisture probes are placed vertically, up to 2 meters deep underground, at a single lateral location [Wu and Liu 2012]. While these probes can be activated separately, from an energy management point of view it is far more efficient to activate them all at the same time (i.e., to have them follow the same measurement schedule). This is because the processor (on the ground wireless node) needs to be on (in wake mode) in order to activate any probe, and once it is on it takes very little extra energy to activate an additional probe. For this reason we will treat a any single location as having a single measurement schedule.

We will assume that the underlying moisture process associated with any location is discrete in time. This is obviously not true, but a sufficiently good representation of reality if the time unit is small enough with respect to the time scale of change in the soil moisture, which as we shall see is not very fast. More importantly, it should be noted that since the measurement device operates in discrete time no matter how high the frequency is, the best "ground truth" data is also inevitably discrete in time as a result. Therefore to adopt a discrete time model allows us to precisely quantify the performance of our method using the best ground truth we have available.

Let $\boldsymbol{x}(i) = [x(i,t), t = 0,1,2,\cdots,N]^\top$ be a column vector that denotes the temporal soil moisture process at location v_i and $X = \left[\boldsymbol{x}(1)^\top \ \cdots \ \boldsymbol{x}(L)^\top\right]^\top$ be the matrix representing the collection of processes at all L locations, where we use the superscript $^\top$ to denote the transpose. Let $\pi = \{\pi_1, \cdots, \pi_L\}$ be the *measurement policy*, where $\pi_i = \{t_1, t_2, \cdots, t_n\}$, $t_1 < t_2 < \cdots < t_n$ and $t_i \in \{1,2,\cdots,N\}$, specifies n sampling/measurement times at location v_i. Assuming perfect measurements (no error or noise[3]), policy π results in $L \times n$ measurements across all locations, and the ensemble is denoted by the matrix X^π whose i-th row is given by $(\boldsymbol{x}(i)^{\pi_i})^\top = [x(i,t_1) \ x(i,t_2) \ \cdots \ x(i,t_n)]$. Our goal is to reconstruct or estimate the original signal X from the set of samples X^π as accurately as possible with $n \ll T$, and we denote the reconstruction or estimation policy (or algorithm interchangably) by $\Gamma(\cdot)$ with $\Gamma(X^\pi)$ being reconstructed signal.

We consider two cases. In the first case the estimation is done for each location independently and $\Gamma(X^\pi) = \{\gamma_i(\boldsymbol{x}(i)^{\pi_i})\}_{i=1,\cdots,L}$, i.e., the estimation is performed separately for each location using only samples from that location, with $\gamma_i(\cdot)$ being the estimation policy for location v_i. The objective is to select the best measurement and estimation policies so as to minimize the field estimation error subject to a constraint on the average sampling rate:

$$\begin{aligned} \underset{\pi,\gamma}{\text{minimize}} \quad & \sum_{i=1}^{L} \mathrm{Err}(\boldsymbol{x}(i), \gamma_i(\boldsymbol{x}(i)^{\pi_i})) \\ \text{subject to} \quad & n/N \leq \alpha \end{aligned}$$

where $\mathrm{Err}(\cdot,\cdot)$ is an error function, e.g., the mean-squared error, and α is the requirement on sampling or measurement rate. This case is particularly applicable when there is only a single location of interest ($L = 1$), or when multiple locations are deemed to have little correlation. We see that in this case the problem reduces to the optimal estimation at a single location, assuming a per-location sampling rate requirement. For this problem we will use compressive sensing (CS) techniques (Section 4).

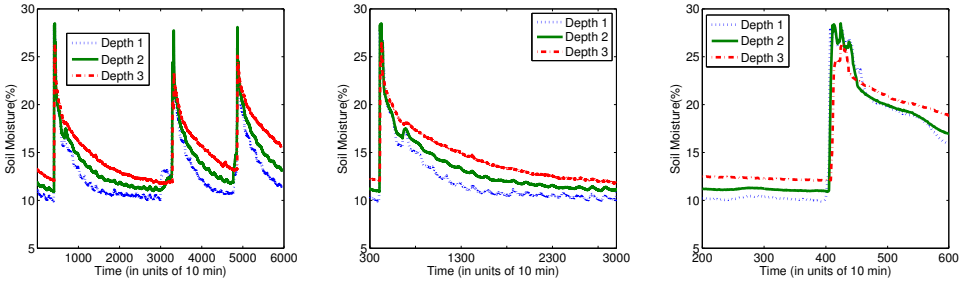
In the second, more general case the policy $\Gamma(.)$ acts on observations from L locations and *jointly* estimates all L processes. The goal is again to seek the optimal measurement schedule that minimizes the total error subject to a maximum sampling rate.

$$\begin{aligned} \underset{\pi,\Gamma}{\text{minimize}} \quad & \mathrm{Err}(X, \Gamma(X^\pi)) \\ \text{subject to} \quad & n/N \leq \alpha \end{aligned}$$

where $\mathrm{Err}(\cdot,\cdot)$ is also an error metric measuring the distance between the real field data X and the estimates $\Gamma(X^\pi)$. This case is applicable when there are multiple locations with possibly correlated observations across locations. For this case we will use matrix completion (MC) techniques (Section 5).

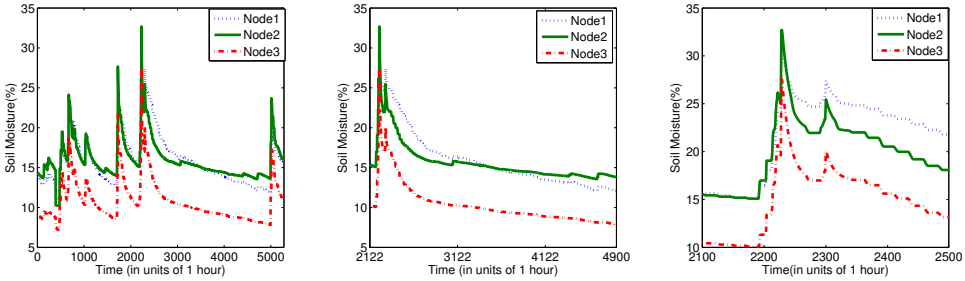
We next discuss the moisture process $\boldsymbol{x}$ used in our study. As mentioned, we will use two data sets. The first one, also referred to as the *Garden data*, was collected at the Matthaei Botanical Garden at the University of Michigan, Ann Arbor ((latitude, longitude) approximately (42.300437, -83.663442)), over a 2-month period between August and October 2009. Three moisture probes were buried at depths 25mm, 67mm and 123mm from the surface, respectively, and took measurements at the rate of once every 10 minutes. This is shown in Figure 1 in a progression of three figures, each with increasing resolution to show both a global and local view of the variation in the process.

[3]We relax this assumption in Section 7.



(a) Entire trace over time. (b) Variation before/after a rainfall. (c) Close-up view of a rainfall event.

Fig. 1. Real soil moisture evolution collected from a botanical garden.



(a) Entire trace over time. (b) Variation before/after a rainfall. (c) Close-up view of a rainfall event.

Fig. 2. Simulated soil moisture evolution on a farm in Oklahoma.

A second data set, also referred to as the *Farm data*, is a simulated one in an environment consistent with the climate and topography of Canton, Oklahoma ((latitude, longitude) approximately (36.00063,-98.63319)), over a 6-month period, between August 2010 and March 2011, at the sampling rate of one measurement per hour. This data is generated by a land surface hydrology simulation MOBIDIC [Castillo 2010], and has been calibrated using location- and time-specific variable exogenous forcings (e.g., rainfall, temperature, cloud cover, and solar radiation) and landscape parameters (e.g., vegetation cover, soil type, and topography), as well as actual data collected at this location. The Farm data contains one depth per location; traces of 3 locations are shown in Figure 2, in a similar progression with increasing resolution.

The reason for using two different sets of data is to provide some diversity in the type of soil moisture processes we use for evaluation. We see that the soil moisture peaks shortly after a rainfall event, with the corresponding moisture level primarily determined by the precipitation. The moisture then slowly dissipates, following a roughly monotonic non-increasing pattern. Over a period of dry weather, the soil moisture level stays relatively constant, resulting in a piece-wise smooth curve between two successive rainfall events. Except for the up-shoot at the onset of a rainfall, the moisture variation exhibits fairly high temporal correlation.

3. COMPRESSIVE SENSING AND MATRIX COMPLETION

In this section we briefly summarize the part of the compressive sensing (CS) and Matrix Completion (MC) literature most relevant to the study presented in this paper.

3.1. Compressive Sensing

Consider a discrete signal given by the vector $\boldsymbol{x}$ of size N. Results in compressive sensing [Candés 2006] have shown that if $\boldsymbol{x}$ is sparse, i.e., if $\|\boldsymbol{x}\|_0 \ll N$ (where the ℓ_0 "norm" $\|\cdot\|_0$ is the number of nonzero elements of $\boldsymbol{x}$), then it is possible to reconstruct $\boldsymbol{x}$ from M random samples produced by a suitably chosen linear transform Φ of $\boldsymbol{x}$: $\boldsymbol{y} = \Phi\boldsymbol{x}$, where $M < N$. The $M \times N$ matrix Φ is usually referred to as the *measurement matrix*. In other words, we can recover signal $\boldsymbol{x}$ from the measurement or observation $\boldsymbol{y}$ if $\boldsymbol{x}$ is sufficiently sparse, subject to some pre-conditions on Φ (more discussed below). In practice, $\boldsymbol{x}$ is usually non-sparse; however, it can often be sparsely represented in an alternative domain. Specifically, $\boldsymbol{x}$ may be further written as $\boldsymbol{x} = \Psi\boldsymbol{s}$, for some $N \times N$ matrix Ψ, where $\boldsymbol{s}$ is the $N \times 1$ coefficient vector in the Ψ domain with $\|\boldsymbol{s}\|_0 = K$, where $K \ll N$. The matrix Ψ will be referred to as the *representation basis*. The measurement vector $\boldsymbol{y}$ can thus be written as

$$\boldsymbol{y} = \Phi\Psi\boldsymbol{s}, \tag{1}$$

and the associated signal recovery problem is to determine $\boldsymbol{s}$ for given measurement $\boldsymbol{y}$ and known matrices Φ and Ψ. The estimate of the original signal is then given by $\boldsymbol{x} = \Psi\boldsymbol{s}$. Clearly, (1) is an under-determined linear system, as the number of equations M is much smaller than the number of variables N (i.e., number of entries of $\boldsymbol{s}$). Finding the solution to this ill-conditioned system has been the subject of extensive studies in recent years.

