

Decentralised Control of Adaptive Sampling in Wireless Sensor Networks

JOHNSEN KHO, ALEX ROGERS, and NICHOLAS R. JENNINGS

University of Southampton

The efficient allocation of the limited energy resources of a wireless sensor network in a way that maximises the information value of the data collected is a significant research challenge. Within this context, this paper concentrates on adaptive sampling as a means of focusing a sensor's energy consumption on obtaining the most important data. Specifically, we develop a principled information metric based upon Fisher information and Gaussian process regression that allows the information content of a sensor's observations to be expressed. We then use this metric to derive three novel decentralised control algorithms for information-based adaptive sampling which represent a trade-off in computational cost and optimality. These algorithms are evaluated in the context of a deployed sensor network in the domain of flood monitoring. The most computationally efficient of the three is shown to increase the value of information gathered by approximately 83%, 27%, and 8% per day compared to benchmarks that sample in a naïve non-adaptive manner, in a uniform non-adaptive manner, and using a state-of-the-art adaptive sampling heuristic (USAC) correspondingly. Moreover, our algorithm collects information whose total value is approximately 75% of the optimal solution (which requires an exponential, and thus impractical, amount of time to compute).

Categories and Subject Descriptors: H.1.1 [**Models and Principles**]: Systems and Information Theory—*Information theory, Value of information*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents, Multiagent systems*

General Terms: Algorithms, Management, Measurement.

Additional Key Words and Phrases: Adaptive sampling algorithm, decentralised decision mechanism, gaussian process regression, information metric.

1. INTRODUCTION

Wireless sensor networks (WSN) have recently generated significant research interest within the academic literature of computer science and electronic engineering. Networks of battery-powered sensor nodes, wirelessly communicating information sampled from the environment to a base-station, have many advantages over their wired counterparts, and they have been demonstrated in applications ranging from

Author's address: School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK; email: {jk05r,acr,nrj}@ecs.soton.ac.uk.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 20YY ACM 0000-0000/20YY/0000-0001 \$5.00

environment and habitat monitoring [De Roure 2005; Padhy et al. 2005; Werner-Allen et al. 2006; Mainwaring et al. 2002; Cardell-Oliver et al. 2005], smart buildings [Guestrin et al. 2005], object tracking [Ledeczi et al. 2005; He et al. 2006], structural health surveillance [Chintalapudi et al. 2006], to other security and health related applications [Kroc and Delic 2003; Lo and Yang 2005].

A key requirement within all of these applications is effective energy management, and this is often addressed through adapting the sensing (or sampling) policies of the sensor nodes. Other energy management techniques in the literature include data compression [Kimura and Latifi 2005] and data fusion (i.e. data aggregation) [Chu et al. 2002; Makarenko and Durrant-Whyte 2004]. However, the effectiveness of the compression techniques is highly dependent on the processing power of the nodes. Here, the total computational overhead increases as both the source and destination node, now, have to spend some processing energy in order to compress and decompress data accordingly. The data fusion technique, however, is typically applied for applications that explicitly consider the spatial correlations of sensor nodes (not the temporal correlations as we have here). Given this and the fact that our target applications have limited resources at each node, we focus on the sampling option.

Sampling policies generally describe a node’s sampling rate (i.e. *how often* a node is required to sample during a particular time interval) and schedule (i.e. *when* a node is required to sample), and much recent work has explored decentralised algorithms that enable the sensor nodes to autonomously adapt and adjust their own sensing policies (see Section 7 for more details). Such solutions are attractive in our context since they remove the bottleneck of a central decision maker (and the need to inform this decision maker of the energy state of each sensor node), and they fully exploit the ever increasing computational capacity of the sensor nodes themselves [Heeks 1999]. Furthermore, they are also more robust than centralised alternatives since there is no single point of failure, and even in the case that communication with the base-station fails (perhaps due to the failure of a node on a multi-hop route to the base-station), the sensor nodes are able to continue to autonomously operate in the absence of any external direction until communication is restored.

To date, such decentralised algorithms have typically been applied to sensor networks deployed for environmental monitoring, and they have specifically considered networks composed of battery powered nodes that exhibit finite lifetimes. Since a sensor node sampling at its maximum rate would deplete its battery in a short period of time, effective sensing policies in this context seek to balance the lifetime of the sensor network as a whole against the value of the information that it collects. To do so, they typically invoke domain specific heuristics that depend upon one or more user specified parameters. For example, the USAC algorithm of Padhy et al. [2006], which is representative of the state-of-the-art in this area, models temporal variations in the environmental parameter being sensed as a piece-wise linear function, and uses a pre-specified *confidence interval* parameter in order to make real-time decisions regarding the sampling rate of the sensor nodes.

However, in many applications, sensor nodes are also capable of harvesting energy from their local environment through different sources (e.g. solar power, wind

energy, or vibration energy). In such cases, additional operating modes become possible, and a common alternative to that described above is to require that the sensors maintain *energy neutral* operation; balancing energy harvesting against energy consumption, in order that they exhibit an indefinite lifetime [Kansal et al. 2007]. In this context, an effective sensing policy must maximise the information that a sensor collects over a particular time interval subject to energy constraints, and this typically involves planning exactly when, within the specified time interval, to take a constrained number of samples. To actually achieve this within a general setting without resorting to domain specific heuristic requires that (i) we can predict the information content of a sensor’s future samples, given a particular sampling schedule, and (ii) that we can then optimise this sampling schedule, subject to energy constraints, in order to maximise the information that will be collected by a sensor node over a particular time interval.

Thus, against this background, in this paper we address these two complementary challenges. In particular, we describe a principled information measure based upon Fisher information and Gaussian process regression, and we present three decentralised algorithms (representing a trade-off in computational cost and optimality) that allow individual sensors to maximise this information measure given their individual energy constraints. In more detail, we make the following contributions:

- We develop a novel generic information metric for sensor networks. This metric represents the temporal variation in the environmental parameter being sensed as an unknown function, and then uses Gaussian process (GP) regression to infer the characteristics (specifically its temporal correlation and periodicity) and value of this function, over a continuous interval, conditioned on samples made at discrete times within the interval. We then use the mean Fisher information over the entire interval (including periods between which sensor samples were taken) as a measure of the information content of these actual sensor samples. Thus, informative sensor samples are those that minimise uncertainty in the value of the environmental parameter over the entire interval¹.
- Using this information metric, we describe three novel decentralised control algorithms for information-based adaptive sampling which represent a trade-off in computational cost and optimality. The first uses GP regression within each sensor node to optimise (using the metric described above) the time at which a constrained number of future sensor readings should be taken. This process is exponential in the number of sensor readings taken, and thus, the second algorithm we present again uses GP regression within the sensor nodes, but performs a greedy approximate optimisation in order that it is more computationally tractable. Finally, we further reduce the computational cost by using a heuristic algorithm within each sensor node, rather than the GP regression, in order to select the times at which future sensor readings should be taken.
- In order to ground and evaluate this approach, we need to exercise it in a particular domain and here we choose flood monitoring and, in particular, the

¹Note that this is somewhat similar to the use of mutual information by Krause et al. [2006] to select the most informative subset of sensor placements from an initial trial deployment, but here we consider temporal rather than spatial correlations (see Section 7 for more details).



Fig. 1. (a) The deployed FLOODNET system in the River Crouch, East Essex in eastern England. (b) A FLOODNET node on site.

FLOODNET sensor network (more details of which can be found in Section 2). In this setting, the heuristic algorithm is empirically shown to increase the value of information gathered by approximately 83%, 27%, and 8% per day compared to benchmarks that sample in a naïve non-adaptive manner, in a uniform non-adaptive manner, and using a state-of-the-art adaptive sampling heuristic (USAC), correspondingly. Furthermore, it provides information whose total value is approximately 75% the optimal solution (which requires an exponential, and thus impractical, amount of time to compute).

The rest of the article is organized as follows. Section 2 briefly describes the FLOODNET domain. In Sections 3 and 4 respectively, we detail the sampling problem we face in a general manner and how we use a GP package to calculate our information metric. Section 5 formulates the three decentralised control algorithms for adaptive sampling. Their performances are then empirically evaluated against a number of benchmarks in Section 6. We then outline the main types of existing adaptive sampling algorithms in the literature in Section 7 and highlight their limitations. Finally, conclusions and areas of future work are discussed in Section 8.

2. THE FLOODNET SENSOR NETWORK

To illustrate our methods, we choose flood detection as our target domain and, given its accessibility, we choose FLOODNET as the specific sensor network [De Roure 2005]. The ultimate aim of FLOODNET is to provide early warning of flooding such that actions can be taken to alleviate risks to people and property. To this end, it is currently deployed to gather precise tide height readings to enable a calibrated hydrological model of the deployment area to be constructed. The network must withstand long term unmanned operation without any significant human intervention as nodes are deployed at a number of hostile and not easily accessible locations where periodic data collections also might not be possible particularly in extreme

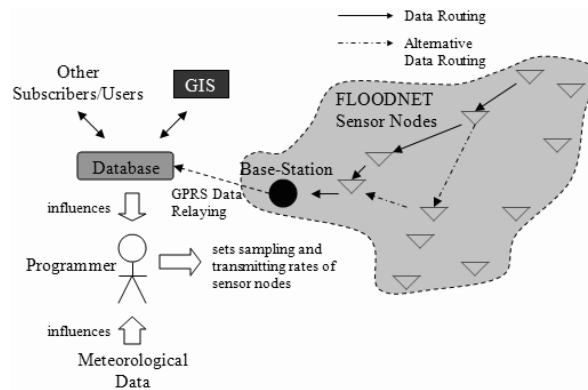


Fig. 2. The centralised FLOODNET infrastructure.

environment conditions (for instance during floods). The network, thus, incurs less costs in terms of setup, operation, and communication compared to equivalent data logging devices.

As can be seen from Figure 1(a), FLOODNET consists of twelve nodes. Each node, shown in Figure 1(b), is based around a BitsyX Single Board Computer (SBC). Since the SBC consumes a significant amount of power (1000mW) when providing field processing capabilities, it is in sleep mode for most of the time. A wireless LAN PC card is used to send and receive data wirelessly from the neighbourhood nodes (requiring an additional 910mW and 640mW of power respectively). Tide height measurements are made with a water-depth transducer sensor module that consumes 70mW of power when activated, and each node is equipped with a rechargeable lead-acid battery and a solar panel, such that it can harvest solar energy to recharge the battery during the day.

Like many other similar applications, FLOODNET currently adopts a centralised regime to control its system whereby each FLOODNET node is centrally programmed to have fixed sampling and transmission rates. Each FLOODNET node currently takes samples and stores them locally on the memory at five minute intervals, and activates the SBC with the corresponding transceiver modules every two hours for the purpose of transmitting the collected data to the base-station via multi-hop routing. The base-station subsequently relays the data to a Geographical Information System (GIS) database using General Packet Radio Service (GPRS). Scientists from the Geography Department of the University of Southampton then use this real-time incoming data within a hydraulic prediction model to make accurate and timely flood forecasts.

The incoming sensed data (together with meteorological data) also influences the programmer in setting the nodes' next sampling and transmission rates (see Figure 2). Our ultimate aim is to remove this centralised point of control (for the reasons described in Section 1), and deploy our decentralised information-based adaptive sampling algorithms within the FLOODNET nodes themselves.

3. PROBLEM DESCRIPTION

We now formalise a description of the generic sampling problem that we face (of which FLOODNET is but one specific instance). To this end, let n be the number of sensors within a sensor network system and the set of all sensors be $I = \{i_1, i_2, \dots, i_n\}$. The sensor network is tasked with monitoring some environmental parameter over multiple days, and we divide the day into a fixed number of time slots, and denote these time slots by the set $H = \{1, \dots, w\}$.

