# Extremum Tracking in Sensor Fields with Spatio-temporal Correlation

Prithwish Basu
Raytheon BBN Technologies

Abhishek Nadamani
Cornell University

Lang Tong
Cornell University

***Abstract*—Physical phenomena such as temperature, humidity, and wind velocity often exhibit both spatial and temporal correlation. We consider the problem of tracking the extremum value of a spatio-temporally correlated field using a wireless sensor network. Determining the extremum at the fusion center after making all sensor nodes transmitting their measurements is not energy-efficient because the spatio-temporal correlation of the field is not exploited. We present an optimal centralized algorithm that utilizes the aforementioned correlation to not only minimize the number of transmitting sensors but also ensure low tracking error with respect to the actual extremum. We use recent order statistics bounds in the formulation of the cost function. Since the centralized algorithm has high time complexity, we propose a suboptimal distributed algorithm based on a modified cost function. Our simulations indicate that a small fraction of sensors is often sufficient to track the extremum, and that the centralized algorithm can achieve about $70\%$ energy savings with almost perfect tracking. Furthermore, the performance of the distributed algorithm is comparable to that of the centralized algorithm with up to $25\%$ more energy expenditure.**

## I. Introduction

Recent advances in embedded sensing and wireless communications and networking technologies has resulted in the proliferation of low power sensor devices that are capable of sharing sensed information with each other over a wireless medium and thus forming a wireless sensor network. A common application of wireless sensor networks is in the domain of environmental monitoring; in particular, they are useful for sensing and tracking variations in physical phenomena such as temperature, pressure, humidity, wind velocity etc. over a geographical area. Wireless sensor nodes usually have low power RF transceivers that regularly transmit sensed data either directly or over multiple hops to the fusion center (FC) which supports user queries on gathered data samples.

A major concern that plagues sensor networks is the limited battery life on sensor nodes since RF communication and idle listening have a significant drain on battery. Since sensors can rarely be recharged once deployed in a remote environment, extending battery life is of paramount importance. Therefore, it is prudent to develop energy efficient mechanisms for using these sensors and thus increase the lifetime of the sensor network. In this paper we consider the problem of accurately monitoring a spatio-temporally correlated physical phenomenon in an energy-efficient manner. We assume that there are a finite number of static sensor nodes that can sense the phenomenon at their locations – we refer to this discretized sampling of a continuous phenomenon as a *sensor field*. Specifically, we are interested in tracking the *maximum* value in the sensor field over time.

In this paper, we focus on the single hop network scenario where all sensors and fusion center (FC) are within transmission range of each other. This is interesting in monitoring applications that involve sensors spread across a geographical area, e.g., rooftop of a building or a field. Typically, FC is connected to a continuous power source whereas the sensor nodes are battery-powered. If the maximum is to be determined at a particular instant of time, then all sensors could transmit their measurements to FC which would then determine the maximum value of the sensor field. However, if there exist spatio-temporal correlations in the sensed data (as seen in data obtained from a weather sensor network testbed at BBN), we show that it is possible to exploit such correlations and come up with an optimal sensor selection policy that saves energy while accurately tracking the maximum value.

We first propose a centralized algorithm that is executed at FC. It determines *which* sensor nodes should be transmitting their sensed measurements in the next epoch. In this algorithm, FC maintains recent history of readings from various sensors and attempts to minimize an objective function that captures both energy consumption and the expected value of *deviation from maximum*, if only a subset of nodes were transmitting. Minimization of this objective function yields low tracking error even if only a small fraction of sensor nodes transmit their measurements.

We also propose a distributed algorithm in which the transceivers of sensor nodes are either in ON or OFF state. When in ON state, they overhear measurements reported by a subset of other transmitting nodes and then make a decision about whether to transmit their measurements in the next epoch or to sleep. In OFF state, the nodes do not sense, transmit, or receive. Every node independently attempts to minimize an objective function that is similar to the one used in the centralized algorithm.

**Related Work:** Boulis et al. propose a heuristic mechanism for performing energy-efficient aggregation in sensor networks [4]. Unlike our proposed technique, their scheme to determine the "maximum" does not take into account the

Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

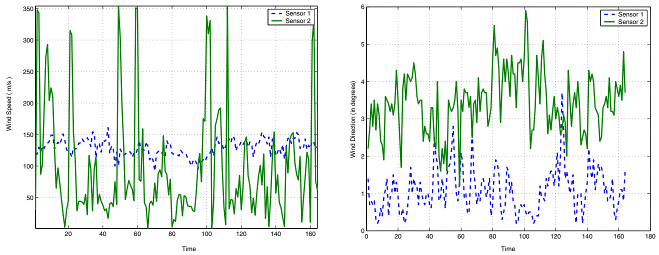

Fig. 1. Wind velocity measurements at two sensors for one hour: (a) Wind Speed; (b) Wind Direction

spatio-temporal correlations of the phenomenon.

Fuhrmann and Widmer address the problem of determining the maximum (or minimum) in a network with large multicast groups [12], [5]. Their objective is to minimize the number of transmissions for determining the maximum using multicast feedback. The data measured at nodes is assumed to be independent, and the algorithm terminates after it has computed the maximum value. On the other hand, our proposed approach exploits the spatio-temporal correlation in the sensor field and continuously tracks the extremum.