There are in general the following classes of approaches. The first class seeks $\boldsymbol{s}$ with the smallest ℓ_0 norm:

$$\min_{\boldsymbol{s}\in\mathbb{R}^N} \|\boldsymbol{s}\|_0 \quad \text{s.t.} \quad \boldsymbol{y} = \Phi\Psi\boldsymbol{s}. \tag{2}$$

Directly solving the above is intractable [Donoho et al. 2006; Candés and Tao 2005], but fast approach exists by using smoothed ℓ_0 norm, see e.g. the SL0 method proposed in [Mohimani et al. 2009]. A second class of approaches bypasses the original l_o minimization problem and instead seeks to solve the ℓ_1 norm minimization problem to reduce complexity, also known as Basis Pursuit (BP), see e.g., [Donoho 2004; l1-magic 2005]:

$$\min_{\boldsymbol{s}\in\mathbb{R}^N} \|\boldsymbol{s}\|_1 \quad \text{s.t.} \quad \boldsymbol{y} = \Phi\Psi\boldsymbol{s}, \tag{3}$$

which can be easily solved using linear programming (LP). The justification for solving (3) is that for large systems of equations, the solution to either minimization is the same [Donoho 2004]. Algorithms exist to solve the above problem in polynomial time, including interior-point methods; there are also faster algorithms aimed at large-scale systems, see e.g., [Gorodnitsky and Rao 1997; Tropp and Gilbert 2007; Needell and Vershynin 2010; Mohimani et al. 2009]. In addition to LP, the algorithms we will examine include Iterative Re-weighted Least Squares (IRWLS) [Gorodnitsky and Rao 1997], and Matching Pursuit (MP), see e.g. OMP [Tropp and Gilbert 2007] and ROMP [Needell and Vershynin 2010]. They are considered faster than LP but with worse estimation quality, especially if the signal is not sufficiently sparse.

Using any of the above mentioned recovery algorithms, a K-sparse signal can be reconstructed from M measurements with high probability if M is such that:

$$M \geq C\mu^2(\Phi, \Psi)K \log N, \tag{4}$$

where C is a positive constant, N is the dimension of the signal, and $\mu(\Phi, \Psi)$ is the coherence between the two matrices Φ and Ψ. Given a pair (Φ, Ψ) of orthobases of $\mathbb{R}^N$, $\mu(\Phi, \Psi)$ can be defined as

$$\mu(\Phi, \Psi) = \sqrt{N} \cdot \max_{1\leq i,j\leq N} | < \phi_i, \psi_j > | \in [1, \sqrt{N}]$$

where ϕ_i and ψ_j are row and column vectors of Φ and Ψ, respectively. Thus given Φ and $\boldsymbol{x}$, Ψ must be chosen carefully: it is desirable to represent $\boldsymbol{x}$ in Ψ domain sparsely (small K); at the same time, it is also desirable to have $\mu(\Phi, \Psi)$ as small as possible. The selection of Ψ to meet both criteria is in general non-trivial, especially when Φ is constrained, as we discuss in Section 4.

3.2. Matrix Completion

Consider an $L \times N$ matrix X with a rank r that satisfies $r \ll LN$. It has been shown in [Candés and Recht 2009] that X can be recovered from randomly selected $O(Nr \log N)$ entries. Formally, let Y denote the observations collected:

$$Y = \Phi_{MC} \circ X,$$

where Φ_{MC} is an $L \times N$ *measurement matrix* with $\Phi_{MC}(i, j) = 1$ if the (i, j) entry is selected and 0 otherwise. Here the operator $\circ$ denotes the Hadamard or element-wise product, i.e., $Y(i, j) = \Phi_{MC}(i, j)X(i, j)$. Given observations Y and the measurement matrix Φ_{MC}, the low-rank matrix completion problem is stated as follows:

$$\min_{X \in \mathbb{R}^{L \times N}} \text{rank}(X) \quad \text{s.t.} \quad Y = \Phi_{MC} \circ X, \tag{5}$$

in which $rank(X)$ returns the rank of X. Similar as before, solving Eq. (5) is intractable due to its NP-hardness. An effective alternative is the nuclear norm relaxation, given by

$$\min_{X \in \mathbb{R}^{L \times N}} \|X\|_* \quad \text{s.t.} \quad Y = \Phi_{MC} \circ X, \tag{6}$$

where $\|\cdot\|_*$ denotes the nuclear norm, defined as the sum of all singular values. There are also other commonly used formulations, including the minimization of the Frobenius norm, see e.g. OptSpace [Keshavan and Oh 2009] and Set [Dai et al. 2011] among others. However, these formulations require a priori knowledge of the rank and the observation Y contains Gaussian noise.

Extensive studies have been done to develop various MC solvers; see e.g., [Becker 2012] for a nice summary. These are generally classified into non-convex and convex optimization problems. Despite the hardness of solving a non-convex problem (i.e. (5)), [Jain et al. 2010] introduced an Iterative Hard-Thresholding (IHT) algorithm with nice properties under certain conditions. Solving (6) is considerably easier; classical convex optimization tools such as Linear Programming, Interior Point, and Semi-Definite Programming may be directly applied. Furthermore, faster algorithms based on Iterative Soft Thresholding (IST) [Goldfarb and Ma 2011], Lagrangian, Penalty and Augmented Lagrangian [Cai et al. 2010] may be used to improve computational efficiency. For simplicity, we will use IHT and IST as the solver of Eq. (5) and (6) in our subsequent experiments, respectively. Their Matlab source code is taken from [Majumdar 2010].

4. APPLYING COMPRESSIVE SENSING TO SINGLE LOCATION RECOVERY

In this section we discuss the selection of a measurement matrix Φ and a representation basis Ψ in the CS framework for the recovery of a single-location signal $\boldsymbol{x}$.

4.1. Measurement Scheduling Matrix Φ

Recall that the original soil moisture signal $\boldsymbol{x}$ is an $N \times 1$ vector, and the $M \times N$ measurement matrix Φ specifies a measurement scheduling policy: it contains a "1" in the (m, n) position ($1 \leq m \leq M$, $1 \leq n \leq N$) if the m-th measurement is taken at time n. The physical nature of the instrument is such that only a single measurement is taken at any scheduled time, i.e., upon actuation, the soil moisture probe takes one measurement of the soil moisture process at the time of actuation; the same point in

the process cannot be measured more than once due to causality. This implies that, regardless of the schedule, Φ contains one and only one "1" element in any row, and at most one "1" in any column, and "0" everywhere else. As $M < N$, there will be exactly $N - M$ empty (all zero) columns, making the Φ matrix extremely sparse. This is very different from what is commonly studied in the literature, e.g., the Gaussian measurement matrix that is very dense with virtually no zero entries. This poses a significant challenge, since in general the measurement matrix is required to be dense with at least one non-zero entry in each column [Candés 2006; Candés et al. 2006]. This same challenge also arose in a routing problem studied in [Quer et al. 2009], where the authors reported less than satisfying results due to the difficulty in finding the right Ψ matrix to match the highly constrained Φ matrix.

With the above constraint in mind, we will consider two types of schedules. The first has periodic sampling times, where measurements are taken at intervals of $\lfloor \frac{N}{M} \rfloor$ discrete units of time; this will also be referred to as the *uniform schedule* (US), and the corresponding matrix denoted as Φ_U. The second follows random sampling times generated using certain probability distribution with an average sampling rate of M/N; this will be referred to as the *random schedule* (RS), and the corresponding matrix denoted as Φ_R. Note that since N is not always an integer multiple of M, in such cases under the uniform schedule the first measurement point is randomly selected within a small range $[1, \lfloor \frac{N}{M} \rfloor]$, with subsequence measurements taken every $\lfloor \frac{N}{M} \rfloor$ time units till we exhaust N. The reason we consider these two relatively simple schedules is due to their ease in implementation. We do compare their performance with more complex schedules (see below and the closed-loop scheduling in Section 8). As we shall see the estimation accuracy of our method turns out to be highly robust against the measurement schedule.

For comparison purposes, we will also consider the commonly studied Gaussian scheduling (GS) matrix Φ_G mentioned above. It should be emphasized that this matrix is *not* practical in our scenario: since each row in this matrix has typically many non-zero entries, it requires each measurement be a linear combination of multiple samples of the soil moisture process. More importantly, as there are virtually no empty columns, this matrix essentially requires the collection of nearly all samples of the original signal. This obviously defeats our basic objective of minimizing the amount of measurements taken.

4.2. Representation Basis Ψ

As mentioned, there are two main criteria in selecting a good representation basis Ψ: (1) its corresponding inverse has to sufficiently sparsify the signal $\boldsymbol{x}$, and (2) it has to be sufficiently incoherent with the measurement matrix Φ. This is highly non-trivial due to the sparse nature of our Φ matrix. In addition, to generate a basis that meets the above two criteria without exploiting any feature of the signal can take a large amount of trial-and-error. Thus typically certain known features of the signal are taken into account in searching for a suitable basis. To this end, we observe that the soil moisture process (seen earlier in Figure 1 and 2) is relatively smooth and slow changing, except at the onset of a rainfall. This suggests that the signal might be sparsely represented if we consider the difference between two adjacent sample values. This motivates the following *difference matrix* M_D:

$$M_D = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \\ 0 & 0 & 0 & \cdots & 0 & -\eta \end{bmatrix}$$

where the last element $\eta > 0$ ensures that M_D is invertible.

Ideally, one would like the projection of $\boldsymbol{x}$ on M_D, $\boldsymbol{s} = M_D\boldsymbol{x}$, to be a vector containing many zero/near-zero entries. If this is the case, then the original signal $\boldsymbol{x}$ can be sparsely represented in the M_D domain as $\boldsymbol{x} = M_D^{-1}\boldsymbol{s}$. In the numerical experiments presented in the next section we will use M_D^{-1} as a choice for the representation basis and denote $\Psi_D = M_D^{-1}$.[4] Note that with the above representation of the signal in the gradient domain, the objective value in (3) is equivalent to the total variation of the signal. Moreover, using the representation basis Ψ_D with our measurement matrix (either Φ_U or Φ_R), it can be shown that the CS reconstruction is reduced to the family of monotone interpolation between any pair of neighboring samples, of which linear interpolation belongs.

The temporal correlation and piece-wise smooth feature of the soil moisture process also suggests that it may be more compactly represented through a Haar (wavelet) transformation M_H. We will thus use $\Psi_H = M_H^{-1}$ as a second choice for the representation basis Ψ in our experiment.

We now check the quality of these two matrices against the two criteria outlined earlier. It turns out that neither Ψ_D nor Ψ_H can produce a precisely K-sparse signal for $K \ll N$ (the amount of non-zero elements are above 50% in both cases). However, the resulting $\boldsymbol{s}$ is approximately K-sparse if we neglect small elements. Table I shows the sparsity of Ψ_D and Ψ_H under different scale N, the signal size. Here the entire data set is segmented into signals of size N, and the transformation is applied over each signal. The overall sparsity is calculated as the sum of all entries of $\boldsymbol{s}$ with values of at least 0.1, and averaged over all signals. We see that the difference matrix is much more effective than Haar transform.

N	Sparsity			
	Garden data		Farm data	
	Ψ_D	Ψ_H	Ψ_D	Ψ_H
64	3.4	9.5	3.6	10.6
128	6.0	19	6.2	21.3
256	11	38.1	11.4	42.6
512	21.2	75.9	22	85
1024	41.3	152.3	43	170.3
2048	81.5	303.5	85.5	340

Table I. Comparison of approximate sparsity.

We next examine the incoherence between these two representation bases and our measurement matrices. As the notion of coherence is not defined for non-orthogonal matrices, in the following we will use its dual, namely incoherence to indirectly measure the correlation between the proposed Φ_U (Φ_R) and Ψ_D (Ψ_H). The incoherence of two matrices are measured as follows [Quer et al. 2009]. Projecting each row of Φ onto the space spanned by the columns of Ψ we get:

$$\zeta_j = (\Psi^T\Psi)^{-1}\Psi^T\phi_j^T, \tag{7}$$

where ϕ_j is the jth row of Φ and ζ_j is the vector of coefficients corresponding to its projection on the space spanned by the columns of Ψ. A measure of the incoherence is

[4]A similar operation aiming to sparsify a spatial 2D signal was used in [Quer et al. 2009], but as mentioned in the introduction it did not lead to very good performance.