Each sensor $i \in I$ can sample at s different rates. The set of possible sampling rates is denoted by $C^i = \{c_1^i, c_2^i, \dots, c_s^i\}$ where $C^i \subseteq \mathbb{Z}^+$ and $c_1^i < c_2^i < \dots < c_s^i$. Specifically, each element of this set, c_k^i , is a positive integer that describes the number of times that the sensor samples during a time slot.

An algorithm (that we devise within this paper) determines the actual sampling rate that each sensor should use within any specific time slot. Thus, each sensor $i \in I$, has an allocated set of sampling actions (i.e. sampling schedules) for each day denoted by $Alloc^i = \{a_1^i, a_2^i, \dots, a_w^i\}$, where $a_k^i \in C^i, \forall k \in H$. Thus, any element, $a_k^i \in Alloc^i$, represents the number of times that the sensor should sample within any specific time slot within the day. Hence, at the end of a day, sensor i will have collected a set of observations, Y^i , at a corresponding set of sampling points, X^i , such that $|X^i| = |Y^i| = \sum_{k=1}^w a_k^i$.

In general, the sensors within the network will deplete their energy resources at different rates since they will have different sampling schedules. Assuming that the remaining battery power available for sampling for sensor i at the beginning of a day is E_r^i , and it requires a certain amount of energy e_s to sample an event, we must ensure that any set of sampling actions satisfies:

$$\sum_{k=1}^w a_k^i e_s \leq E_r^i \quad (1)$$

such that the sum of all the energy required to do the sampling actions on that day must not exceed the remaining battery power. Note that our choice of imposing the energy constraint over a 24 hour period is a natural one since it represents a daily cycle in which the sensor node recharges its battery during daylight, and gradually depletes it during the night. Furthermore, note that we do not include the constant transmitting and receiving variables into the equation since sensors transmit their recorded readings in every two hour period to the base-station using a non-adaptive multi-hop routing method (with a routing table created at system start up time)².

A sensor's preferences express the satisfaction of any particular action when faced with a choice between different alternatives. In our case, the actions correspond to the different sampling rates that a sensor may choose to perform within any particular time slot, and the preferences express the information content of the data collected by performing the corresponding actions. A preference structure brings together all the alternatives, V , and represents a sensor's preferences over the set of possible outcomes. Now, there are several choices that can be made regarding

²Note that we also assume that the communication costs do not depend on the quantity of samples taken. However, more complex relationships can also be modelled; the algorithms that we shall consider are constrained to take a maximum number of samples within a 24 hour period, and the details of how this maximum number is calculated are not restricted in any way.

the definition of a mathematical model for preference structures (see Chevalayre et al. [2006] for a review), but here we choose a simple cardinal structure since it allows a sensor to make individual comparisons between its sampling actions (i.e. whether sensor i is obtaining greater information value by sampling at rate c_u^i than c_z^i). Furthermore, in future work it will also allow comparisons between multiple sensors (i.e. whether sensor i is obtaining a greater information value by sampling at rate c_u^i than sensor j operating at c_r^j).

In more detail, a cardinal preference structure consists of a valuation function (i.e. a utility function or a mathematical function used to calculate the value or goodness of a certain action taken by nodes) given by $v : V \rightarrow Val$, where Val is a set of numerical values (typically, N , \mathbb{R} , $[0, 1]$, or \mathbb{R}^+).

4. INFORMATION METRIC

Building on the problem description above, an algorithm needs a way to value the various observations that the sensors may make. Within the data fusion and tracking literature, where spatially correlated sensor readings typically represent the estimated position of a target, there are a number of standard techniques for doing this. Most common, is the use of Fisher information, whereby the estimated position of the target is represented as a multi-dimensional probability distribution, and Fisher information is used to quantify the uncertainty represented by this distribution [Bar-Shalom et al. 2001; Chu et al. 2002; Frieden 2004; Zhao and Guibas 2004].

In this work, we follow a similar procedure. If at any point in time, we are able to calculate an estimate of the value of the environmental parameter being sensed, and this estimate is represented by a predictive distribution with mean, $\hat{\mu}(t)$, and variance, $\hat{\sigma}^2(t)$, then the mean Fisher information over any period of time between t_1 and t_2 is given by:

$$FI = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \frac{1}{\hat{\sigma}^2(t)} dt \quad (2)$$

The estimated value of the environmental parameter between times t_1 and t_2 is informed by the samples that the sensor actually takes, and in the next section we specifically describe how we can perform this estimation in a principled Bayesian framework using Gaussian process regression.

Finally, we note that we consider the value of the information collected by the sensor network as a whole to simply be the sum of the information collected by each individual sensor, and thus, we are explicitly not considering correlations between different sensors. Relaxing this assumption is a focus of our future work, and we discuss it in more detail in Section 8.

4.1 Gaussian Process Regression

As described above, in order to calculate the mean Fisher information, we must use the actual (and possibly noisy) samples taken by the sensor to estimate the value of the environmental parameter being sensed over a continuous period of time (including times between those at which samples were actually taken). Furthermore, this estimate must represent a full predictive distribution with both a mean and a variance. Hence, we use Gaussian process (GP) regression to generate this estimate.

This principled approach allows us to perform Bayesian inference about functions; in our case, the function representing the value of the environmental parameter over time [Rasmussen and Williams 2006]. Such techniques have a long history in geospatial statistics [Cressie 1991], and more recently have been used as a generic non-parametric probabilistic model for spatially correlated phenomena [Guestrin et al. 2005; Ertin 2007]. In contrast, here we use it with temporally correlated sensor samples.

In more detail, a GP regression takes as inputs a set of g training or sampling points, $X = \{x_1, x_2, \dots, x_g\}$, and a set of g noisy observations or target values, $Y = \{y_1, y_2, \dots, y_g\}$ [Mackay 1998; Seeger 2004]. Given a covariance function that describes the correlation between sensor readings at any two times (we shall discuss this function in more detail later), the GP is able to infer the posterior predictive distribution (i.e. the conditional distribution available after the GP has observed the training set, the target set, and the covariance function) of the environmental parameter at any other set of n test points, $X' = \{x'_1, x'_2, \dots, x'_n\}$. This predictive distribution is represented by a mean, $\hat{\mu}(X') = \{\hat{\mu}(x'_1), \hat{\mu}(x'_2), \dots, \hat{\mu}(x'_n)\}$, and a variance, $\hat{\sigma}^2(X') = \{\hat{\sigma}^2(x'_1), \hat{\sigma}^2(x'_2), \dots, \hat{\sigma}^2(x'_n)\}$, given by:

$$\hat{\mu}(x'_i) = k^T (K + \sigma^2 I_n)^{-1} Y \quad (3)$$

$$\hat{\sigma}^2(x'_i) = C(x'_i, x'_i) - k^T (K + \sigma^2 I_n)^{-1} k \quad (4)$$

where K is a $g \times g$ matrix for the training set covariances, k is a $g \times 1$ vector identifying the training-test set covariances (i.e. a row vector of the covariances of x'_i with all variables in X), $C(x'_i, x'_i)$ is the covariance of x'_i , I_g is a $g \times g$ identity matrix, and σ^2 is added Gaussian noise of the training set accordingly.

The variance is then used to calculate the mean Fisher information of the interval spanned by X' of n test points. Thus, the mean Fisher information, over the interval X' , conditioned on the set of observations represented by the sensor readings, Y , taken at times X , is given by:

$$FI(X) = \frac{1}{n} \sum_{i=1}^n \frac{1}{\hat{\sigma}^2(x'_i)} \quad (5)$$

Note that this is a discretization of Equation 2, whose resolution is determined by the number of prediction points that cover the period of interest (i.e. by the value of n). Furthermore, note that the value of Fisher information calculated above does not depend directly on the actual samples (since there is no dependence on Y in the expression for $\hat{\sigma}^2(x'_i)$ in Equation 3). Finally, we remark that the use of the notation, $FI(X)$, will become clear in Section 5.1 when we consider the calculation of the Fisher information metric using just a subset of the samples in X .

4.2 Covariance Functions

A key assumption of the GP regression technique described above is that points in time within X' that are close together are likely to have similar predicted values within $\hat{\mu}(X')$. Furthermore, training points in X which are close to estimation points in X' are those that are most informative. This notion of closeness or similarity is defined by a covariance function. The covariance function is a crucial ingredient within GP regression. It allows prior information concerning the domain

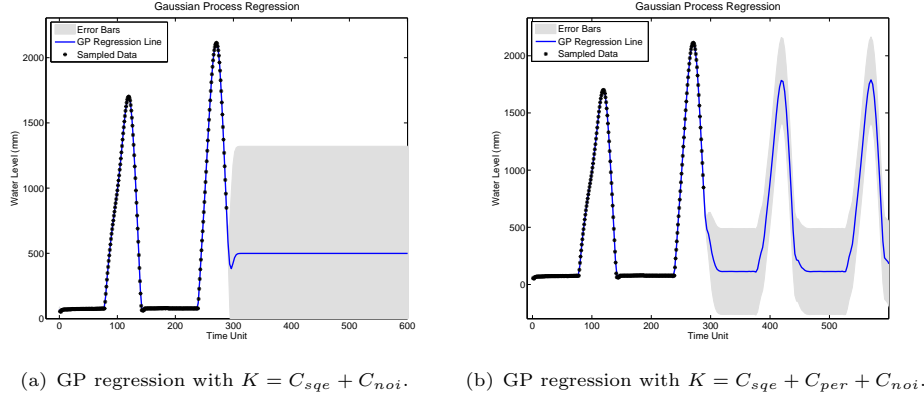


Fig. 3. Gaussian Process regression applied to FLOODNET water level data using covariance functions with and without period terms.

problem to be incorporated into the inference (for instance that the environmental parameter being sensed varies smoothly over time and/or is periodic), and thus, it influences the quality of the predictions made. While much empirical guidance for the choice of covariance functions does exist, there is no formal methodology for determining this choice automatically [Rasmussen 2004].

In our case, we choose a commonly used covariance function termed *squared exponential* or Gaussian covariance function:

$$C_{sqe}(x, x') = v_{sqe} \cdot \exp \left[-\frac{(x - x')^2}{\Lambda_{sqe}^2} \right] \quad (6)$$

where v_{sqe} is the weighting of this term, and Λ_{sqe} is the length scale which is the correlation length that represents the length along which successive target values are strongly correlated. We choose this because it is infinitely differentiable, and thus, capable of modelling smoothly varying environmental parameters. Moreover, Girard [2004] shows that this covariance function has good general modelling abilities and predictive performance comparable with that of neural networks. However, this choice is not fundamental to our algorithms and alternatives such as *rational quadratic*, *linear*, or *exponential* could also be used in other cases. Furthermore, these alternative forms can also be combined together (by summation or multiplication) to derive a rich family of possible covariance functions³.

Since many WSNs monitoring environmental phenomena show a periodical pattern between days in their readings (as we have with FLOODNET tide data), we also use a *periodic* covariance function:

$$C_{per}(x, x') = v_{per} \cdot \exp \left[-\frac{2 \sin^2 \left(\frac{x - x'}{p} \right)}{\Lambda_{per}^2} \right] \quad (7)$$

³In general, if we use a covariance function with sufficient flexibility, then we can be confident that the machinery of Bayesian inference will automatically reduce the weighting of terms that are not necessary to describe the data that has been observed.