N	$\mathbf{I}(\Phi_R, \cdot)$	
	Ψ_D	Ψ_H
64	63	8
128	125	19
256	255	34
512	512	81
1024	1024	131
2048	2048	324

Table II. Comparison of incoherence.

then defined as

$$\mathbf{I}(\Phi, \Psi) = \min_{j=1,\cdots,N} \left[\sum_{i=1}^{N} 1\{\rho_i^j \neq 0\} \right] \in [1, N], \tag{8}$$

where ρ_i^j is the ith entry of vector ζ_j and $1\{A\}$ is the indicator function. The larger this quantity, the more incoherent the two matrices. In Table II, we show the incoherence obtained from (8), for the random measurement matrix Φ_R with the two choices of Ψ at different scales. We see Ψ_D has much higher incoherence with the measurement matrix than Ψ_H does. Thus the representation basis Ψ_D fares better than Ψ_H in both the sparsity and the incoherent criteria.

5. APPLYING MATRIX COMPLETION TO MULTIPLE LOCATION RECOVERY

To apply matrix completion techniques to the recovery of soil moisture processes at multiple locations, a measurement matrix Φ_{MC} needs to be chosen and a recovery algorithm is used to determine the signal X from the set of measurements $Y = \Phi_{MC} \circ X$.

An important prior for matrix completion to work effectively is for the signal matrix X to be a low-rank matrix. We therefore begin by examining if the soil moisture data meets such a requirement. We perform the low-rank test on the simulated *Farm data* as we currently lack high quality real data involving a sufficient number of locations. The tool we use is the Principal Component Analysis (PCA), which transforms a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables (i.e. principal components). In general, the number of principal components represents the rank of the original signal matrix. PCA is usually done through singular value decomposition (SVD) on a matrix with zero empirical mean, by shifting the mean of the original data matrix to zero. Following methods used in [Lakhina et al. 2004], we first shift X to zero-mean and denote the resulting matrix by X^c, i.e., the (i,j)-th element $X^c(i,j) = X(i,j) - \frac{1}{N}\sum_{j=1}^{N} X(i,j)$. We then use the ratio of the variance captured by the top k components and the total data variance, namely $\sum_{i=1}^{k} \|X^c u_i\|^2 / \sum_{i=1}^{N} \|X^c u_i\|^2$, where u_i is the ith principal component of X^c. If most of the data variability resides in the first r component (See Section 3.2), X^c can be effectively represented by an r-dimensional subspace of $\mathbb{R}^N$ and has an approximated r low-rank structure.

Figure 3 plots the ratio of total variance captured by the top k principal components of X^c under different network scales (i.e. L). For simplicity in this illustration we used a square matrix X, i.e., $L = N$. This figure shows that in all cases most of the data variance is well captured by no more than 4 principals components, much less than N (or L), suggesting that the soil moisture signal exhibit good low-rank structure.

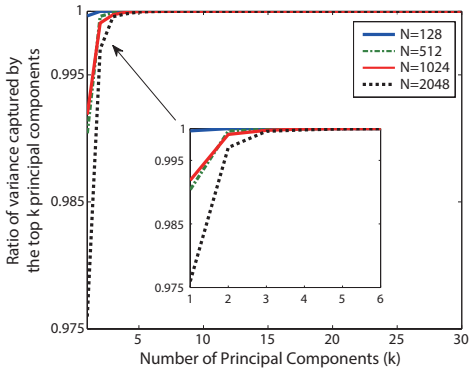


Fig. 3. Low-rank structure of the Farm data

6. NUMERICAL EXPERIMENTS

In this section, we examine the performance of applying CS and MC techniques to the soil moisture measurement scheduling and estimation problem. Throughout this section, the Garden data refers to the surface level (top level) soil moisture process shown in Figure 1. Unless otherwise specified, we trim the real and simulated data to 4096 observations (or discrete times steps) in total, for the convenience of the experiment. As matrix completion involves multiple locations, we will only use the simulated Farm data for experiments on MC since the Garden data is not of sufficiently large scale.

To perform the experiments, we divide each data set into W windows (or signal/matrix) of N points each. Our sampling and recovery algorithms are applied to each window separately and similarly. To be specific, we will use $\boldsymbol{x}_w$, a $N \times 1$ discrete signal, to denote the original data associated with the w-th window ($1 \leq w \leq W$) at a single location. When L locations are available, we use X_w, a $L \times N$ matrix, to denote the w-th window of all L locations. $\boldsymbol{x}_w$ and X_w are used in the CS and MC recovery frameworks, respectively. Within each window, $M < N$ measurements are taken at each location. Thus across the entire field LM measurements are collected. The goal of CS and MC is to reconstruct $\boldsymbol{x}_w$ and X_w from the M and SM direct observations, respectively.

Breaking the data set into windows of N allows us to balance the computational complexity and delay with estimation accuracy. If the measurement and reconstruction is done in a close-to-real time fashion, then it is desirable to perform these operations over a smaller N. On the other hand, larger N generally results in better estimates, though at the expense of increased computational complexity.

For a single location, the reconstruction quality is measured by the following average error criterion:

$$AvgError = \frac{1}{NW} \sum_{w=1}^{W} \|\boldsymbol{x}_w - \hat{\boldsymbol{x}}_w\|_1, \tag{9}$$

where W is the number of windows ($W = 4096/N$); $\boldsymbol{x}_w$ and $\hat{\boldsymbol{x}}_w$ are the true and estimated signal of window w; $\|\boldsymbol{x}_w - \hat{\boldsymbol{x}}_w\|_1$ is the sum of absolute errors in the estimate. This error is further averaged over 20 random trials when the measurement schedule is generated randomly. When there are L locations, the estimation accuracy is measured by $FieldAvgError$, which is the average of $AvgError$ over the L locations. The sampling times under a random schedule (RS) is generated using a uniform proba-

bility distribution. In the CS framework, the scale-down rate η used in the difference matrix M_D is 0.001.

6.1. Performance of Compressive Sensing

In this subsection, we investigate the effectiveness of CS techniques applied to the single-location scenario. The overall quality of signal reconstruction is determined by three elements: the choice of measurement scheduling matrix Φ, the choice of the representation basis Ψ, and the choice of a recovery algorithm (also referred to as a *solver* below).

In the following, we will first examine what types of solvers work best with our choices of Φ (Φ_U and Φ_R) and and Ψ (Ψ_D and Ψ_H). We then compare the performance of different combinations of these matrices. We end with an adaptive scheduling method to further improve the capability of CS to capture the onset of rainfall events and to adjust subsequent sampling rates.

6.1.1. Effect of recovery algorithms. We first investigate what CS recovery algorithms or solvers work best with our selection of Φ and Ψ. A candidate list of these algorithms is discussed in Section 3.1, and they include SL0, IRWLS, OMP and BP (using LP). Of these, SL0 aims to minimize the l_o norm, while the others aim at minimizing the l_1 norm. The Matlab codes of IRWLS, OMP, and LP are obtained from Sparse Lab [Sparse Lab 2007] and SL0 from [SL0 2008].

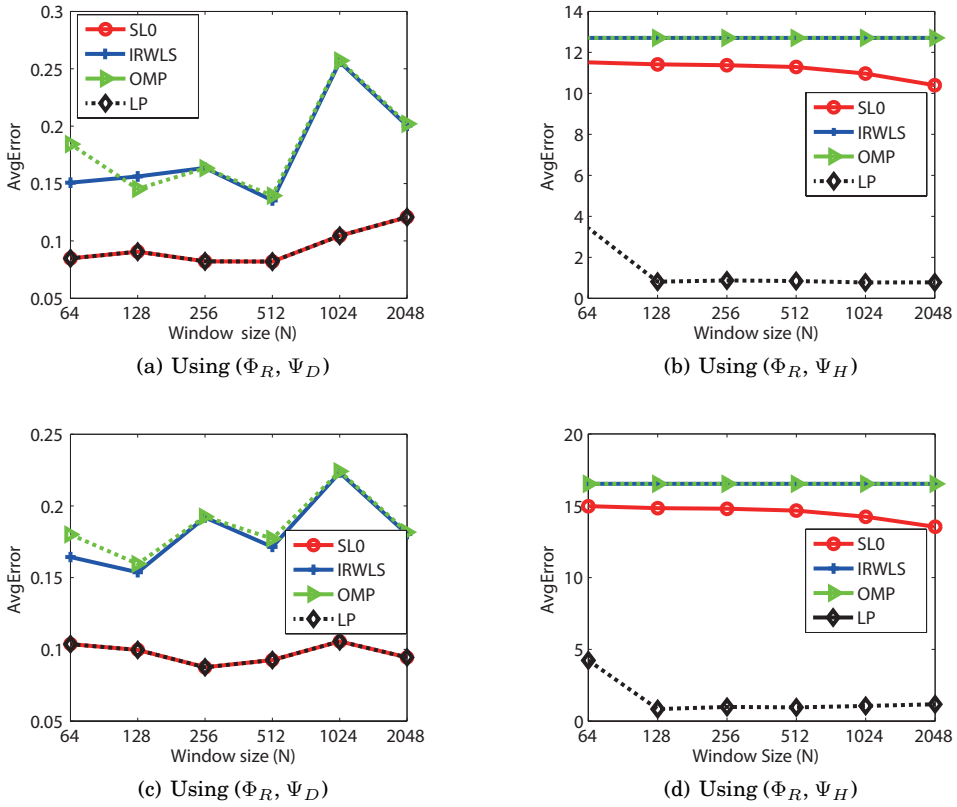
A set of experiments are run by increasing the window size N. For each window, the number of measurement M is set to 10% of N. As the Haar basis requires N to be a power of two, we set $N = 2^p, p = 6, 7, 8, 9, 10, 11$. For each value of N we evaluate the average estimation error in applying one of the above algorithms to the two data sets while using Φ_R as the measurement matrix, and Ψ_D and Ψ_H respectively as the representation basis. The results are shown in Figure 4. Our main observations are as follows.

Firstly, compared to Haar, the difference matrix Ψ_D shows significant advantage regardless of the data and the solver used: the $AvgError$ of Ψ_H is at least 10 orders of magnitude higher than Ψ_D. As discussed earlier, Ψ_H sparsifies the data better than Ψ_D, while the later has much higher incoherence with the measurement matrix Φ_R. This observation thus suggests that in this case incoherence is more critical in determining the effectiveness of these solvers.

Secondly, we see that there are substantial performance gaps among different recovery algorithms. When Ψ_D is used, we see from Figure 4(a) and 4(c) that SL0 and LP perform the best (these two curves almost completely overlap) and they significantly outperform IRWLS and OMP. When Ψ_H is used, Figure 4(b) and 4(d) (the two curves of OMP and IRWLS almost completely overlap as well) again show that LP performs the best and significantly so. Furthermore, since the soil moisture content is measured between 10 and 35 (in %), an estimation procedure effectively fails if its $AvgError$ exceeds 10. In this sense except for LP, all other solvers do not work successfully with Ψ_H. These observations suggest that the estimation quality of LP and SL0 is much more robust to the level of sparsity than the faster solvers (OMP, IRWLS) when there is sufficient incoherence (in the case of Ψ_D). At the same time, LP is also robust to weak incoherence when there is sufficient sparsity (in the case of Ψ_H).

Based on these results, for the rest of our numerical evaluation we will limit our attention to SL0 and LP. As LP is more computationally costly than SL0, we will only use SL0 when Ψ_D is used, and LP when Ψ_H is used.

6.1.2. Effect of measurement matrices. We next study the impact of scheduling methods on the reconstruction quality. The three measurement matrices we examine are random (Φ_R), uniform (Φ_U) and Gaussian (Φ_G) introduced in Section 4.1. Note again that



(a) Using (Φ_R, Ψ_D) (b) Using (Φ_R, Ψ_H)

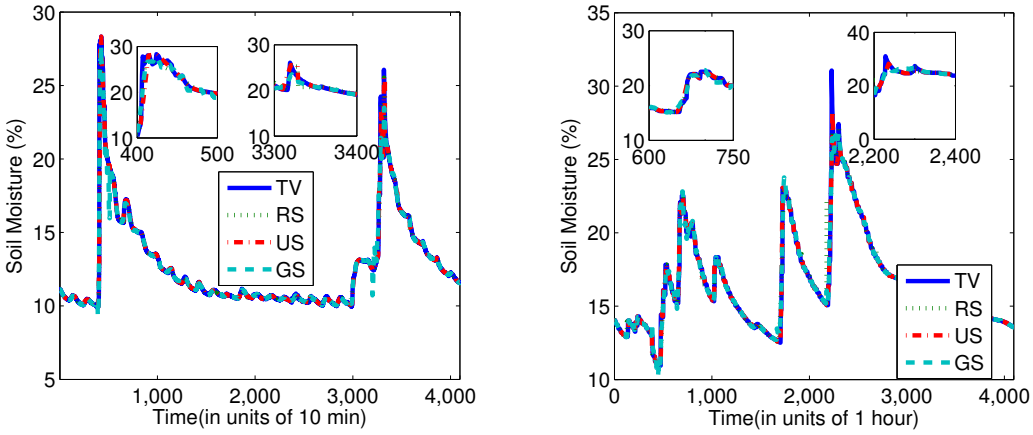
(c) Using (Φ_R, Ψ_D) (d) Using (Φ_R, Ψ_H)

Fig. 4. Comparison of reconstruction performance: (a)(b) Garden data; (c)(d) Farm data

the Gaussian matrix is used as a point of comparison; it is not a practical scheduling policy for our problem.

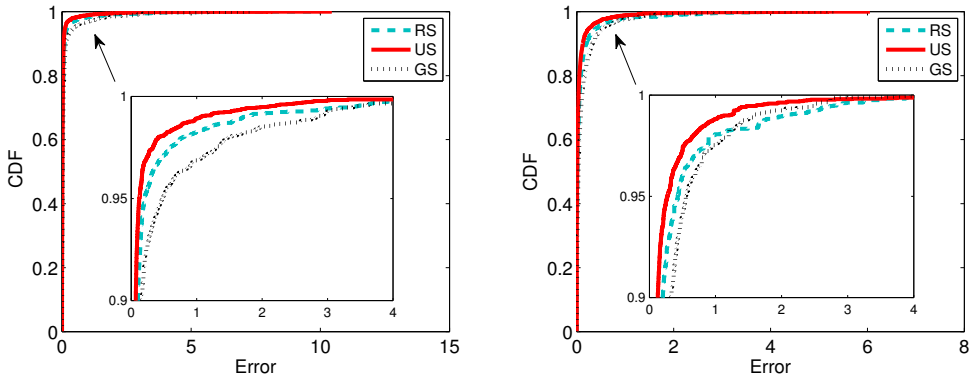
Figure 5 illustrates the reconstruction quality under these three scheduling methods, with respect to the true soil moisture values (denoted as TV in the figure). In this set of results, the representation basis and solver used are Ψ_D and SL0, respectively, the window size is $N = 128$, and the sampling rate is $M/N = 10\%$. To observe the finer differences among these methods, Figure 5 also provides zoomed-in views around two peaks in the original data. We also provide the empirical CDF of absolute estimation error in Figure 6, which is measured as the absolute difference between the real value and the estimate. This difference (the x-axis) has a unit of % soil moisture level. We see that all three scheduling methods perform very closely and provide reconstruction of the original signal to very high accuracy. Empirically, more than 90% of the reconstructed time series has error less than 1.

To be more precise in quantitative comparison, Table III further shows the corresponding $AvgError$ calculated over different time segments in the data. Here, a range of "Total" includes the entire period (i.e, from 1 to 4096), while the ranges "Peak1" and "Peak2" refer to the two peaks amplified in Figure 5 within each data set. Peak1 is given by the time intervals [400, 500] and [600, 750] for the Garden and the Farm data, respectively. Peak2 is given by the time intervals [3300, 3400] and [2200, 2400] for the Garden and the Farm data, respectively. As a comparison point, we also present the same set of results when the Haar basis matrix Ψ_H and the LP solver are used in



(a) Garden data (b) Farm data

Fig. 5. Performance comparison of different scheduling methods, using Ψ_D and SL0.



(a) Garden data (b) Farm data

Fig. 6. Performance comparison of different scheduling methods, using Ψ_D and SL0.

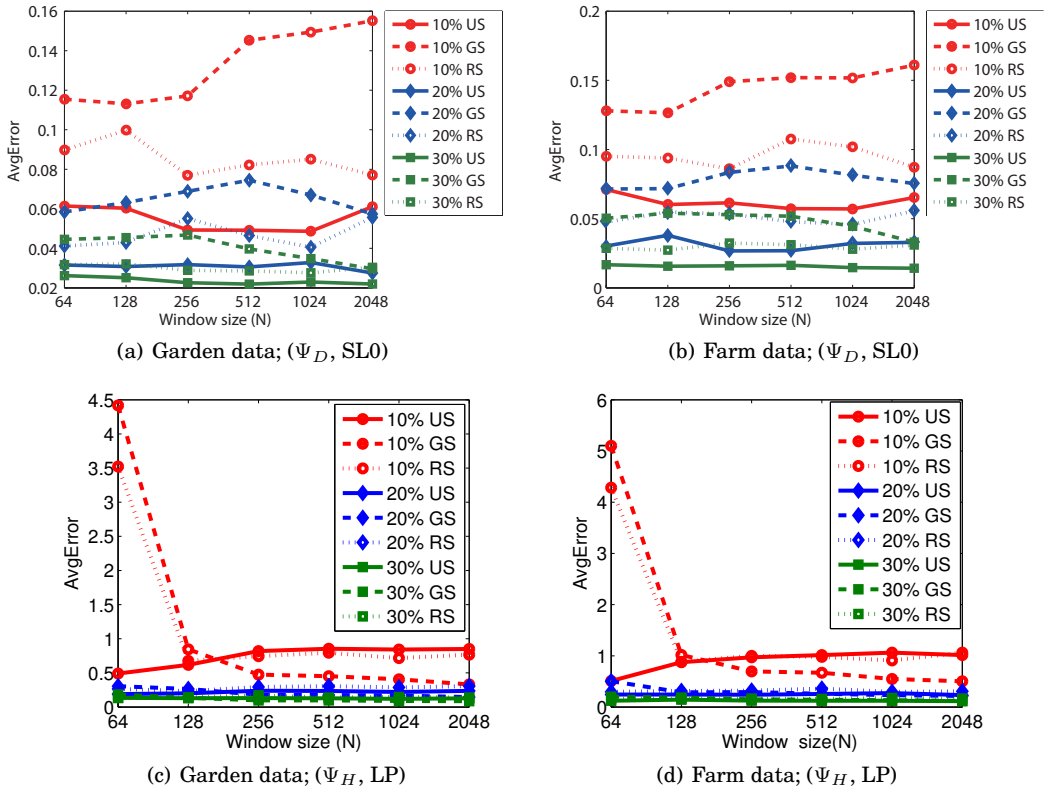
same table. We repeat the same calculation at different sampling rates, from 10% to 30%, and present these results in Figure 7.

When the same (Ψ, solver) pair is used, the main factor affecting the final reconstruction performance is the incoherence between Ψ and the measurement matrix Φ. The incoherence between (Φ_R, Φ_U, Φ_G) and Ψ_D is 125, 127, and 128, respectively, at $N = 128$. Thus it is not surprising that we see very close performance as evidenced in Figure 5. In particular, we see that the uniform (periodic) scheduling provides the best overall performance.

It is however interesting to see, from Table III, that both uniform and random scheduling outperform Gaussian scheduling, especially under the pair (Ψ_D, SL0). This is somewhat surprising, because the Gaussian measurement matrix has the highest incoherence with Ψ_D though by only a small amount. This is possibly due to the fact that while in theory Ψ_G should perform the best, such results often rely on the signal being precisely sparse, but in our case the signal x is only approximately sparsified.

(Ψ, solver)	Data	Range	Φ_R	Φ_U	Φ_G
(Ψ_D, SL0)	Garden	Total	0.0827	0.0623	0.1040
		Peak1	0.8738	0.7682	1.5367
		Peak2	0.2340	0.3795	0.3732
	Farm	Total	0.0909	0.0688	0.1280
		Peak1	0.4712	0.2552	0.3628
		Peak2	0.2156	0.3148	0.4973
(Ψ_H, LP)	Garden	Total	0.8553	0.7537	0.6473
		Peak1	12.9222	0.1744	7.1455
		Peak2	4.2919	4.1016	2.0057
	Farm	Total	0.9797	0.9537	1.3647
		Peak1	2.7720	0.5587	1.8389
		Peak2	4.0324	4.5034	13.3529

Table III. Comparison of scheduling methods.



(a) Garden data; (Ψ_D, SL0)

(b) Farm data; (Ψ_D, SL0)

(c) Garden data; (Ψ_H, LP)

(d) Farm data; (Ψ_H, LP)

Fig. 7. Performance comparison of different sampling rates.

6.1.3. Effect of representation basis. The above results show the different matrix M_D to perform very well in terms of the average estimation average. Recall, however, that the original observation that motivated this matrix was the slow-changing dry-down periods of the soil moisture process. It is thus natural to question how well it performs during the peak periods (the sharp increases during the onset of rainfalls and quick drop-offs immediately following rainfalls), and whether there are other, better representation bases (e.g. DCT, FFT, Haar, etc.) if one is only concerned with the peak peri-

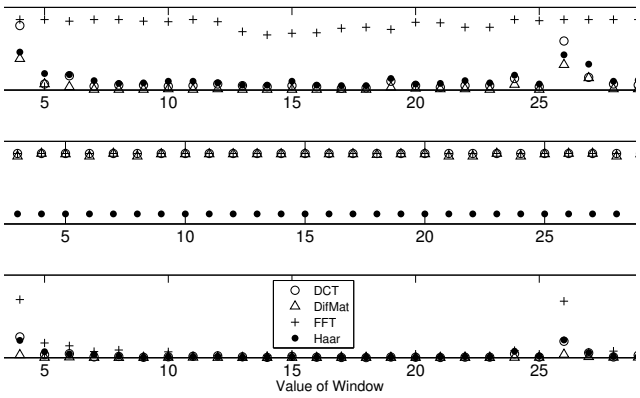
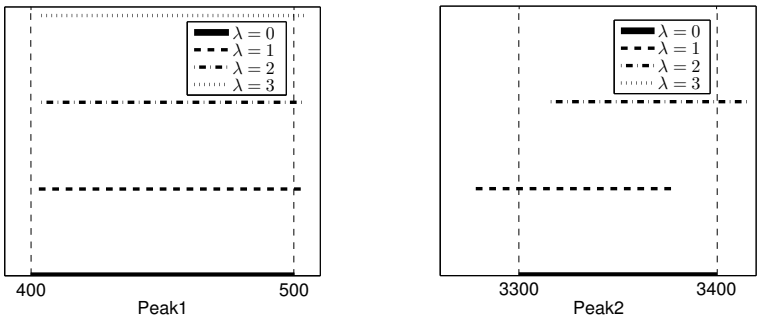


Fig. 8. Performance within each window with $N = 128$, using the Garden Data.

ods. To answer this question, we separately calculate the sparsity, incoherence and estimation error within each window of the signal, using different representation bases. The results are shown in Figure 8, corresponding to each window $w = 1, 2, \cdots, 32$, for a window size of $N = 128$ (this choice can guarantee that Peak1 and Peak2 each occupies an entire window). To ensure recovery performance, LP is used as the solver for all tested representation basis. We see that except for the two peak periods, the different bases lead to very similar performances, with the different matrix outperforming slightly in sparsity. What is interesting is that it appears that it is its superiority in sparsification, and in particular during the peak periods, that has given the different matrix the advantage in better accuracy. It also appears that between sparsity and incoherence the former plays a much bigger role – Haar is the worst in terms of incoherence but ultimately outperforms DCT and FFT because of better sparsity.