Parameter	Value
Variance for C_{sqe} (v_{sqe})	log 112.2733
Variance for C_{per} (v_{per})	log 956.3368
Variance for C_{noi} (v_{noi})	log 3.3753
Correlation length scale for C_{sqe} (Λ_{sqe})	log 19.4994
Correlation length scale for C_{per} (Λ_{per})	log 0.7882
Unit period scale for C_{per} (p)	log 736.3052

Table I. Example hyperparameters for the covariance functions learned from FLOODNET data.

where v_{per} is the weighting of this term, p is the periodicity of the data, and Λ_{per} is the length scale. In addition, an *independent* covariance function with weighting v_{noi} is used to represent Gaussian distributed noise in sensor readings:

$$C_{noi}(x, x') = v_{noi} \cdot \kappa, \text{ where } \kappa = \begin{cases} 1 & \text{if } x = x' \\ 0 & \text{if } x \neq x' \end{cases} \quad (8)$$

These three separate terms are combined by simply adding them together, and this combination is shown to provide accurate water level estimates for the FLOODNET domain. For example, Figure 3 illustrates this by comparing (a) the case where $K = C_{sqe} + C_{noi}$ which excludes the periodic term, and (b) where $K = C_{sqe} + C_{per} + C_{noi}$ which includes it. In both cases, the markers represent the sensor samples, the solid line indicates the mean of the predictive distribution, and the shaded area represents its variance.

The weightings, v_{sqe} , v_{per} and v_{noi} , characteristic lengths, Λ_{sqe} and Λ_{per} , periodicity, p , and sensor noise, σ^2 , are collectively termed hyperparameters, and in general, we do not know their values *a priori*. However, a number of techniques can be used to infer their values from the sensor readings themselves. Within the GPML⁴ package that we use here, techniques for learning the hyperparameters are based on maximisation of the log likelihood function using an efficient conjugate gradient-based optimization algorithm [Bishop 2006]. However, there is no guarantee that the marginal log likelihood does not suffer from multiple local optima. Thus, we use a multi-start process for setting good initial hyperparameters prior whereby we restart the maximisation of log likelihood from a number of different starting points, and select the one that results in the maximum log likelihood⁵. We perform this learning prior to performing regression whenever new data is present, and Table I shows an example of these hyperparameters.

Within the FLOODNET domain, whenever the water level raw data points are closely related (i.e. they have a small covariance matrix or they are more frequently sampled), the variances of the estimated values, $\hat{\sigma}^2(X')$, will decrease. The Fisher information value will, on the other hand, increase as it is the inverse uncertainty of the estimate (see Figure 4). To illustrate this, consider the scenario illustrated in Figure 5. Assume one of the FLOODNET nodes has a set of twelve noisy measurements per hour for a given day. Thus, the node samples at five minute intervals such that the total number of samples (g) on that day is 288 (12 samples/hour x 24 hours). Now, in order to find out the value of this set of data, the node performs

⁴<http://www.gaussianprocess.org/gpml/>

⁵In future work we intend to investigate the use of fully Bayesian approaches to maintain a distribution over possible hyperparameter values [Osborne et al. 2008].

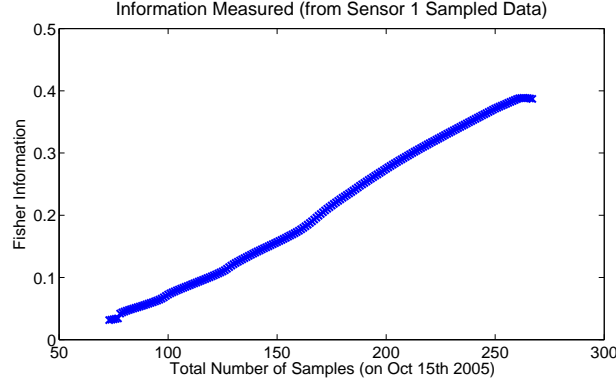
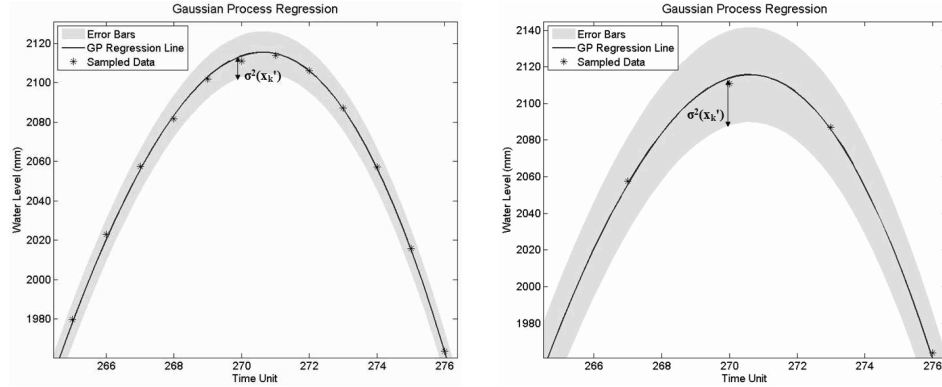


Fig. 4. Fisher information value (FI) gathered over one day period plotted against the number of samples. As expected, the value of information generally increases (i.e. we are more certain in the value of the environmental parameter being sensed) as the sensor nodes takes more samples. Note that in the case of a GP with fixed hyperparameters, we would expect to observe a concave function since the observations are correlated.



(a) Regression on sampled data taken between time unit 265 and 276, with 12 samples. Sensor 1 collects Fisher information value of $FI = 0.1263$.

(b) Regression on sampled data taken between time unit 265 and 276, with 4 samples. Sensor 1 collects Fisher information value of $FI = 0.5168 \cdot 10^{-1}$.

Fig. 5. GP regression example using sampled data collected from FLOODNET sensor 1 operating on Oct 14th 2005. One time unit represents a five minute interval.

the GP regression utilizing the training and target sets (sets X and Y correspondingly) to produce the predictive distribution with mean $\hat{\mu}(X')$ and variance $\hat{\sigma}^2(X')$. Given this, Equation 5 can then be used to determine the information content of the samples.

5. DECENTRALISED INFORMATION-BASED ADAPTIVE SAMPLING

Given the problem description and information metric, the objective in this work is to now derive an algorithm that can automatically determine the allocation of actions each day, $Alloc$, that will maximise the total mean Fisher information col-

lected by the sensors (i.e. the utilitarian social welfare), subject to the energy constraint (in Equation 1). To this end, we now present three novel decentralised control algorithms for information-based adaptive sampling that achieve this. Each algorithm represents a different trade-off between computational cost and optimality. We start with an exponential algorithm that maximises the Fisher information metric by performing GP regression on the individual sensor nodes, and progress to a more computationally efficient algorithm that uses a heuristic approach, rather than the GP regression, in order to determine the allocation of actions each day.

Each of these algorithms follows the same broad pattern. On any specific day, each sensor $i \in I$ may be in one of two modes: an updating mode in which the sensor samples at a predefined maximum rate throughout all time slots, or a standard sampling mode in which it samples according to the allocation, $Alloc^i$. The algorithms use the samples taken whilst the sensor is in its updating mode to calculate the allocation of actions, $Alloc^i$, to be used whilst the sensor is in its standard sampling mode. The frequency with which the sensor enters the updating mode is determined by the system designer, and depends on the variability of the environment. In relatively static environments the allocation will remain valid for sometime, and thus, updating can occur less frequently. In more dynamic settings the allocation must be updated more often⁶. Note that since the sensor samples at its maximum rate whilst in the updating mode, then the more often updating is performed, the less samples can be taken during any day whilst the sensor is in its standard sampling mode (in order to maintain energy neutral operation). In the experiments that we present in Section 6 the sensors update once every three days.

5.1 The Optimal Adaptive Sampling Algorithm

A first approach to providing a decentralised algorithm is to simply deploy the GP regression algorithm on each node, and then to find the subset of the samples that were taken whilst the sensor was in its updating mode that maximise our Fisher information metric (whilst also satisfying the energy constraints of the sensor when it is in its standard sampling mode). Thus, in more detail, if X^i is the set of sampling points taken whilst sensor i was in its updating mode, we wish to solve:

$$\arg \max_{X_s^i} FI(X_s^i) \quad (9)$$

where $FI(X_s^i)$ is our Fisher information metric calculated using GP regression, as described in Section 4, subject to the constraint that $X_s^i \subseteq X^i$ and $e_s |X_s^i| \leq E_r^i$. Given this subset of sampling points, we then calculate the allocation of sampling actions, $Alloc^i = \{a_1^i, a_2^i, \dots, a_w^i\}$, by simply counting the number of sample points in each time slot such that:

$$a_k^i = \sum_{j=1}^{|X^i|} \begin{cases} 1 & \text{if } x_j^i \in X_s^i \text{ and in time slot } k \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in I, \forall k \in H \quad (10)$$

⁶We note that setting this parameter automatically is part of our future work. We expect that this can be achieved by measuring the information content of samples collected each day, and re-entering the updating mode when this shows a significant departure from that which is expected.

Algorithm 1 Greedy Optimal Adaptive Sampling.

```

1:  $gpFI \leftarrow \{\}$ 
2: while  $addSamp > 0$  do                                     ▷ While there is an additional sample to allocate
3:   for each  $sampToTry \in remSP$  do                           ▷ Iterates each remaining sampling point
4:      $preSampTemp \leftarrow preSamp \cup sampToTry$ 
5:      $gpFI \leftarrow gpFI \cup CALCFIUSINGGP(preSampTemp)$      ▷ Calculates GP information value
6:   end for
7:    $[maxFI, indexOfNextSP] \leftarrow MAX(gpFI)$ 
8:    $nextSP \leftarrow NEXTSAMPLINGPOINT(indexOfNextSP)$        ▷ Finds nextSP
9:    $preSamp \leftarrow preSamp \cup nextSP$                    ▷ nextSP included into preSamp
10:   $remSP \leftarrow remSP \setminus nextSP$                    ▷ nextSP excluded from remaining sampling point (remSP)
11:   $gpFI \leftarrow \{\}$ 
12:   $addSamp \leftarrow addSamp - 1$ 
13: end while

```

Now, a naïve approach to finding this optimal subset is to simply enumerate all possible combinations. This approach, however, is too computationally intensive and works only for very small problems as it very rapidly becomes intractable. For instance, in the case of FLOODNET in which a sensor takes 288 samples a day whilst in its updating mode, but can only take 144 samples a day in its standard sampling mode, this algorithm would need to evaluate more than 10^{100} (C_{144}^{288}) solutions. This is clearly impossible to compute in a reasonable amount of time regardless of processor speed (for more details, see Section 6.3.2 for the run time performance of the algorithms that we present in this section).

5.2 The Greedy Optimal Adaptive Sampling Algorithm

Since the naïve enumeration approach is infeasible, we need a smarter means of tackling this problem. Thus, we devise a greedy optimal adaptive sampling approximation algorithm that again deploys the GP regression algorithm on the sensor node, but then works by allocating one additional sampling point at a time until there are no more samples to add. The allocated sampling points can not be altered in subsequent iterations (i.e. they are fixed). This significantly reduces the number of possibilities to compute compared to the naïve optimal algorithm which considers the whole set of combinations of sampling points as possible solutions. For example, in the case discussed above where 144 out of the possible 288 sampling points must be selected, we need only evaluate 31104 solutions (as compared to C_{144}^{288} solutions above). Nevertheless, this method is still reasonably slow to be run on nodes with the type of limited computational power found in WSNs in general and FLOODNET in particular.

In more detail, this algorithm works as follows (see Algorithm 1). At setup time, the vector variable $gpFI$ that temporarily records all the evaluated information values (FI as in Equation 5), is initialized to a null set (line 1). Each sensor $i \in I$ then presets a number of samples ($preSamp$) and equally distributes them into its time slots. Following this initialization phase, the sensor then uses its energy resources to iteratively sample $addSamp$ times more from the possible remaining sampling points $remSP$ (line 2). On each iteration, the sensor evaluates the information value of each remaining sampling point (line 5). It then allocates one sample at the sampling point $nextSP$ where the information value is increased the most (lines 7-9). At the end, the chosen $nextSP$ is included into the vector variable $preSamp$ (line 9). It is then excluded from the vector variable $remSP$ and the

variable $gpFI$ is cleared (lines 10 and 11 respectively). This repeats until there are no more samples to add.

The vector variable $preSamp$ eventually contains the greedy selection of sampling points, and thus, the allocation of sampling actions, $Alloc^i = \{a_1^i, a_2^i, \dots, a_w^i\}$, can again be found by simply counting the number of sample points in each time slot such that:

$$a_k^i = \sum_{j=1}^{|X^i|} \begin{cases} 1 & \text{if } x_j^i \in preSamp \text{ and in time slot } k \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in I, \forall k \in H \quad (11)$$

Finally, we note that the GP regression algorithm itself is relatively computationally expensive, and thus, we next present a heuristic algorithm that enables the sensor nodes to use a simpler means of valuing information in order to gain a faster performance.

5.3 The Heuristic Information-Based Adaptive Sampling Algorithm

We first describe a simplified valuation function that avoids the need to perform GP regression on the sensor node, and we then present the algorithm that we use to select sampling points in order to maximise it.

5.3.1 The Valuation Function. In this algorithm, we value information heuristically rather than computing it using Equation 5. We do this because the iterative process of calculating the information value using the GP regression technique is computationally expensive, and both the algorithms presented above require that this process is performed repeatedly. Specifically, we use simple linear regression and develop an information function that is based on the standard deviation of the best-fit regression line. This is appropriate, since given a small enough time window, the relationship between the time and environmental observations data can be approximated as a piecewise linear function.

Using this method, the uncertainty in a set of sensor readings is expressed in confidence bands about the linear regression line. The confidence band has the same interpretation as the standard deviation of the residuals (termed SE in Equation 13, where p represents the number of data points and \hat{y} is the new value of y calculated from the newly found slope b_1 and intercept b_0 variables), except that it varies according to the location along the regression line. The distance of the confidence bands from the regression line (τ_k) at point x_k is:

$$\tau_k = SE \sqrt{\frac{1}{p} + \frac{(x_k - \bar{x})^2}{\sum_{j=1}^p (x_j - \bar{x})^2}} \quad (12)$$

where x_k is the location along the x-axis data points where the distance is being calculated and \bar{x} is the mean value of X .

$$SE = \sqrt{\frac{\sum_{j=1}^p (y_j - \hat{y}_j)^2}{p - 2}} \quad (13)$$

where

$$\hat{y}_j = b_0 + b_1 x_j \quad (14)$$

and

$$b_0 = \bar{y} - b_1 \bar{x} \quad (15)$$

and

$$b_1 = \frac{\sum_{j=1}^p (x_j - \bar{x})(y_j - \bar{y})}{\sum_{j=1}^p (x_j - \bar{x})^2} \quad (16)$$

In order to perform this simple linear regression properly, the input must consist of at least three data points. This is because if there are only two data points they will produce a smooth linear regression line (with no standard deviation), while anything less than that will result in invalid inputs. For these reasons, we enforce the fact that a sensor must at least sample once in every time slot (defined as the minimum sampling rate, therefore, $E_r^i \geq w \cdot e_s, \forall i \in I$, where as defined in Section 3, w is the number of time slots and e_s is the energy required for one sample). In this way, given the standard deviation and the confidence bands, we are able to tell whether one set of observations is more valuable than another which, in turn, allows us to define a value associated to every action.

Given the expressions above, we can now derive the total deviation, TD , for a set of data points by calculating the area between the confidence bands, and this total deviation represents our uncertainty over this period. Specifically, we use a trapezoidal numerical integration method for this [Rabinowitz and Davis 2006]. The trapezoid approximation (i.e. trapezoid sum T_s) of $\int_a^b f(x)dx$ that is associated with the partition $a = x_1 < x_2 < \dots < x_n = b$ is given by:

$$\begin{aligned} T_s &= \frac{1}{2}[f(x_1) + 2f(x_2) + \dots + 2f(x_{n-1}) + f(x_n)]\Delta x \\ &= \frac{1}{2}[f(x_1) + 2 \sum_{j=2}^{n-1} f(x_j) + f(x_n)]\Delta x \end{aligned} \quad (17)$$

and, thus, we are now able to derive:

$$TD_n = \frac{1}{2}[2\tau_1 + 4 \sum_{j=2}^{n-1} \tau_j + 2\tau_n]\Delta x \quad (18)$$

For example, consider the case shown in Figure 6 that uses real data collected from Sensor 1 on Oct 14th 2005 between time unit 201 and 209 (where one time unit represents a five minute interval). Figure 6(a) shows a case where 9 samples are taken and Figure 6(b) shows another case where only 5 are taken. The X axis represents the time unit, whilst the Y axis represents the water level. The solid line denotes the simple linear regression line, whilst the curved dashed lines demarcate the confidence bands (it represents the boundaries of all possible straight lines). In the case where 9 samples are taken, the procedure just described allows us to calculate the total deviation, TD , as 14.6050, while in the case of 5 samples, it is 35.5968. In both cases, the total deviation is represented by the shaded area between the confidence bands; less uncertainty is denoted by a smaller area.

Given this total deviation, we can now simply derive the gain in information value (or the decrease in uncertainty) when different sampling decisions are made.

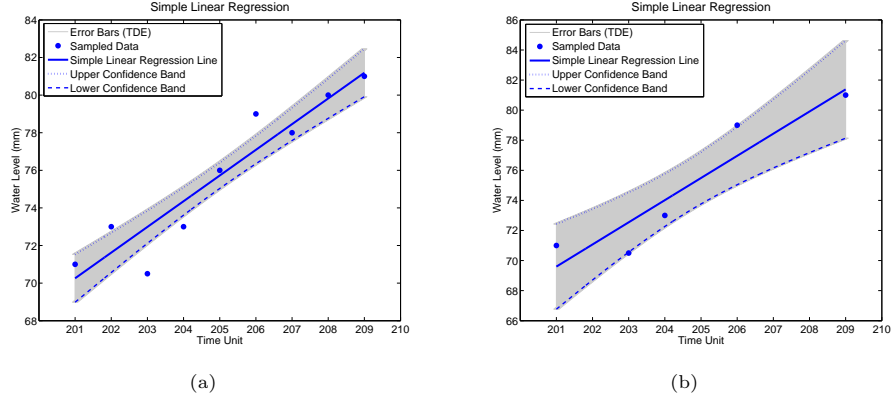


Fig. 6. Example application of the linear regression based valuation metric applied to data collected from Sensor 1 on Oct 14th 2005 between time unit 201 and 209. Two cases are shown; (a) where 9 samples are taken and (b) where only 5 samples are taken.

More formally, $Gain_u^i(k)$ is defined as the reduction in total deviation that sensor i can achieve by taking samples at rate c_u^i rather than the minimum sampling rate c_1^i in time slot k , and is given by:

$$Gain_u^i(k) = TD_{c_1^i}(k) - TD_{c_u^i}(k), \text{ where } c_u^i \in C^i, i \in I, k \in H \quad (19)$$

This minimum sampling rate is applied as a basis where a sensor gains zero value.

The data values for each sensor are often best represented in a table format, as shown in Table II (this sampled data is chosen arbitrarily for illustrative purposes and its values are calculated using Equation 19). In this table, the columns represent the time slot, for instance where $column = 3$, if this particular sensor i chooses to sense at sampling rate c_2^i , c_3^i , or c_s^i , in return it will gain a corresponding reduction in total deviation of 27.59, 43.79, or 55.23 (that is $Gain_2^i(3)$, $Gain_3^i(3)$, or $Gain_s^i(3)$ respectively) compared to if it had only taken samples at its minimum sampling rate (c_1^i) during the same period.

As described earlier, when in its updating mode, each sensor samples at its maximum rate (c_s). Now, by taking subsets of samples (corresponding to the set of actions specified in Table's II row header) from the full set and performing the linear regression on these subsets, we obtain a new total deviation for each subset. The values that will be assigned to the table are the total deviation difference between sampling at the minimum rate and at other rates. For instance where $column = 3$, if the total deviation that is produced with a subset of samples c_2^i taken during time slot 3 ($TD_{c_2^i}(3)$) has a value of 128.66, while that of a minimum sampling rate c_1^i taken between the same period ($TD_{c_1^i}(3)$) is 156.25, then the value inside $column = 3$ and $row = 2$ ($Gain_2^i(3)$), will be 27.59.

5.3.2 The Algorithm. We now focus on how to search for an allocation of sensor's actions that maximises the information metric described above. For this purpose, we introduce V as a $s \times w$ matrix with s number of actions and w number of

	1	2	3	w
c_1^i Sampling Rate	0	0	0	0
c_2^i Sampling Rate	0.29	1.21	27.59	3.88
c_3^i Sampling Rate	0.31	1.51	43.79	8.92
c_x^i Sampling Rate
c_s^i Sampling Rate	0.33	1.55	55.23	13.45

Table II. Action-value table for sensor i .

time slots:

$$\mathbf{V}^i = \begin{bmatrix} v_{11}^i & v_{12}^i & \dots & v_{1w}^i \\ \vdots & \vdots & \ddots & \vdots \\ v_{s1}^i & v_{s2}^i & \dots & v_{sw}^i \end{bmatrix} \quad \mathbf{D}^i = \begin{bmatrix} d_{11}^i & d_{12}^i & \dots & d_{1w}^i \\ \vdots & \vdots & \ddots & \vdots \\ d_{s1}^i & d_{s2}^i & \dots & d_{sw}^i \end{bmatrix}$$

such that v_{uk}^i represents the value that sensor i will get if it chooses to perform action c_u^i in time slot k (i.e. $Gain_u^i(k)$). \mathbf{D} is a matrix of binary values and each of the elements corresponds to a decision variable (a “1” represents a state where the sensor carries out the corresponding c_u^i action, whilst a “0” represents another state where the sensor does not carry out the corresponding c_u^i action). For instance, when $d_{11}^i = 1$ then this sensor i chooses to perform action c_1^i in time slot 1. This also means that $d_{u1}^i = 0, \forall u \in C^i \setminus c_1^i$.

In more detail, the objective function to be maximised is defined in Equation 20. The constraint in Equation 21 states that every sensor can only elect one action at any particular point of time, whereas that in Equation 22 states that the total number of samples taken by it must not exceed the maximum number of samples it can take on that day:

$$\{V_{max}^i, D_{max}^i\} = \arg \max_{\{V^i, D^i\}} \sum_{u \in C^i, k \in H} v_{uk}^i d_{uk}^i, \forall i \in I \quad (20)$$

subject to:

$$\sum_{u=1}^s d_{uk}^i = 1, \forall i \in I, \forall k \in H \quad (21)$$

$$\sum_{k=1}^w c_1^i d_{1k}^i + c_2^i d_{2k}^i + \dots + c_s^i d_{sk}^i \leq N^i, \forall i \in I \quad (22)$$

where N^i is calculated such that $N^i e_s \leq E_r^i$ as described in Equation 1.

This problem, as formulated above, can be cast as a person-task assignment problem⁷ [Yong et al. 1993]. Given this insight, we can solve the problem using binary integer programming (BIP) [Chen et al. 2000], which is a subset of linear programming. A popular method to solve this numerically is the simplex algorithm and in this case we exploit the GNU Linear Programming Kit⁸ (GLPK) to do so.

⁷In the assignment problem, we want to assign a set of people to do a set of tasks. Each person takes a certain number of minutes to do a certain task, or cannot do a particular task at all, and each person can be assigned to exactly one task. The ultimate aim, here, is to minimize the total time taken to do all of the tasks.