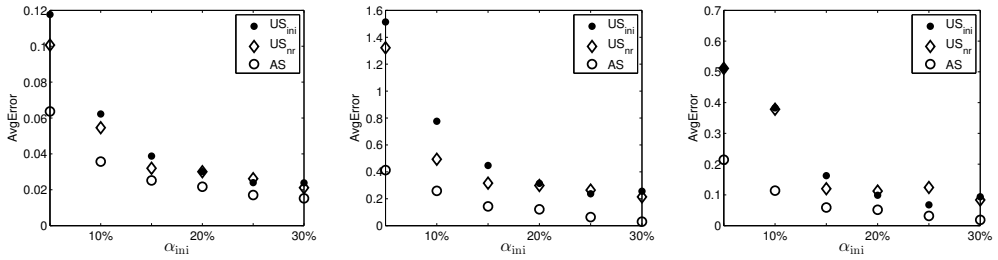
6.1.4. Adaptive measurement scheduling. The previously considered random and uniform sensing schedules are notably open-loop: they do not require historical information or current state of the process. If, however, one can acquire information on rainfall events, then intuitively one should measure more frequently during rainfall and reduce it during draught. This is also evidenced in the above results showing the majority of the estimation error arising from peak periods.

In this subsection we introduce a simple adaptive sampling (AS) algorithm that uses a threshold-based detection method to catch the start of a rainfall event. Specifically, if the difference between two consecutive observations is more than certain pre-set threshold $\lambda > 0$, a rainfall event is considered to have happened and the AS algorithm starts measurement at a higher sampling rate. This high-sampling period ends at sometime T after the initial detection, when the algorithm switches back to the normal, lower-sampling rate. Note that this end point does not necessarily coincide with the physical end of rainfall; it simply reflects the observation that typically after a while the data starts changing more slowly either due to the end of rain and the start of dry-down, or due to moisture saturation. This is also a simple way to avoid having to detect the end of a rainfall, which has proven harder to do than detecting onset. Different values of the threshold λ will have an impact on the detection accuracy: a large λ is more likely to have more misses, while a small value leads to more false positives. Figure 9 illustrates the rainfall detection performance using different threshold values and a constant $T = 100$. Here dashed lines denote the estimated rainfall durations with different λ (i.e. y-axis), while the solid line with $\lambda = 0$ represents the ground truth



(a) Detection of Peak1 (b) Detection of Peak2

Fig. 9. Performance of rainfall detection using different thresholds λ.



(a) Entire data (b) Peak1 (c) Peak2

Fig. 10. Comparison of different scheduling methods on the Garden data.

of the two rainfall events. We see that using $\lambda = 3$ fails to detect the second peak (no appearance in Figure 9(a)). This means that $\lambda \geq 3$ fails to detect Peak2.

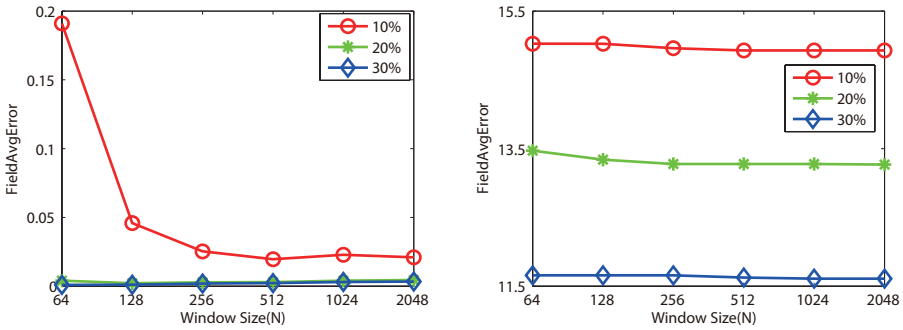
In the following experiments we will use $\lambda = 2$ and $T = 100$. The initial, normal sampling rate is denoted by $\alpha_{\rm ini}$. Upon detection of a rainfall the sampling rate increases by k-fold, $k > 1$ being the extension factor for the next T time slots, and the sampling rate returns to the initial value $\alpha_{\rm ini}$ afterwards. Figure 14(a) shows the comparison among results of different scheduling methods by varying $\alpha_{\rm ini}$ from 5% to 30% with a 5% increment. Results on the two peaks are shown in Figure 6.1.4 and 14(c). In these figures, AS denotes the adaptive sampling method using the above outlined detection procedure to determine the start and end of rainfall events with $k = 3$, and $\text{US}_{\rm ini}$ denotes uniform sampling with rate $\alpha_{\rm ini}$, while $\text{US}_{\rm nr}$ denotes uniform sampling with the same sampling cost as AS. Note that under AS, we use $\lambda = 2$ and $T = 100$ to determine when to use the higher sampling rate $k\alpha_{\rm ini}$; as a result the average sampling rate is higher than $\alpha_{\rm ini}$ if at least one rainfall event is detected.

As we can see, for a given $\alpha_{\rm ini}$, adaptive sampling outperforms uniform sampling overall, and the advantage is shown more clearly in Figure 6.1.4 and 14(c) for the two peaks. Furthermore, when $\alpha_{\rm ini}$ is very small the detection algorithm may completely miss a rainfall event. This is seen in Figure 14(c) at $\alpha_{\rm ini} = 5\%$ and 10%, which leads to $\text{US}_{\rm ini}$ and AS having the same estimation error.

6.2. Performance of Matrix Completion

In this part, we show the performance of using MC to reconstruct soil moisture processes at multiple locations simultaneously. The data we use is the simulated Farm data, which contains 2594 locations, each with 5280 measurements. To avoid excessive computation by the solvers (i.e. IHT and IST) that involve intensive matrix operation (i.e., SVD), from this data set we select a subset containing $L = 128$ locations, each with 4096 measurements. The window size is set to $N = 128$.

6.2.1. Effect of the solver. Figure 11 shows the recovery performance using IHT and IST together with RS under different sampling rates and window sizes; the RS measurement schedules are chosen independently across different locations. We see that IST, which solves the convex nuclear norm objective, shows significant advantage over IHS, which is aimed at approximating solutions to a non-convex optimization problem (Eq. (5)). In fact, for this data set IHT appears to be intractable as the measured error are consistently above 10. For IST, the effect of window size is negligible when the sampling rate is more than 20%. Thus we can choose a smaller window if computational resources are limited or small latency is required. On the other hand, it is a bit surprising to see that sampling rate above 20% seems to bring negligible additional gain in accuracy.



(a) IST (b) IHT

Fig. 11. Performance comparison of IST and IHT using RS.

6.2.2. Effect of the measurement matrix. We further compare the use of RS and US together with IST. We fix the window size at 128, sampling rate at 20%, and let the RS schedules be chosen independently across different locations. As we have seen in Figure 11(a), MC is highly accurate under RS; an example of the recovered signal at a randomly chosen location is shown in Figure 12. When US is used across locations with asynchronous sampling times such that there is at least one node taking measurement at each time instant, we obtain reconstructed signals with similar quality as RS; the illustration is omitted for brevity. However, US can be problematic when the schedules are synchronized in such a way that no measurements are taken at one or more time instances in a window. This means that the resulting measurement matrix would contain at least one zero-column. The low-rank recovery mechanism can then output any arbitrary linear combinations of other columns as the reconstructed soil moisture levels for these time slots with zero columns. An extreme case is when the sampling rates of US are the same for all locations and the schedules of measurement

Node	Range	MC	CS
Node 43	Total	0.0214	0.1346
	Peak1	0.0152	0.4689
	Peak2	0.0343	0.4682
Node 86	Total	0.0338	0.1065
	Peak1	0.0210	0.7682
	Peak2	0.0390	0.3629
Node 90	Total	0.0418	0.1615
	Peak1	0.0972	0.6070
	Peak2	0.0697	0.9619

Table IV. Comparison between MC and CS - Part II: $AvgError$ of 3 locations.

are aligned, and we would obtain as a result of the solver, a vast majority of estimates being zero.

6.2.3. Comparing results using MC versus CS. We end this section with a comparison of CS and MC, shown in Figure 12, whereby under the CS method the same sampling and recovery process is applied independently to each location. The results here are based on RS with 20% sampling rate under both methods. Under CS, the representation basis and solver used are Ψ_D and SL0, respectively. Under MC, we choose IST as the solver. Figure 12 suggests that both CS and MC are quite effective. To get a closer look, the $AvgError$ of 3 randomly chosen (out of 128) locations are detailed in Table IV. The advantage of MC over CS is quite obvious. This is because under CS the recovery at each location only relies on neighboring measurements in time, while MC benefits from neighboring observations both in time and in space.

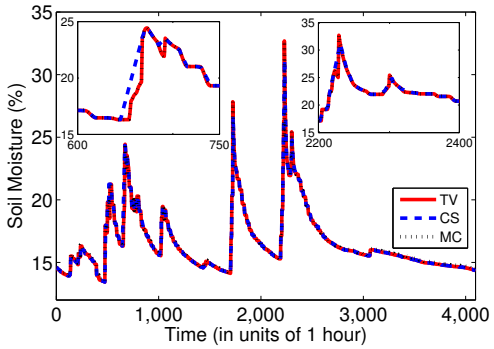


Fig. 12. CS versus MC using RS - Part I: reconstructed signals.

7. ROBUSTNESS OF SPARSE SAMPLING

In practice, anomaly in measurements may occur due to various causes including sensor faults. With outliers in the measured data, direct application of our proposed methods may provide inferior reconstruction results for the following reason. Unlike GS under which the measurements (as entries in y or Y) are taken as linear combinations of all signal values, thus mitigating the effect of occasional outliers, our measurement schemes (RS or US) take directly samples of the underlying soil moisture process; thus they can be quite vulnerable to outliers. However, if we assume that outliers are

sparsely distributed in both space and in time, which is most likely the case in reality especially if they are due to sensor faults, our methods can be augmented using results from existing literature to be robust against intermittent though possibly significant data corruption, as we detail in this section. We will limit our attention to data anomaly/outlier rather than more pervasive noise present in the measurements. This is because while standard calibration formula for the in-situ capacitance probes we use in field deployment (i.e. model ECH_2O EC-5 from Decagon) produces a nominal accuracy of 3-4% under various soil types, an improved soil moisture calibration model shown in [Moghaddam et al. 2010a] can generate an accuracy better than 1%. Thus in all our discussion we have treated the measurements as noise-free.

7.1. Robust CS

For CS-base schemes, [Luo et al. 2009] provided a method of reshaping the measured data with compromised readings as a new sparse signal under a new basis in order to address the presence of abnormal readings. Consider each outlying point as the superposition of a normal signal and a deviating spike. Denote then the soil moisture measurements with outliers by a vector x of length N such that $x = x_0 + x_1$, where x_0 is the normal values that is approximately sparse in the Ψ_D domain and x_1 the spike. Since the outliers are sporadic, x_1 is exactly sparse in the time domain. Therefore, x can be further written as $x = \Psi_D s_0 + I s_1$, where I is the identity matrix, s_0 the sparse coefficient vector of x_0 in Ψ_D and $s_1 = x_1$ the sparse outlier signal.

Construct a new representation basis $\Psi = [\Psi_D \ I]$ and a stacked signal vector $s = [s_0^\top s_1^\top]^\top$. Given the (approximate) sparsity of s following the above discussion and $x = \Psi s$, we obtain the (approximate) sparsity of x in the Ψ domain. Using RS or US, the collected measurements are given by $y = \Phi x = \Phi\Psi s$. We have now reformulate the problem with measurements containing outlier in the CS framework, and the best approximation of s can be obtained via recovery algorithms discussed in Section 3.1. Let $\hat{s}$ of length $2N$ be the overall reconstructed signal, with the first N entries of $\hat{s}$ denoting the estimated coefficients of the original normal data in the Ψ_D domain, and the other N entries denoting the outlier vector s_1, in which the non-zero elements indicate the estimated positions of outliers in the measurements.

7.2. Robust MC

Similar to the idea underlying robust CS, under MC the entire field measurements X can be rewritten as $X = X_0 + S$, where the normal measurements X_0 is a low-rank matrix and S is the sparse outlier matrix, and the observation matrix Y is given by $Y = \Phi_{MC} \circ (X_0 + S)$. The robust MC in essence solves the matrix decomposition problem, decomposing Y into a low-rank matrix X_0 and a sparse S. Unlike the robust CS presented above which recasts the problem into an equivalent outlier-free CS problem ((3)), existing robust MC techniques in general consider variants of the outlier-free MC formulation given in ((6)). A natural variant in this context is a regularized version of 6, given by

$$\min_{X \in \mathbb{R}^{L \times N}} \|X_0\|_* + \lambda \|S\| \quad \text{s.t.} \quad Y = \Phi_{MC} \circ (X_0 + S), \tag{10}$$

where $\|\cdot\|$ is an appropriately chosen matrix norm, and λ is the regularization parameter. This formulation has been investigated in [Chen et al. 2011a; Chen et al. 2011b], and the similar formulation with full observation (i.e., Φ_{MC} being an all-one matrix) has been studied in a general context of matrix decomposition, e.g., [Chandrasekaran et al. 2011]. An implementation of the solution process using Augmented Lagrange Multiplier methods has been proposed by [Chen et al. 2011b]; however, this approach suffers from its lack of scalability. An alternative formulation is studied in [Waters

et al. 2011], shown as follows:

$$\min_{X_0, S \in \mathbb{R}^{L \times N}} \|Y - \Phi_{MC} \circ (X_0 + S)\|_2 \quad \text{s.t.} \quad \text{rank}(X_0) \leq r, \|S\|_0 \leq K. \tag{11}$$

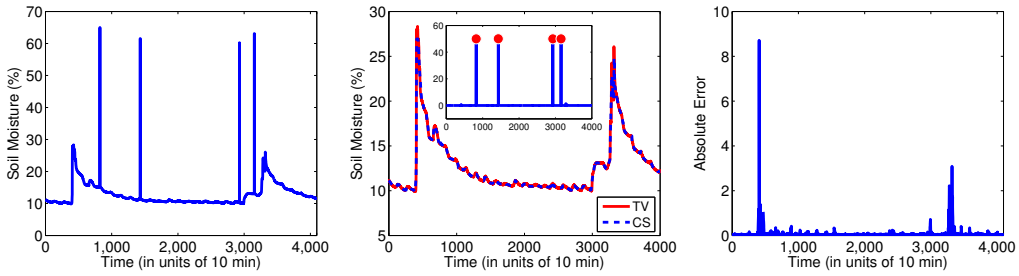
and a greedy algorithm SpaRCS is developed for (11), which is computationally efficient and scalable.

It should be noted that the above methods require certain prior knowledge on the data and error, in order to be able to appropriately set the regularization parameter λ or the bounds r and K on rank and sparsity; this can be difficult to do in practice.

7.3. Performance evaluation

Below we present the performance of the above robust methods with measurements containing outliers. Since the maximum of soil moisture readings seen in our data is around 35, we synthesize the outliers by adding a value of 50 to the original measurement at randomly chosen time instances from the scheduled times; the radio of abnormal to normal measurements is set to 0.001. In addition, the sampling rate is set to 0.1 and RS is used as the underlying sampling strategy. For both robust CS and robust MC, we use the window size $N = 128$.

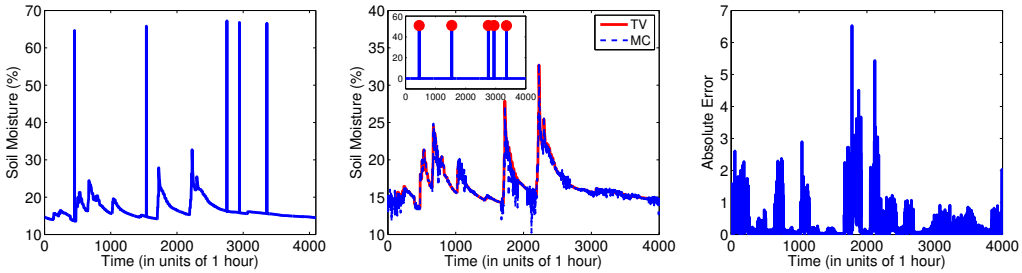
Figure 13 shows the data with synthesized outliers and the reconstruction outcomes of the CS-based method on the Garden data. As can be seen, though the measurements contain several outliers, we are able to recover the original normal moisture process with high accuracy as well as the positions of the outliers. Similar performance can be observed on the the simulated *Farm data*; there results are omitted for brevity.



(a) Data with outliers (b) Reconstructed signal and estimated outliers (c) Absolute estimation error

Fig. 13. Robustness of CS-based methods on the Garden data.

To assess the robust MC scheme, we use the simulated Farm data again due to the lack of large-scale real data, with $L = 128$ locations/sensors over 4096 time slots. Outliers are synthesized for each location in the same way as under robust CS, and we randomly choose one location to examine the recovery performance. We use the SpaRCS algorithm (11), which was reported to have superior recovery accuracy and convergence rate compared to the other formulations including (10). The synthesized data and the results are reported in Figure 14. For the results shown here, we have used $r = 1$ as the rank constraint for the chosen number of nodes and the window size, in light of the empirical low-rank structure we have observed in Figure 3; we set the sparsity constraint K to the exact sparsity of the outlier part S, which would be infeasible in practice. The robustness of MC appears less satisfactory than that of CS in this investigation, and the reconstruction is noisier. We have observed even worse recovery using higher values of the rank constraint r, and various estimates of the



(a) Data with outliers (b) Estimated outliers (c) Absolute estimation error

Fig. 14. Robustness of MC-based methods on the Farm data.

sparsity constraint K. We attribute its relatively poor performance against robust CS to the difficulty in selecting these parameters.

The above comparison may suggest that ultimately CS may be a preferred sparse sampling and reconstruction scheme in practice, when outliers are a legitimate concern. This is in addition to a weakness observed earlier on MC, which is its susceptibility to synchronization in measurement schedules across different locations. Synchronized schedules may be much easier to implement in practice and more energy efficient to manage, thus giving CS another advantage.

8. CLOSED-LOOP MEASUREMENT SCHEDULING

As pointed out in the introduction, the type of scheduling policies studied so far (uniform, random, Gaussian) are all open-loop ones, i.e., the scheduling decision is independent of the past and current state of the soil moisture process, independent of past scheduling decisions, and does not exploit any physics in the soil moisture dynamics. If one possesses statistical information on the underlying random process, then a conceptually more desirable approach is a closed-loop one, where the physics of the process, as well as the past observations and decisions are taken into account when making the next measurement decision. Note however that unlike the compressive sensing based open-loop approach developed in the preceding section, this closed-loop approach in general requires training in order to learn the statistics from past data. In this section we describe such an approach which formulates a partially observable Markov decision problem (POMDP); more details on this method can be found in [Shuman et al. 2010]. We then compare its performance with CS-based open-loop approach[5]. We further examine whether there is performance gain in combining these two approaches.

Under a closed-loop framework, the soil moisture evolution (again at a single location) is modeled as a discrete-time stochastic process $\{X_t\}_{t=0,1,2,\ldots}$. A decision U_t, $t = 1, 2, \cdots$, is made at time t: $U_t = 1$ denotes taking a measurement at time t and $U_t = 0$ otherwise. If a measurement is taken, then a perfect observation $Y_t = X_t$ is made at time t, otherwise; a "blank" observation results.

An estimated process $\hat{X}_t$ of the soil moisture is formed, with each new observation, using all past observations (some of which are blanks) and all past scheduling decisions:

$$\hat{X}_t = h_t\big(Y_0, Y_1, \cdots, Y_t; U_1, U_2, \cdots, U_t\big). \tag{12}$$

[5]We do not attempt a comparison with MC based approach as the estimation problem involving multiple locations using POMDP is near computationally intractable.

Similarly, the scheduling decision for time $t+1$ is based on all prior observations and scheduling decisions:

$$U_{t+1} = g_t\big(Y_0, Y_1, \cdots, Y_t; U_1, U_2, \cdots, U_t\big) \in \{0, 1\}. \tag{13}$$

The sequences $\boldsymbol{h} := \big(h_1, h_2, \cdots\big)$ and $\boldsymbol{g} := \big(g_1, g_2, \cdots\big)$ are the *estimation* and *scheduling* policies, respectively. The optimal policy pair $(\boldsymbol{g}^\star, \boldsymbol{h}^\star)$ may be derived by adopting a certain cost (or reward) objective; below is an example of an infinite-horizon expected discounted cost:

$$(\boldsymbol{g}^\star, \boldsymbol{h}^\star) = \arg\min_{(\boldsymbol{g},\boldsymbol{h})} \mathbb{E}^{\boldsymbol{g},\boldsymbol{h}}\bigg\{ \sum_{t=1}^{\infty} \alpha^{t-1} \cdot \Big[c(U_t) + \rho\big(X_t, \hat{X}_t\big) \Big] \bigg\}, \tag{14}$$

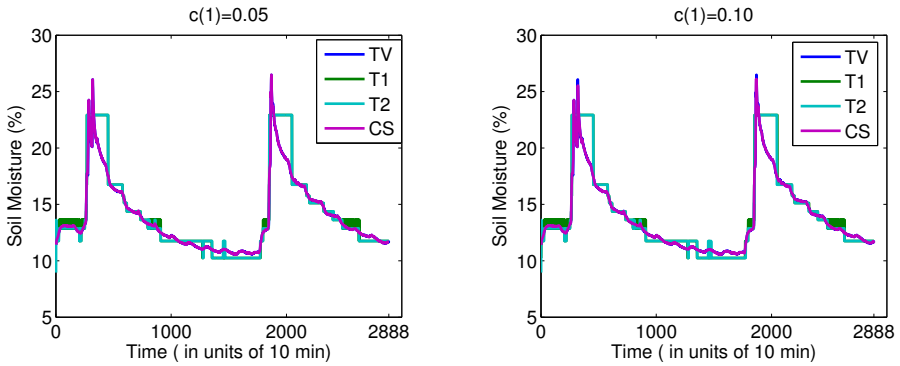
where $\alpha \in \{0, 1\}$ is the discount factor, $c(U_t)$ is the measurement cost, and $\rho\big(X_t, \hat{X}_t\big)$ is a penalty on estimation error, e.g., the mean squared error. The expectation is taken over known statistics of the process X_t.