⁸<http://www.gnu.org/software/glpk/>

Algorithm 2 Heuristic Information-Based Adaptive Sampling.

```

1: updSSched  $\leftarrow$  TRUE ▷ Sensor in updating mode
2: sRate  $\leftarrow$  MAX_S_RATE ▷ Sensor samples at maximum rate
3: readings  $\leftarrow$  {}
4: loop
5:   if sTime = NOW then ▷ Time to sample
6:     readings  $\leftarrow$  PERFORMSAMPLING(sTime)
7:     if  $\neg$ updSSched then
8:       timeSlot  $\leftarrow$  TIMESLOT(sTime)
9:       sRate  $\leftarrow$  GETSRATE(timeSlot) ▷ Changes sampling rate according to the timeSlot
10:    end if
11:    SETSTIME(sTime + sRate) ▷ Sensor sets its next sampling time
12:  end if
13:  if tTime = NOW then ▷ Time to transmit
14:    uError  $\leftarrow$  CALCUEERROR(readings) ▷ Calculates information uncertainty in current readings
15:    if dateChanged then
16:      daysCount  $\leftarrow$  daysCount + 1
17:      if updSSched then
18:        CALCTDREDUCTION(readings) ▷ Computes reduction in total deviation
19:        FINDSSCHEDULE() ▷ Binary integer programming solver
20:        updSSched  $\leftarrow$  FALSE
21:      end if
22:    end if
23:    if  $\neg$ updSSched  $\wedge$  HASENOUGHENERGY()  $\wedge$  (daysCount  $\geq$  CONST) then
24:      updSSched  $\leftarrow$  TRUE ▷ Schedule updated
25:      daysCount  $\leftarrow$  0
26:      sRate  $\leftarrow$  MAX_S_RATE
27:    end if
28:    SETTTIME(tTime + tRate) ▷ Sensor sets its next transmitting time
29:    readings  $\leftarrow$  {}
30:  end if
31: end loop

```

Having described the techniques that we use, we now seek to present the rest of the heuristic information-based adaptive sampling algorithm (see Algorithm 2 and Figure 7). Specifically, the algorithm, which is distributed and installed on each sensor in the network, provides a means for the individual nodes to adjust their own sampling rates based only upon their local historical data and remaining energy resources. Now, within the initialization phase, some required variables are set. These include the boolean variable *updSSched* which is set *TRUE* to indicate that the sensor starts in its updating mode, and then having calculated an allocation of sampling actions for subsequent days, enters its standard sampling mode.

Following the initial updating phase, each sensor $i \in I$ enters an infinite loop state. On each iteration, it checks its sampling and transmitting time. Whenever the current loop represents the time that it needs to sample (line 5 or state 2), the function *PerformReading* instantiates a new *reading* and attaches it to the end of the variable *readings*. Subsequently, if the sensor is not in updating mode, its *sRate* is assigned a value equal to the sampling rate in its schedule, corresponding to the appropriate time slot (line 9 or state 3). The sensor then sets its next sampling time variable *sTime*. Inside the same loop iteration, whenever the sensor is also required to transmit its current readings (line 13 or state 4), it firstly calculates the total deviation in this set of readings by using the simple linear regression method described earlier (i.e. calling function *CalcUError* with Equation 18). Later, if the sensor detects that it has entered the following day and it is also in updating mode, it will call the function *CalcTDReduction* (with Equation 19) to compute the reduction in total deviation that the sensor can achieve by taking more samples

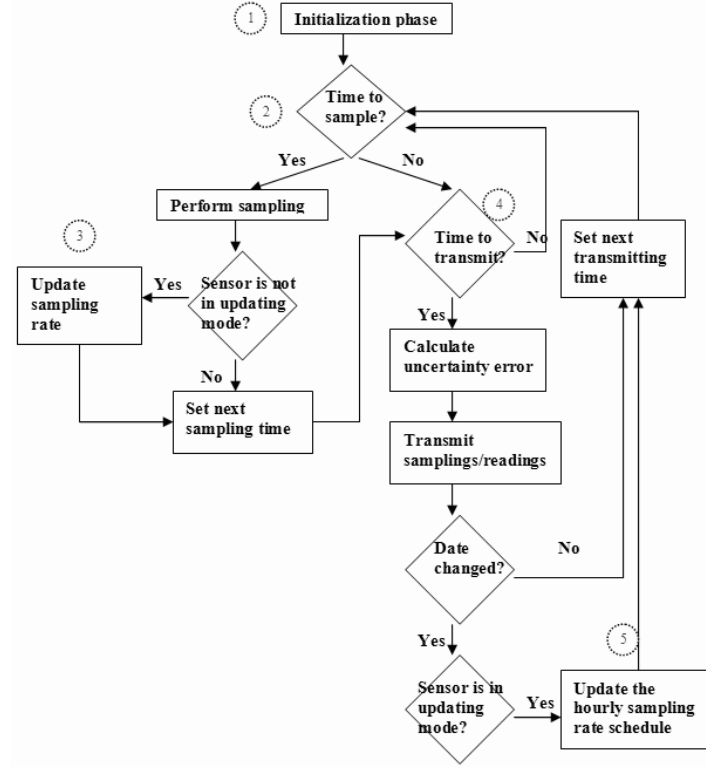


Fig. 7. State diagram of the algorithm.

than the minimum sampling rate. Function *FindSSchedule* then uses the BIP GLPK solver to evaluate (in real-time) the best allocations of sensor i 's schedule and resources that maximise the total deviation reduction (i.e. Equation 20), given the sensor's current energy constraints of Equations 21 and 22 (line 19 or state 5). The allocation of sampling actions, $Alloc^i = \{a_1^i, a_2^i, \dots, a_w^i\}$, is thus determined by:

$$a_k^i = c_u^i, \text{ where } u \in \{1..s\} | d_{uk}^i = 1, d_{uk}^i \in D_{max}^i \quad \forall i \in I, \forall k \in H \quad (23)$$

6. EVALUATION

Having described the three decentralised algorithms for adaptive sampling, we now turn to their evaluation in order to examine their performance and effectiveness. We first describe the experimental setup and the benchmarks, and then go onto the actual evaluation.

6.1 Network and Parameters Initialization

In our experiments, we use a simulation of the FLOODNET network, driven by real data for batteries, tide readings, and cloud cover (used to model solar energy harvesting). The experiments are run using FLOODNET's actual topology with a fixed number of nodes (twelve) at fixed locations (i.e. the nodes are immobile).

The sampled data model (worth approximately eight days of measured data starting from Oct 14th 2005 00.00AM) for each node was fixed for each instance of the experiments. All the cloud parameters (including the cloud coverage, wind speed, and cloud thickness) are initialized with realistic data (at FLOODNET’s site) available in METAR⁹ format. These are all done in order to reproduce the FLOODNET scenario as realistically as possible. The remaining battery energy of each sensor and its recharging rate are set to be low so that it can not continuously sample at its maximum rate. Given these constraints, sensors must therefore allocate resources and schedule themselves to determine how often and when to sample efficiently in order to maximise their collected information value.

In a simulation run, nodes can fail due to their battery depletion, but they can not be added or removed. For the sake of simplicity and in order to exploit all the possible changes in the system, at this point of time, we only consider four different actions ($s = 4$) describing the sensor’s sampling rate. Thus, sensors can either sample one, three, six, or twelve times per hour (i.e. $C^i = \{1, 3, 6, 12\}$, $\forall i \in I$).

6.2 Benchmark Algorithms

In our experiments, the benchmark algorithms include:

— *A Naïve Non-Adaptive Sampling Algorithm.* This dictates that each sensor $i \in I$ should sample at its maximum rate, c_s^i , whenever there is enough battery energy to do so. The sensor’s sampling behaviour is therefore non-adaptive and can be described as:

$$a_k^i = c_s^i \quad \forall i \in I, \forall k \in H \quad (24)$$

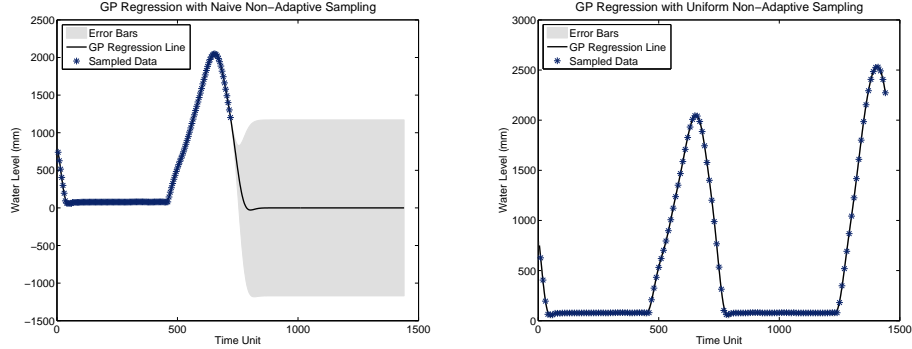
— *A Uniform Non-Adaptive Sampling Algorithm.* This dictates that each sensor $i \in I$ in the network should simply choose to divide the total number of samples it can perform in a day (N^i where $N^i = \frac{E_r^i}{e_s}$) equally into its time slots, such that:

$$a_k^i = \arg \max_u u \quad \text{where } u \in \left\{ c_y^i \in C^i \mid c_y^i \leq \frac{N^i}{w} \right\} \quad \forall i \in I, \forall k \in H \quad (25)$$

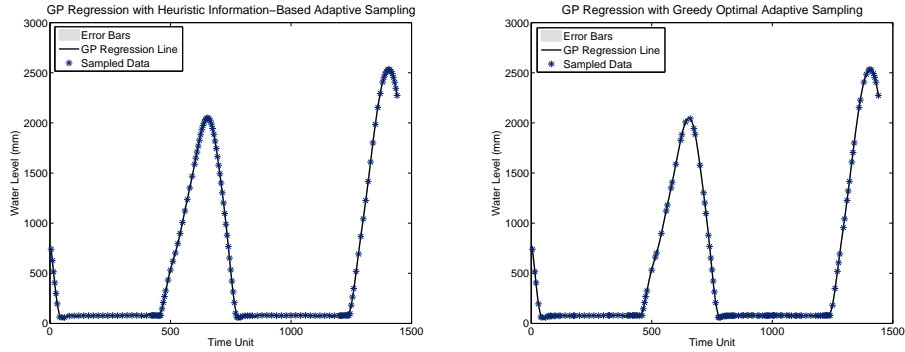
— *A Utility-based Sensing And Communication (USAC) Algorithm.* This is a state-of-the-art algorithm that lets each sensor adjust its sampling rate depending on the rate of change of its observations (see Section 7 for more details). Specifically, the algorithm uses a linear regression method which is run to determine the next predicted data, $dat(t + 1)$, with some bounded error (termed its *confidence interval*, ci). If the next observed data falls outside ci , the node sets its sampling rate to the maximum rate in order to incorporate this phase change. However, if data falls within the ci , it implies that the node is allowed to reduce its sampling rate for energy efficiency due to the presence of predictable information that has a low value. The USAC algorithm does not have a notion of time slot (as described in Section 3) and therefore, each sensor has the flexibility to change its own sampling rate at any point of time¹⁰.

⁹<http://weather.noaa.gov/weather/metar.shtml>

¹⁰All other algorithms, on the other hand, dictate that each sensor should only change its sampling rate at subsequent time slots.



(a) GP Regression with Naïve Non-Adaptive Sampling Algorithm collecting Fisher information value of $FI = 1.804 \cdot 10^{-3}$. (b) GP Regression with Uniform Non-Adaptive Sampling Algorithm collecting Fisher information value of $FI = 3.278 \cdot 10^{-2}$.



(c) GP Regression with Heuristic Information-Based Adaptive Sampling Algorithm collecting Fisher information value of $FI = 7.396 \cdot 10^{-2}$. (d) GP Regression with Greedy Optimal Adaptive Sampling Algorithm collecting Fisher information value of $FI = 1.045 \cdot 10^{-1}$.

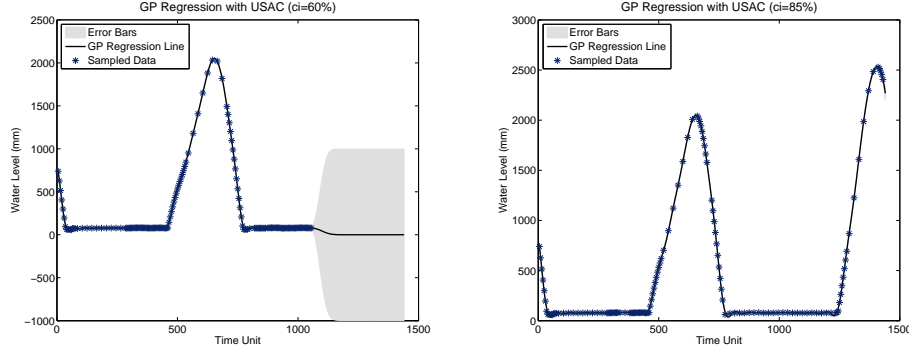
Fig. 8. GP Regression evaluated on daily basis for Sensor 1 collecting samples on Oct 15th 2005. One time unit represents a one minute interval.

Now, assuming that the sampling rate of sensor $i \in I$ at t point of time, $Sr^i(t)$, is equal to c_x^i , where $c_x^i \in C^i$, then its sampling rate at $t + 1$ is defined as:

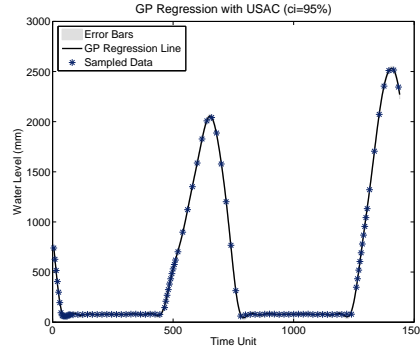
$$Sr^i(t + 1) = \begin{cases} c_{x-1}^i & \text{if } Lo(ci) \leq dat(t + 1) \leq Up(ci) \\ c_s^i & \text{otherwise} \end{cases} \quad (26)$$

where $Lo(ci)$ and $Up(ci)$ are the lower and upper bound of ci respectively. As the setting of ci is central to USAC's operation, and because no guidelines are given about what values to use, here we use the following range of values: 60%, 85%, and 95%. This is, we believe, sufficient to fully examine USAC's performance in this domain.

—*Unconstrained Sampling.* This ignores the constrained energy of the sensor, and allows the sensor to sample its maximum rate for the entire trial period. This represents an absolute upper bound on the value of information that can be collected, but clearly can not actually be implemented in practice (since the



(a) GP Regression with USAC ($ci=60\%$) collecting Fisher information value of $FI = 4.058 \cdot 10^{-3}$. (b) GP Regression with USAC ($ci=85\%$) collecting Fisher information value of $FI = 6.142 \cdot 10^{-2}$.



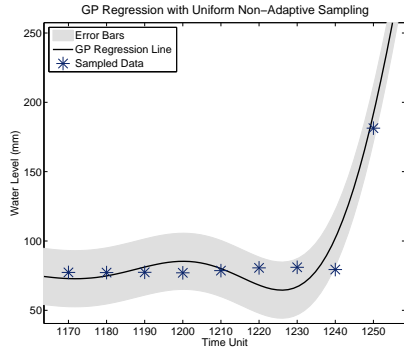
(c) GP Regression with USAC ($ci=95\%$) collecting Fisher information value of $FI = 5.559 \cdot 10^{-2}$.

Fig. 9. GP Regression with USAC using different values of ci evaluated on daily basis for Sensor 1 collecting samples on Oct 15th 2005. One time unit represents a one minute interval.

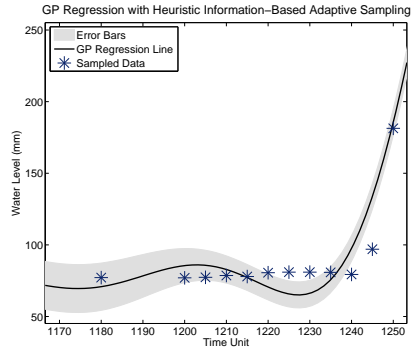
sensors will deplete their batteries before the end of the trial period).

6.3 Results

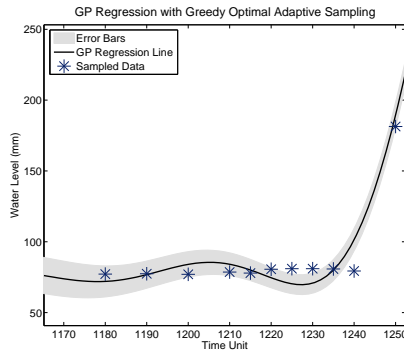
The aim of these experiments is to compare the total Fisher information value gathered at the base-station from each of the individual sensors for the various decentralised control regimes. To this end, Figures 8 and 9 show the comparison of Fisher information values that are evaluated (on a daily basis using the GP regression technique as per Equation 5) with different sets of Sensor 1 observations collected using the different algorithms. It also shows that Sensor 1 obtains the lowest uncertainty in its set of readings on that day and, hence, the highest information value (FI), when it collects samples using the greedy optimal adaptive sampling algorithm. With the heuristic information-based adaptive sampling algorithm, on the other hand, it collects slightly less information value. However, this value is significantly higher compared to those collected using the two non-adaptive and USAC benchmarks.



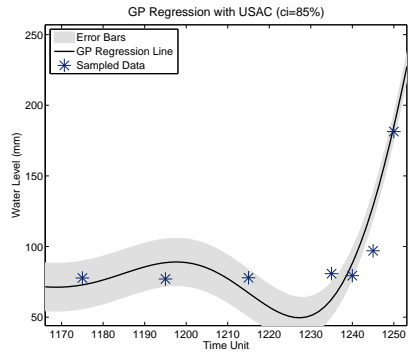
(a) GP Regression with Uniform Non-Adaptive Sampling Algorithm.



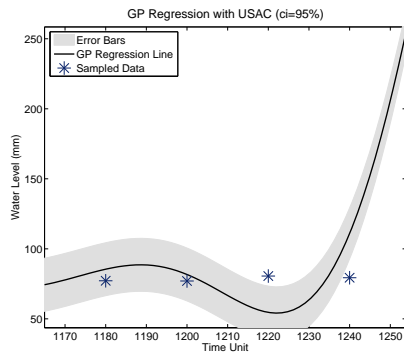
(b) GP Regression with Heuristic Information-Based Adaptive Sampling Algorithm.



(c) GP Regression with Greedy Optimal Adaptive Sampling Algorithm.

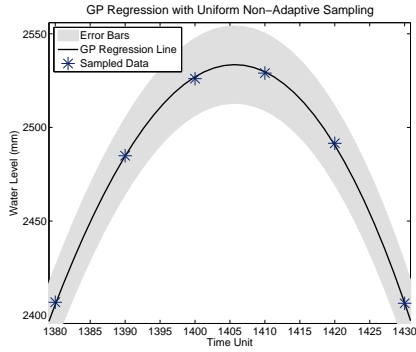


(d) GP Regression with USAC ($ci=85\%$).

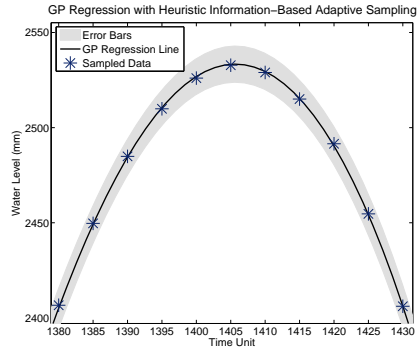


(e) GP Regression with USAC ($ci=95\%$).

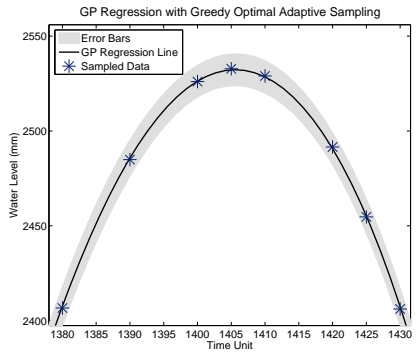
Fig. 10. GP Regression on sampled data taken between time unit 1170 and 1250 from Sensor 1 operating on Oct 15th 2005. One time unit represents a one minute interval.



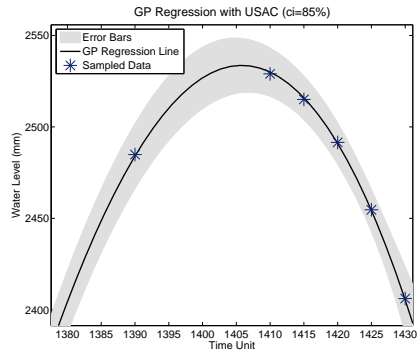
(a) GP Regression with Uniform Non-Adaptive Sampling Algorithm.



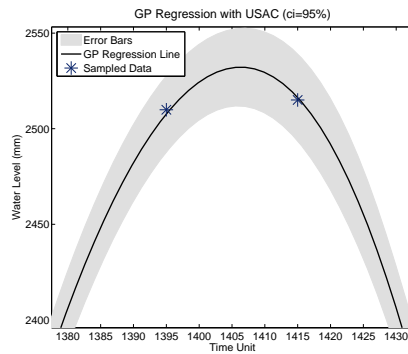
(b) GP Regression with Heuristic Information-Based Adaptive Sampling Algorithm.



(c) GP Regression with Greedy Optimal Adaptive Sampling Algorithm.



(d) GP Regression with USAC ($ci=85\%$).



(e) GP Regression with USAC ($ci=95\%$).

Fig. 11. GP Regression on sampled data taken between time unit 1380 and 1430 from Sensor 1 operating on Oct 15th 2005. One time unit represents a one minute interval.

The actual $\hat{\sigma}^2(x)$ in Figures 8 and 9 are too small to be visible on such a scale (for enlarged versions and to show more clearly how the heuristic adaptive sampling algorithms outperform the non-adaptive and USAC algorithms, see Figures 10 and 11¹¹). Moreover, we also compare the computational time performance of the various algorithms in order to provide the other side of the optimality and computational time trade-off.

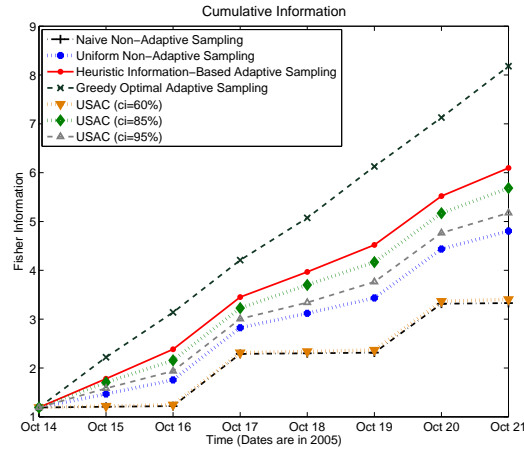
6.3.1 Information Value Analysis. As can be seen in Figure 12(a), the heuristic information-based adaptive sampling algorithm performs well; compared to the naïve and uniform non-adaptive approaches respectively, this algorithm consistently increases the total Fisher information collected by about 83% and 27% per day over the trial period. The plot clearly shows the superiority of the heuristic information-based adaptive sampling algorithm and that the information value of the data collected found by it is approximately 75% of the greedy optimal's.

Furthermore, in comparison with case of sensors that can sample unconstrained by power requirements (i.e. they can sample at the maximum rate throughout the trial period) our heuristic information-based adaptive sampling algorithm is approximately 66% of this upper bound. This upper bound corresponds to the peaks in Figure 12, and thus, we do not show it as an additional line in this plot.