Compared to an open-loop approach, the above closed-loop framework is conceptually precise (it has an explicit and well-defined optimization criterion), and it allows one to adjust the tradeoff between the measurement cost and the estimation error (by for instance introducing weights for the two cost terms). The main disadvantage of such an approach lies in (1) it requires a priori statistical knowledge of $\{X_t\}$, which may only be available through training and is often an approximation, and (2) it may be computationally intractable due to the large state space.

In the experiments below, we will assume $\{X_t\}$ to be first-order Markov, with which the above problem becomes a POMDP. This allows us to limit our attention to the class of Markov policies. We will further assume that X_t can only take on a finite number of values (i.e., soil moisture is quantized), to limit the state space. Specifically, the quantization levels used are given by $Q = [8, 9.5, 11, 12.5, 13.25, 14, 14.75, 15.5, 16, 17.5]$ (all in %).

With these assumptions, the soil moisture data is firstly quantized, and then the first T quantized values are used as a training set to generate a state transition matrix $\mathbb{P}$ that describes the evolution of the discrete-time discrete-valued process $\{X_t\}$. The POMDP problem defined in Eqn (14) is solved using Cassandra's pomdp-solve package [Monahan 1982]. The cost function $c(U_t)$ is set to 0.05, 0.10, 0.50 and 1.00 for $U_t = 1$ and 0 otherwise in different experiments, which is intended to control the sampling cost. The penalty $\rho(\cdot)$ is set to be the sum of absolute error, and the discount factor α is set to 0.99. An interested reader is referred to [Shuman et al. 2010] for more on parameter options. The amount of data in the training set (T) is set to 1000 and 3000, respectively. These are referred to as T1 and T2 in Figure 15, respectively. The solution to (14), in the form of a policy pair is then applied to the segment of garden data [3001, 5888], with the resulting estimation error reported in Figure 15 and Table V. Figure 16 further compares the CDF of the absolute estimated error (referred to as Error) for the same data set (Garden data, [3001, 5888]) using Φ_U, representation basis Ψ_D and solver SL0, with a window size of $N = 128$.

We see that compressive sensing based reconstruction performs significantly better than the closed-loop approach, under a range of cost levels and sampling rates (the rates cannot be perfectly controlled in the experiments, and are organized into groups of similar values). Furthermore, the open-loop approach improves much faster as the sampling rate increases. The reason for the performance difference, as well as the slow improvement of the closed-loop approach, lies in the fact that in order to be computationally tractable, we had to quantize the soil moisture in solving (14). As a result the output of the estimation/reconstruction is also quantized, which contributed to a significant part of the error; this phenomenon is also visible in Figure 15. We also examine



(a) Sampling Rate $\approx 51\%$ (b) Sampling Rate $\approx 41\%$

(c) Sampling Rate $\approx 18\%$ (d) Sampling Rate $\approx 13\%$

Fig. 15. Performance under different sampling rates - Part I: reconstructed signal.

Method	T	Sampling rate (%)	AvgError
$(\Phi_U, \Psi_D$, SL0)	0	51	0.018
$c(1) = 0.05$	1000	51	0.481
	3000	51	0.418
$(\Phi_U, \Psi_D$, SL0)	0	41	0.018
$c(1) = 0.10$	1000	47	0.523
	3000	38	0.460
$(\Phi_U, \Psi_D$, SL0)	0	19	0.040
$c(1) = 0.50$	1000	22	0.652
	3000	17	0.594
$(\Phi_U, \Psi_D$, SL0)	0	13	0.049
$c(1) = 1.00$	1000	16	0.691
	3000	11	0.646

Table V. Open-loop versus closed-loop.

the effect of combining closed-loop scheduling (the output scheduling policy g^* from (14)), and the reconstruction based on Ψ_D and SL0, i.e., to replace the measurement given by Φ with g^*. The comparison results over the same data segment are shown in Table VI.

We conclude that using closed-loop scheduling adds very little, if any at all, to the reconstruction accuracy. This shows that the compressive sensing method developed in

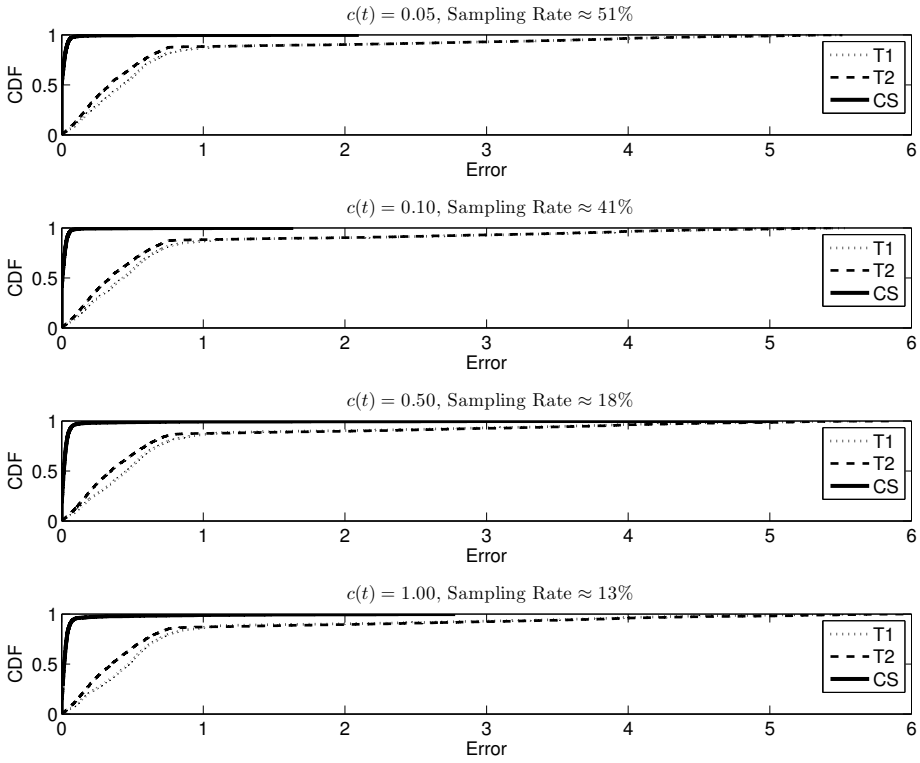


Fig. 16. Performance under different sampling rates - Part II: CDF of estimation error.

Method	T	Sampling rate (%)	$c(1)$	AvgError
$(\Phi_U, \Psi_D, \text{SL0})$	0	11	1.0	0.0763
$(\mathbf{g}^*, \Psi_D, \text{SL0})$	3000	11	1.0	0.0750
$(\Phi_U, \Psi_D, \text{SL0})$	0	17	0.5	0.0533
$(\mathbf{g}^*, \Psi_D, \text{SL0})$	3000	17	0.5	0.0607
$(\Phi_U, \Psi_D, \text{SL0})$	0	37	0.1	0.0241
$(\mathbf{g}^*, \Psi_D, \text{SL0})$	3000	37	0.1	0.0285
$(\Phi_U, \Psi_D, \text{SL0})$	0	51	1.0	0.0191
$(\mathbf{g}^*, \Psi_D, \text{SL0})$	3000	51	1.0	0.0195

Table VI. Closed-loop scheduling, open-loop estimate

the previous sections performs very well indeed, with relatively little room to improve. It also suggests that the combination of Ψ_D and SL0 is extremely robust to the type of measurement schedules used.

9. DISCUSSION AND CONCLUSION

We considered the problem of monitoring soil moisture evolution using a wireless network of in-situ sensors. We showed that at the cost of small estimation error we can significantly reduce energy consumption by taking a sparse set of measurements. We discussed how to apply sparse sensing techniques to achieve this, including the use of compressive sensing and matrix completion. We showed that both can be made to deliver very good estimation accuracy with no more than 10% of the standard sampling rate. To give a more concrete sense of how much this level of reduction in sampling

rate can ultimately contribute to the overall energy saving in the operation of an entire monitoring system, we quote below some quantities based on a soil moisture monitoring system Ripple-2 developed jointly at the University of Michigan and the University of Southern California. The design and development of this system have been partially documented in [Moghaddam et al. 2010b]. We estimate that the lifetime of a wireless node under our system implementation would be significantly increased, from around 6 months to nearly 5 years by sampling at an average of 100-min intervals compared to 10-min intervals.

We showed that MC performs slightly better than using CS independently at each location when multiple locations are involved, by exploiting not only temporal but also spatial correlation. However, we also showed that MC has two weaknesses: (1) the robust version of MC underperforms robust CS when outliers in measurements are present, and (2) MC requires measurement schedules at each location to be sufficiently randomized; this may be undesirable in practice, as synchronized schedules (especially uniform schedules is both easier to implement and more energy efficient from an operation standpoint. This is because the power consumption of the sensor module in an on state is much higher than in an off/sleep state, thus our desire to minimize the number of times to activate the sensors and to minimize the amount of time a sensor remains in the on state. If different sensors have different measurement schedules, they will also have different sleep schedules; it then becomes challenging to maintain connectivity, and in general sensors will need to be on for much longer periods of time because they may need to wait for other sensors to wake up and so on. Even if sensors only talk to a local coordinator and not to each other, using a synchronized schedule means that the coordinator needs to be on for much short periods of time, thus conserving energy.

ACKNOWLEDGMENTS

This work was carried out at the University of Michigan as part of the following three projects: Soil Moisture Smart Sensor Web Using Data Assimilation and Optimal Control (NASA grant NNX06AD47G), Ground Network De- sign And Dynamic Operation For Near Real-Time Validation of SpaceBorne Soil Moisture Measurements (NASA grant NNX09AE91G) and "Land Information System for SMAP and AirMOSS" (NASA grant NNX12AO54G), all through the Earth Science Technology Office, Advanced Information Systems Technologies program (AIST).The research of X. Wu is partially supported by NSFC under grant No. 61373146.

The authors would like to thank other members of the project team: Profs. M. Moghaddam, D. Teneketzis, and D. Entekhabi, and Y. Liu, H. Tavafoghi, Y. Ouyang, A. Silva, M. Burgin, D. Clewley, R. Akbar and A. Castillo. The authors would also like to thank the anonymous re- viewers for comments and suggestions that greatly helped improve the quality of the paper.

REFERENCES

I. F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. 2002. A Survey on Sensor Networks. *IEEE Communications Magazine* 40, 8 (August 2002), 102–114.