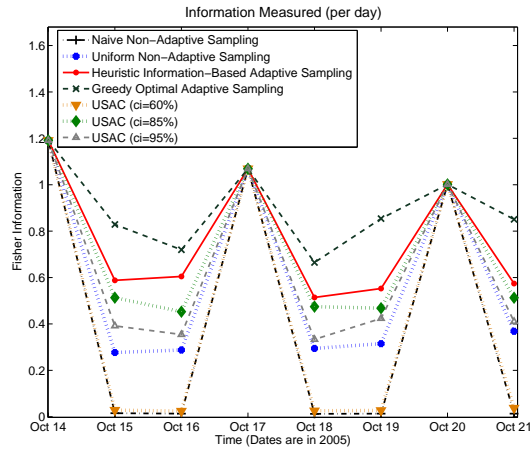
The heuristic information-based adaptive sampling algorithm also outperforms USAC for all values of ci . The main reason is due to the absence of a forward planner in USAC. In more detail, Figure 9(a) shows how USAC ($ci = 60\%$) behaves poorly with performance similar to that of the naïve one. With this non-carefully chosen ci value, a small change in environmental readings will trigger each sensor to change its sampling rate to its maximum (of which energy is never reserved for possible future usage). As it does not have the power to continuously sample at this maximum rate, it often runs out of energy during a day and so collects no information for a long period of time. Figures 9(c), 10(e), and 11(e), on the other hand, show how USAC ($ci = 95\%$) performs in a similar fashion to the uniform non-adaptive one. In this setting, the next predicted data is highly likely to fall within the bounded error, ci , and hence, the algorithm dictates that each sensor should decrease its sampling rate due to the presence of low value predictable readings. Here, the harvested and renewable energy is not allocated effectively because the algorithm does not maintain the nodes in energy neutral operation mode (i.e. balance the amount of energy harvesting against that of energy consumption). USAC's optimal ci value for FLOODNET data is found to be 85% in which the algorithm collects information with value 8% lower than that collected by our heuristic information-based adaptive sampling algorithm over the trial period.

Additionally, Figure 12(b) shows more clearly how the adaptive sampling algorithms achieve this performance. After leaving the schedule updating mode (i.e. the second day of a simulation, as can be seen in Figures 13 and 14), a sensor is able to perform adaptive sampling by conserving its battery energy in order to take more samples during the most dynamic events, while taking fewer samples during

¹¹In these figures, the GP regression using the naïve non-adaptive and USAC ($ci = 60\%$) sampling algorithms are not shown because the plots would simply be covered by the error bars as can be seen from Figures 8(a) and 9(a) respectively.



(a) Cumulative information gathered over an 8-day period plotted against time.



(b) Information gathered daily over an 8-day period plotted against time.

Fig. 12. Information measurement graph performances.

the static ones. In our case, the dynamic events of a tide occur at the time it comes in (specifically when the sensor rises off mud, between 07.00 and 09.00 in Figure 13), reaches the peak (between 10.00 and 11.00), and goes out (between 12.00 and 14.00). During these events, sensors normally set their sampling rates to a maximum value (i.e. in our case, at five minute intervals). As a result, from the second day onward, Figure 12(b) shows a gain in information value collected (particularly during the dynamic events), except when the sensors are in updating mode (on Oct 17th and 20th). In this mode, all sensors sample at their maximum rates (as discussed in Section 5.3.2), therefore on those dates, the information valuations of

ACM Journal Name, Vol. V, No. N, MMM 20YY.

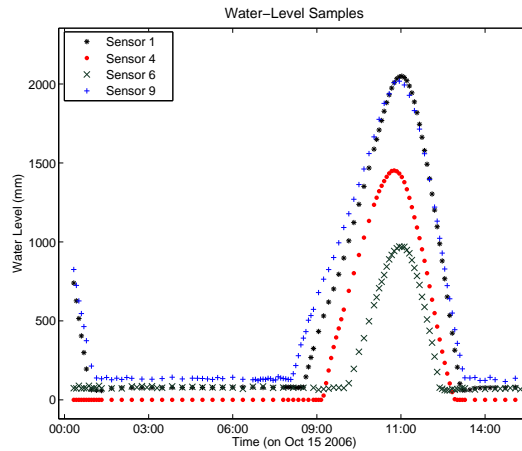


Fig. 13. Water samples gathered on the second day of the simulation using the heuristic information-based adaptive sampling algorithm. Graph only displays some selected nodes for better visibility.

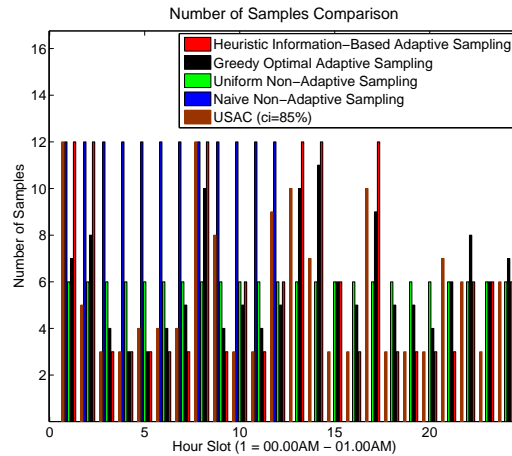


Fig. 14. Number of samples taken in each hour slot from Sensor 1 operating on Oct 15th 2005. The heuristic information-based adaptive sampling algorithm behaves in the same pattern as the greedy optimal one. Both generally samples more often during dynamic events while sacrificing some energy to sample less during static ones.

the five approaches are the same.

6.3.2 *Run Time Performance.* The optimal adaptive sampling algorithm works only for very small problems as it very rapidly becomes infeasible for even small- to medium-sized ones. For instance, consider an adaptive sampling problem in which a node has sufficient battery capacity to sample only 3 times from the possible 288 sampling points in a day. In this scenario, there are $10^5(C_3^{288})$ solutions to

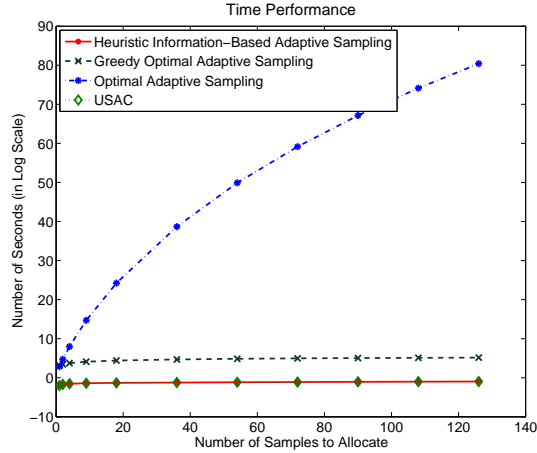


Fig. 15. Computational Time Performance.

enumerate, which is just about possible to do with a modern computer in a finite amount of time. However, for a slightly larger problem, where a node has the flexibility to sample 36 times from the possible 288 sampling points in a day, there are now $10^{37}(C_{36}^{288})$ solutions to enumerate, which is intractable in a reasonable amount of time, even for a very fast computer. Assuming a 3GHz desktop PC on which the GP regression technique takes approximately 5 seconds, we estimate this would take 10^{30} years to compute. In more detail, Figure 15 shows a comparison of the computation time of all the three algorithms that we consider.

The greedy optimal adaptive sampling algorithm significantly reduces the computation time, since it reduces the number of possible combinations of sampling points that must be compared. However, it is still too slow to be run on the nodes that have similar computational power to FLOODNET's. This is because there are still $\frac{1}{2} \cdot addSamp \cdot ((remSP - addSamp) + remSP)$ possible solutions to iterate. For instance, for a problem in which a node is capable of allocating 36 samples from the possible 288 sampling points, there are still 9720 solutions to evaluate, and experiments indicate that this takes approximately 14 hours to compute on a standard 3Ghz desktop PC.

The heuristic information-based adaptive sampling algorithm, on the other hand, runs in real time on the current configuration. This is due to the performance of the linear programming optimization technique and the simplified information valuation metric. Again, consider the 36 sample problem for example, this algorithm calculates the preferred solution in hundreds of milliseconds. Finally, due to the similar nature of linear regression method used for valuing information, USAC also executes in real time.

7. RELATED WORK

The three algorithms that we present in Section 5 all exploit temporal correlations within the data from a sensor to select the most informative sampling points. The

key intuition here is that when the data is highly correlated, and thus changing slowly, there is no need to sample frequently. A number of other algorithms within the literature also use temporal or spatial correlations (or both) in order to make effective sampling decisions, and we review them here.

With respect to spatial correlations between sensors, Willett et al. [2004] have studied the backcasting adaptive sampling method in which multiple nodes that are spatially correlated form small subsets of nodes that then communicate their information to a fusion coordinator. Based upon this information, the coordinator then selectively activates additional nodes in order to achieve a target error level. In our setting we do not attempt to fuse information from separate sensors, and since we require a decentralised solution with minimal additional communication between sensor nodes, we do not address the spatial correlations between the sensors (although we acknowledge that these spatial correlations almost certainly do exist).

More similar to our work, Krause et al. [2006] use a Gaussian process to model the spatial correlations between sensors, and then use these correlations to select the subset of sensor placements that is most informative. The mutual information metric that they use is very similar to the Fisher information metric that we present here. Since they perform their selection after learning the covariance function of the Gaussian process they can exploit the submodularity of their information metric to place a bound on the performance of a greedy selection scheme. In our case, the Gaussian process represents temporal correlations, and since we relearn the hyperparameters of the Gaussian process based on the sampling schedule that we have selected, we cannot exploit submodularity to derive a bound on our own greedy optimal adaptive sampling algorithm (although the methodology is similar).

With regard to the Fisher information based metric that we presented in this paper, we note that Fisher information has previously been within the sensor network literature to value position estimates in tracking scenarios. Specifically, Chu et al. [2002] describe *Information-Driven Sensor Querying* (IDSQ) for an array of sensing nodes that are used to estimate the position of a target being tracked. The algorithm is decentralised, and each sensor selectively chooses to fuse its own information with that of other available sensors, in order to update its current belief about the target's position. It is the job of IDSQ to direct each sensor to fuse the most valuable data (i.e. data that more accurately represents the target's position), and to do so, a Fisher information measure is used. The approach is similar to ours in that a valuation function is used to determine the value of previously collected observations and this is then used to decide on the next sampling plans of each sensor. However, the information metric derived here is based on a fused position estimate, whereas we use Fisher information to value a set of temporally correlated sensor samples.

Similarly, Makarenko and Durrant-Whyte [2004] describe a negotiation-based *Bayesian Decentralised Data Fusion* (BDDF) technique for an array of wireless nodes in a network. Their work accounts for the uncertainty inherent in such tracking applications, and a Bayesian non-linear filtering method is used to aggregate sensed data. The local filter of a node fuses the observations, and these fused observations are used to decide the node's next sensing plan.

In a somewhat different setting, but still concerned with the decentralised ap-

proaches to decision making within sensor networks, Mainland et al. [2005] present a market-based approach for determining efficient node resource allocations in WSNs. Rather than manually tuning node resource usage, or providing specific algorithms as we do here, *Self Organising Resource Allocation* (SORA) defines a virtual market in which nodes sell goods (e.g. data sampling, data relaying, data listening, or data aggregation) in response to global price information that is established by the end-user. However, this approach again involves an external coordinator to determine and set the prices in order to induce the desired global behaviour for the network. Moreover, it is not clear how this should actually be done in practice.

Most similar to our work, and the one that exploits temporal correlations between sensed data, is the *Utility Based Sensing and Communication* (USAC) algorithm proposed by Padhy et al. [2006]. As discussed in Section 6.2, this is a decentralised control regime for adaptive sampling, designed for an environmental WSN measuring subglacial movement (Glacsweb). The adaptive sampling aspect of the algorithm models temporal variations in the environmental parameter being sensed as a piece-wise linear function, and then uses a pre-specified *confidence interval* parameter in order to make real-time decisions regarding the sampling rate of the sensor nodes. Linear regression is used to predict the value of future measurements, and if the actual sensor reading exceeds the confidence interval parameter, the sensor starts sampling at an increased rate. However, this parameter (and several others such as the window length and actual sampling rate) must be carefully selected, and a poorly chosen value can result in very poor performance in our setting. Since the algorithm does not explicitly perform any forward planning, the sensor can rapidly deplete its battery if the increased sampling rate is constantly re-triggered by data that is far from linear (as can be seen in Figure 12).