M. S. Andersland and D. Teneketzis. 1996. Measurement Scheduling for Recursive Team Estimation. *Journal of Optimization Theory and Applications* 89, 3 (April 1996), 615–636.

M. Athans. 1972. On the Determination of Optimal Costly Measurement Strategies for Linear Stochastic Systems. *Automatica* 8, 1972 (August 1972), 397–412.

J. S. Baras and A. Bensoussan. 1989. Optimal Sensor Scheduling in Nonlinear Filtering of Diffusion Processes. *SIAM Journal of Control and Optimization* 27, 4 (July 1989), 786–813.

D. Baron, M.B. Wakin, M.F. Duarte, S. Sarvotham, and R.G. Baraniuk. 2006. Distributed Compressed Sensing. *IEEE Transactions on Information Theory* 52, 12 (December 2006), 5406–5425.

Stephen Becker. 2012. Matrix Completion Solvers. (2012). http://www.ugcs.caltech.edu/~srbecker/wiki/Category:Matrix_Completion_Solvers

J. F. Cai, E. J. Candés, and Z. Sun. 2010. A Singular Value Thresholding Algorithm for Matrix Completion. *SIAM Journal on Optimization* 20, 4 (2010), 1956–1982.

E. Candés and T. Tao. 2006. Near Optimal Signal Recovery from Random Projections: Universal Encoding Strategies? *IEEE Transactions on Information Theory* 52, 12 (December 2006), 5406–5425.

Emmanuel J. Candés. 2006. Compressive Sampling. In *Proceedings of the International Congress of Mathematicians*. 265–272.

Emmanel J. Candés and Benjamin Recht. 2009. Exact Matrix Completion via Convex Programming. *Foundations of Computational Mathematics* 9, 6 (2009), 717–772.

E. J. Candés, J. Romberg, and T. Tao. 2006. Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information. *IEEE Transactions on Information Theory* 52, 4 (February 2006), 489–509.

E. J. Candés and T. Tao. 2005. Decoding by Linear Programming. *IEEE Transactions Information Theory* 51, 12 (December 2005), 4203–4215.

Aldrich Castillo. 2010. *Parallelizing the Distributed Hydrologic Model MOBIDIC*. Technical Report. Environmental Engineering,Massachusetts Institute of Technology.

V. Chandrasekaran, S. Sanghavi, P. Parrilo, and A. Willsky. 2011. Rank-Sparsity Incoherence for Matrix Decomposition. *SIAM Journal on Optimization* 21, 2 (2011), 572–596.

Yudong Chen, A. Jalali, S. Sanghavi, and C. Caramanis. 2011a. Low-rank Matrix Recovery from Errors and Erasures. In *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*. 2313–2317.

Yudong Chen, Huan Xu, Constantine Caramanis, and Sujay Sanghavi. 2011b. Robust matrix completion with corrupted columns. In *Proceedings of the 28th International Conference on Machine Learning*. 873–880.

J. Cheng, W. Willinger, and L. Qiu. 2012. Spatio-Temporal Compressive Sensing and Internet Traffic Matrices (Extended Version). *IEEE/ACM Transactions on Networking* 20, 3 (2012), 662–676.

J. Cheng, Q. Ye, H. Jiang, and D. Wang. 2013. STCDG: An Efficient Data Gathering Algorithm Based on Matrix Completion for Wireless Sensor Networks. *IEEE Transactions on Wireless Communications* 12, 2 (2013), 856–861.

Wei Dai, Olgica Milenkovic, and Ely Kerman. 2011. Subspace Evolution and Transfer (SET) for Low-Rank Matrix Completion. (July 2011). http://arxiv.org/pdf/1006.2195.pdf

Amol Deshpande, Carlos Guestrin, Samuel R. Madden, Joseph M. Hellerstein, and Wei Hong. 2004. Model-driven Data Acquisition in Sensor Networks. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30 (VLDB '04)*. VLDB Endowment, 588–599.

D. Donoho. 2006. Compressed Sensing. *IEEE Transactions on Information Theory* 52, 2 (February 2006), 4036–4048.

D. L. Donoho. 2004. For Most Large Underdetermined Systems of Equations, the Minimal l1-norm Near-Solution Approximates the Sparsest Near-Solution. Technology Report. (2004).

D. L. Donoho, M. Elad, and V. Temlyakov. 2006. Stable Recovery of Sparse Overcomplete Representations in the Presence of Noise. *IEEE Transactions Information Theory* 52, 1 (January 2006), 6–18.

J. Evans and V. Krishnamurthy. 2001. Optimal Sensor Scheduling for Hidden Markov Model State Estimation. *Internat. J. Control* 74, 18 (December 2001), 1737–1742.

D. Goldfarb and S. Ma. 2011. Convergence of Fixed-point Continuation Algorithms for Matrix Rank Minimization. *Foundations of Computational Mathematics* 11, 2 (2011), 183–210.

I. F. Gorodnitsky and B. D. Rao. 1997. Sparse Signal Reconstruction from Limited Data using FOCUSS, a Re-weighted Minimum Norm Algorithm. *IEEE Transactions on Signal Processing* 45, 3 (March 1997), 600–616.

Sina Jafarpour, Weiyu Xu, B. Hassibi, and R. Calderbank. 2009. Efficient and Robust Compressed Sensing Using Optimized Expander Graphs. *Information Theory, IEEE Transactions on* 55, 9 (Sept 2009), 4299–4308.

P. Jain, R. Meka, and I. Dhillont. 2010. Guaranteed Rank Minimization via Singular Value Projection. In *Proceedings of the Neural Information Processing Systems Conference (NIPS)*. 937–945.

X. Ji, Y. He, J. Wang, W. Dong, X. Wu, and Y. Liu. 2014. Walking down the STAIRS: Efficient Collision Resolution for Wireless Sensor Networks. In *The 33th IEEE International Conference on Computer Communications (NFOCOM 2014)*.

R.H. Keshavan and S. Oh. 2009. Optspace: A Gradient Descent Algorithm on the Grassman Manifold for Matrix Completion. (2009). http://arxiv.org/abs/0910.5260v2/

Andreas Krause, Ajit Singh, and Carlos Guestrin. 2008. Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. *J. Mach. Learn. Res.* 9 (June 2008), 235–284.

V. Krishnamurthy. 2001. Algorithms for Optimal Scheduling and Management of Hidden Markov Model Sensors. *IEEE Transactions on Signal Processing* 50, 6 (June 2001), 1382–1397.

l1-magic 2005. l1-magic: Recovery of Sparse Signals Via Convex Programming. (2005). www.acm.caltech.edu/l1magic/downloads/l1magic.pdf.

A. Lakhina, M. Crovella, and C. Diot. 2004. Diagnosing Network-wide Traffic Anomalies. In *ACM SIGCOMM 04*. 219?30.

Y. Li, L. W. Krakow, E. K. P. Chong, and K. N. Groom. 2009. Approximate Stochastic Dynamic Programming for Sensor Scheduling to Track Multiple Targets. *Digital Signal Processing* 19, 6 (December 2009), 978–989.

C. Luo, F. Wu, C. W. Chen, and J. Sun. 2009. Compressive Data Gathering for Large-Scale Wireless Sensor Networks. In *Proceedings of the 15th annual international conference on Mobile computing and networking*. 145–156.

Angshul Majumdar. 2010. Matlab Source Code. (2010). http://www.mathworks.com/matlabcentral/fileexchange/26395-matrix-completion-via-thresholding

M. Moghaddam, D. Entekhabi, Y. Goykhman, Ke Li, Mingyan Liu, A. Mahajan, A. Nayyar, D. Shuman, and D. Teneketzis. 2010a. A Wireless Soil Moisture Smart Sensor Web Using Physics-Based Optimal Control: Concept and Initial Demonstrations. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of* 3, 4 (Dec 2010), 522–535.

M. Moghaddam, X. Wu, M. Burgin, A. Castillo, D. Entekhabi, Y. Goykhman, K. Li, M. Liu, M. Liu, A. Nayyar, A. Silva, Q. Wang, and D. Teneketzis. 2010b. Soil Moisture Sensing Controller And optimal Estimator (SoilSCAPE): First Deployment of the Wireless Sensor Network and Latest Progress on Soil Moisture Satellite Retrieval Validation Strategies. In *Earth Science Technology Forum (ESTF2010)*.

Hosein Mohimani, Massoud Babaie-Zadeh, and Christian Jutten. 2009. A Fast Approach for Overcomplete Sparse Decomposition Based on Smoothed L0 Norm. *IEEE Transactions On Signal Processing* 57, 1 (January 2009), 289–301.

G. E. Monahan. 1982. A Survey of Partially Observable Markov Decision Orocesses: Theory, Models, and Algorithms. *Management Science* 28, 1 (January 1982), 1–16.

NASASP 2006. NASA Strategic Plan. (2006). http://www.nasa.gov/

D. Needell and R. Vershynin. 2010. Signal Recovery from Inaccurate and Incomplete Measurements via Regularized Orthogonal Matching Pursuit. *IEEE Journal of Selected Topics in Signal Processing* 4, 2 (February 2010), 310–316.

G. Quer, R. Masiero, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi. 2009. On the Interplay Between Routing and Signal Representation for Compressive Sensing in Wireless Sensor Networks. In *Information Theory and Applications Workshop (ITA 2009)*. 206–215.

Nithya Ramanathan and Benjamin Recht. 2009. Suelo: Human-assisted Sensing for Exploratory Soil Monitoring Studies. In *SenSys'09*. 197–210.

David Shuman, Ashutosh Nayyar, Aditya Mahajan, Yuriy Goykhman, Ke Li, Mingyan Liu, Demosthenis Teneketzis, Mahta Moghaddam, and Dara Entekhab. 2010. Measurement Scheduling for Soil Moisture Sensing: From Physical Models to Optimal Control. *Proceedings of the IEEE Special Issue on Sensor Networks and Applications* 98, 11 (November 2010), 1918–1933.

SL0 2008. SL0. (2008). http://ee.sharif.ir/~SLzero/

Sparse Lab 2007. Sparse Lab. (2007). http://sparselab.stanford.edu/

Z. Tian and G. Giannakis. 2007. Compressed Sensing for Wide band Cognitive Radios. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 1357–1360.

Joel A. Tropp and Anna C. Gilbert. 2007. Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit . *IEEE Transactions on Information Theory* 53, 12 (December 2007), 4655–4666.

J. Wang, Y. Liu, M. Li, M. Rossi, W. Dong, and Y. He. 2011. Qof: Towards comprehensive path quality measurement in wireless sensor networks.. In *The 30th IEEE International Conference on Computer Communications (NFOCOM 2011)*. 775–783.

Andrew E. Waters, Aswin C. Sankaranarayanan, and Richard Baraniuk. 2011. SpaRCS: Recovering low-rank and sparse matrices from compressive measurements. In *Advances in Neural Information Processing Systems*. 1089–1097.

X. Wu and M. Liu. 2012. In-Situ Soil Moisture Sensing: Optimal Sensor Placement and Field Estimation. *ACM Transactions on Sensor Network* 8, 4 (2012), 1–33.

X. Wu, Y. Wu, M. Liu, and L. Zheng. 2011. In-Situ Soil Moisture Sensing: Efficient Random Sensor Placement and Field Estimation using Compressive Sensing. In *Proceedings of International Conference on Wireless communications, networking and mobile computation*. 1–6.

Received May 2013; revised February 2014; accepted May 2014