A limited amount of work also attempts to exploit both spatial and temporal correlations. In particular, Rahimi et al. [2004] present a *Nested Stratified Random Sampling* (NSRS) policy for a set of mobile nodes exploring three-dimensional environmental structure and phenomena. In their work, spatio-temporal correlations exist between readings locally within each individual node, and the sampling distribution of each node is determined by these correlations. The work, however, does not model the uncertainty in sensor readings. Furthermore, with mobile nodes, we believe that the information content of each node's reading should be dependent on the node's position at the time when this reading is actually taken. However, their work does not address this important issue.

More recently, Dang et al. [2007] have proposed an adaptive sampling algorithm to find the optimal cruise path of a mobile sensor node in order to collect data that maximally reduces the uncertainty of a data assimilation model (that is based on the Sigma Point Kalman Filter). Here, the environment being monitored is modelled as a set of grid points that are available for the sensor to sample at. The next sampling point is chosen to be the point that results in the lowest trace of the predicted covariance matrix indicating how uncertain the estimated environment state is. However, with this algorithm, there is an issue of scalability as the computation time of searching the next sampling point increases exponentially when extra nodes are introduced.

Finally, Osborne et al. [2008] use a multi-output Gaussian process to explicitly

model both temporal and spatial correlations between a small number of sensors. The GP is used for adaptive sampling whereby it can determine both the time, and the sensor from which the next sample should be taken, to ensure that the uncertainty regarding the environmental parameter being measured at each sensor location stays below a pre-specified threshold. However, as with the USAC algorithm, there is no forward planning. Moreover, the algorithm is centralised, since it requires information from all of the sensors in order to model the spatial correlations between them, and it is relatively computationally expensive; the novelty in the paper above, being a computationally efficient formalism of the GP. The computational cost precludes it being deployed on the current generation of sensor nodes, and furthermore, since it requires sensors to exchange data with one another, it would also incur additional communication cost that could possibly outweigh any saving achieved through more effective sampling.

8. CONCLUSIONS AND FUTURE WORK

In this article, we have focused on issues associated with energy management in general and adaptive sampling in particular. We have developed a principled information metric based upon Fisher information and Gaussian process regression that allows the information content of a sensor's observations to be expressed, and given a set of sensor readings, we have shown how an optimal and a greedy optimal adaptive sampling approximation algorithm can be devised. They are, however, only tractable for very small problems, and thus, we have developed a more practical, heuristic information-based adaptive sampling algorithm with the ultimate aim of maximizing the information value of the data collected at a base-station, given the limited energy resources available. This approach is a better choice for larger sampling problems (beating the benchmarks and obtaining performance close to the optimal one, but with much lower time complexity). The empirical results show that all three decentralised control algorithms for information-based adaptive sampling are effective in balancing the trade-offs associated with wanting to gain as much information as possible by sampling as often as possible, with the constraints imposed on these activities by the limited power available.

Although the effectiveness of these algorithms is evaluated within the FLOOD-NET domain, the challenges that are involved here are very similar to those that occur in the design of many other WSNs. Specifically, many WSNs are being deployed in the domain of environmental phenomena monitoring, and data in these settings typically exhibits periodic features (as we have with the tides) due to the natural cycle of day and night. The GP regression algorithm learns this periodicity from the data, and thus, can be applied directly. The linear programming technique, together with the utility functions and constraints, can also be adapted to meet the design objectives of other WSNs in general.

Our ongoing research topics include: (i) developing an adaptive multi-hop routing and transmitting algorithm that will allow sensors to make principled trade-offs between using their energy reserves to take more samples, or to relay data from another sensor to the base-station, (ii) the incorporation of spatial correlations between sensors into both the information metric and the sampling algorithm, such that sensors that are physically close together, and may be taking similar readings,

can autonomously divide the sampling tasks between themselves, and (iii) real deployment in order to investigate how well the algorithm scales and performs in practice.

REFERENCES

- BAR-SHALOM, Y., LI, X. R., AND KIRUBARAJAN, T. 2001. *Estimation with Applications to Tracking and Navigation*. ISBN: 047141655X. Wiley Interscience.
- BISHOP, C. M. 2006. *Pattern Recognition and Machine Learning*. ISBN: 0387310738. Springer.
- CARDELL-OLIVER, R., SMETTEM, K., KRANZ, M., AND MAYER, K. 2005. A reactive soil moisture sensor network: Design and field evaluation. *International Journal of Distributed Sensor Networks* 1, 2 (April-June), 149–162.
- CHEN, C. S., KANG, M. S., HWANG, J. C., AND HUANG, C. W. 2000. Application of binary integer programming for load transfer of distribution systems. In *PowerCon 2000 - International Conference on Power System Technology*. Vol. 1. 305–10.
- CHEVALEYRE, Y., DUNNE, P., ENDRISS, U., LANG, J., LEMAITRE, M., MAUDET, N., PADGET, J., PHELPS, S., RODRIGUEZ-AGUILAR, J., AND SOUSA, P. 2006. Issues in multiagent resource allocation. *Informatica* 30, 3–31.
- CHINTALAPUDI, K., FU, T., PAAK, J., KOTHARI, N., RANGWALA, S., CAFFREY, J., GOVINDAN, R., JOHNSON, E., AND MASRI, S. 2006. Monitoring civil structures with a wireless sensor network. *IEEE Internet Computing* 10, 2 (March-April), 26–34.
- CHU, M., HAUSSECKER, H., AND ZHAO, F. 2002. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. *International Journal of High Performance Computing Applications* 16, 3, 293–313.
- CRESSIE, N. A. C. 1991. *Statistics for spatial data*. John Wiley & Sons.
- DANG, T., BULUSU, N., FENG, W., FROLOV, S., AND BAPTISTA, A. 2007. Adaptive sampling in the columbia river observation network. In *Proceedings of the 5th ACM Conference on Embedded Networked Sensor Systems - ACM Sensys 2007*. Sydney, Australia, 429–30.
- DE ROURE, D. 2005. Floodnet: A new flood warning system. *Ingenia* 23, 49–51.
- ERTIN, E. 2007. Gaussian process models for censored sensor readings. In *Proceedings of the 14th IEEE International Workshop on Statistical Signal Processing - SSP 2007*. 665–9.
- FRIEDEN, B. 2004. *Science from Fisher Information: A Unification*. ISBN: 0521009111. Cambridge University Press.
- GIRARD, A. 2004. Approximate methods for propagation of uncertainty with gaussian process models. Ph.D. thesis, University of Glasgow, Scotland, UK.
- GUESTRIN, C., KRAUSE, A., AND SINGH, A. P. 2005. Near optimal sensor placements in gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning - ICML 2005*. 265–272.
- HE, T., KRISHNAMURTHY, S., LUO, L., YAN, T., GU, L., STOLERU, R., ZHOU, G., CAO, Q., VICAIRE, P., STANKOVIC, J. A., ABDELZAHER, T. F., HUI, J., AND KROGH, B. 2006. Vigilnet: An integrated sensor network system for energy-efficient surveillance. *ACM Trans. Sensor Networks* 2, 1 (February), 1–38.
- HEEKS, R. 1999. Centralised versus decentralised management of public information systems: A core-periphery solution. Tech. rep., Institute of Development Policy and Management, Paper 7, Manchester, UK. January.
- KANSAL, A., HSU, J., ZAHEDI, S., AND SRIVASTAVA, M. B. 2007. Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems* 6, 4.
- KIMURA, N. AND LATIFI, S. 2005. A survey on data compression in wireless sensor networks. In *Proceedings of the ITCC 2005 International Conference on Information Technology: Coding and Computing*. Vol. 2. 8–13.
- KRAUSE, A., GUESTRIN, C., GUPTA, A., AND KLEINBERG, J. 2006. Near-optimal sensor placements: maximizing information while minimizing communication cost. *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, 2–10.

- KROC, S. AND DELIC, V. 2003. Personal wireless sensor network for mobile health care monitoring. In *6th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services - TELSIKS 2003*. Vol. 2. 471–4.
- LEDECZI, A., NADAS, A., VOLGYESI, P., BALOGH, G., KUSY, B., SALLAI, J., PAP, G., DORA, S., MOLNAR, K., MAROTI, M., AND SIMON, G. 2005. Countersniper system for urban warfare. *ACM Trans. Sensor Networks* 1, 2 (November), 153–77.
- LO, B. P. L. AND YANG, G. Z. 2005. Key technical challenges and current implementations of body sensor networks. In *2nd International Workshop on Wearable and Implantable Body Sensor Networks - BSN 2005*. 1–5.
- MACKAY, D. J. C. 1998. Introduction to gaussian process. In *Proceedings of Neural Networks and Machine Learning*. 133–65.
- MAINLAND, G., PARKES, D. C., AND WELSH, M. 2005. Decentralised, adaptive resource allocation for sensor networks. In *Proceedings of the 2nd USENIX/ACM Symposium on Networked Systems Design and Implementation - NSDI 2005*. 315–28.
- MAINWARING, A., POLASTRE, J., SZEWCZYK, R., CULLER, D., AND ANDERSON, J. 2002. Wireless sensor network for habitat monitoring. In *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications*. 88–97.
- MAKARENKO, A. AND DURRANT-WHYTE, H. 2004. Decentralized data fusion and controls in active sensor network. In *Proceedings of the 7th International Conference on Information Fusion*. Stockholm, Sweden, 479–86.
- OSBORNE, M. A., ROGERS, A., RAMCHURN, S., ROBERTS, S. J., AND JENNINGS, N. R. 2008. Towards real-time information processing of sensor network data using computationally efficient multi-output gaussian processes. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN 2008)*. 109–120.
- PADHY, P., DASH, R. K., MARTINEZ, K., AND JENNINGS, N. R. 2006. A utility-based sensing and communication model for a glacial sensor network. In *Proceedings of the 5th International Conference on Autonomous Agents and Multi-Agent Systems*. Hakodate, Japan, 1353–1360.
- PADHY, P., MARTINEZ, K., RIDDOCH, A., HART, J. K., AND ONG, R. 2005. Glacial environment monitoring using sensor networks. In *Proceedings of Real-World Wireless Sensor Networks - REALWSN 2005*. Stockholm, Sweden, 10–14.
- RABINOWITZ, P. AND DAVIS, P. J. 2006. *Methods of Numerical Integration*, 2nd ed. ISBN: 0486453391. Dover Publications.
- RAHIMI, M., PON, R., KAISER, W. J., SUKHATME, G. S., ESTRIN, D., AND SRIVASTAVA, M. 2004. Adaptive sampling for environmental robotics. In *2004 IEEE International Conference on Robotics and Automation*. Vol. 4. 3537–44.
- RASMUSSEN, C. E. 2004. Gaussian processes in machine learning. *Artificial Intelligence* 3176, 63–71.
- RASMUSSEN, C. E. AND WILLIAMS, C. K. I. 2006. *Gaussian Processes for Machine Learning*. ISBN: 026218253X. The MIT Press.
- SEEGER, M. 2004. Gaussian processes for machine learning. *International Journal of Neural Systems* 14, 2 (April), 69–106.
- WERNER-ALLEN, G., LORINCZ, K., RUIZ, M., MARCILLO, O., JOHNSON, J., LEES, J., AND WELSH, M. 2006. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing* 10, 2 (March-April), 18–25.
- WILLETT, R., MARTIN, A., AND NOWAK, R. 2004. Backcasting: adaptive sampling for sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*. 124–33.
- YONG, B. C., KUROKAWA, T., TAKEFUJI, Y., AND HWA, S. K. 1993. An $o(1)$ approximate parallel algorithm for the n-task-n-person assignment problem. In *Proceedings of 1993 International Joint Conference on Neural Networks - IJCNN 1993*. Vol. 2. Nagoya, 1503–6.
- ZHAO, F. AND GUIBAS, L. 2004. *Wireless Sensor Networks: An Information Processing Approach*. Morgan Kaufmann.

Received Month Year; revised Month Year; accepted Month Year