Many researchers have investigated exploiting spatial correlation for wireless sensor network applications [11], [8]. Borrowing terminology of [4], these applications correspond to "snapshot aggregation" and do not take into account the temporal characteristics of the sensed data.

Akyildiz et al. emphasize the importance of spatial and temporal correlation in designing MAC protocols for wireless sensor networks [1]. But they consider the spatial and temporal correlation independently and not together. Moreover, their analysis on how spatial and temporal correlation can be used for MAC protocol design is mostly qualitative.

Like our proposed schemes, Mergen et. al. [7] have also considered the case where the sensors communicate directly to FC. Their analysis shows that the cost of listening can be a dominant factor in a dense network. They have proposed wake up schemes and have determined the capacity of such systems from an information theoretic perspective.

## II. Spatio-temporal Correlation in Sensor Data

In order to verify our intuition about spatio-temporal correlation in physical phenomena, we gathered weather data from a sensor network testbed recently deployed on the rooftops of various buildings at BBN. Each node consists of an embedded PC, a 802.11a/b/g interface, and various sensors for monitoring weather conditions as well as air pollutants. The sensors can detect weather measurements, such as wind speed and direction, temperature, air pressure, relative humidity, and rainfall. We collected pressure, temperature, humidity, wind speed and direction data for two sensors in the testbed (located approximately 85 meters apart) over a period of 12 hours.These sensors currently transmit the monitored data to FC periodically. Since they are connected to AC power outlets, these sensors have abundant energy resources, and energy efficiency is currently not an issue in the testbed. However, in the foreseeable future, some of these nodes may be operating on battery while monitoring physical phenomena such as maximum temperature or maximum wind velocity. Therefore, if all sensors transmit their sensed data it would result in unnecessary energy expenditure and would reduce the lifetime of the sensor network significantly. If however, the sensors could learn the spatio-temporal statistics of the sensed data and adapt themselves such that only certain sensors transmit, then we can determine and track the extremum values of the underlying fields with high accuracy.

Figure 1 shows wind velocity measurements; the maximum values could be determined with sensor 2 transmitting most of the time and sensor 1 transmitting at certain times. Therefore, it makes sense to use the spatio-temporal correlation characteristics of the sensor field in order to make decisions about which sensors should transmit with the objective of saving energy while not sacrificing accuracy significantly.

Stochastic modeling of weather data is a reasonably mature discipline. In particular, modelling wind velocity is of great importance in civil engineering from a structural engineering point of view [10], [2]. It is used to forecast maximum wind speed to determine the worst case wind load for a structure. It is also required in wind energy production systems for forecasts of power which are generally derived from forecasts of speed. Researchers have explored the possiblities of modelling wind speed data with a first order Markov chain model [10]. Others have proposed techniques for forecasting wind speed, based on cross correlation at neighboring sites [2]. Accurate modeling of wind direction is important in coastal applications for determining worst case directional load on the structure.

We modeled the data collected from our testbed as a first order Spatio Temporal Auto Regressive (STAR) model. In particular, the first order spatio temporal process can be written as $X_1[n] = \phi_{10} X_1[n-1] + \phi_{11} X_2[n-1] + \epsilon[n]$, where $\phi_{10}$ and $\phi_{11}$ are called the space-time partial autocorrelation functions. $\phi_{10}$ measures the extent of temporal correlation whereas $\phi_{11}$ measures spatial correlation. $X_i[n]$ is the measured value at sensor $i$ at time $n$. We used techniques in [9] for estimating the parameters for the spatio-temporal process measured from the testbed. Specifically for the wind speed data, $(\phi_{10}, \phi_{11}) = (0.6305, 0.2568)$ and for the wind direction data, $(\phi_{10}, \phi_{11}) = (0.9535, 0.0260)$. From these correlation coefficients, we observe that wind speed has both spatial and temporal correlation but wind direction is mostly temporally correlated with little spatial correlation.

## III. Extremum Tracking Problem under Spatio-temporal Correlation

We consider the scenario in which $N$ sensors, distributed randomly over an area $\mathcal{A}$, are attempting to sense a spatio-temporal field at discrete time instants. They transmit the results 1to FC, which then determines the extremum value at each time instant $n$.

The basic tradeoff here that should be exploited is the following: if more sensors transmit, the tracking error would

TABLE I
MATHEMATICAL NOTATION

| Symbol | Description |
|---|---|
| $n$ | Discrete instant (or epoch) of time. We use this convention instead of the usual $t$ because the latter is typically used for modeling continuous time. |
| $X_i(n) = V(x_i, y_i, n)$ | Random variable that indicates the value of the field being measured at node $i$ at time instant $n$. $(x_i, y_i)$ is the location of sensor $i$. |
| $\mathbf{X}[n] = [X_1(n), X_2(n), \ldots X_N(n)]$ | Vector of measurements at the sensors at time $n$. |
| $U_i$ | Random variable associated with sensor $i$ that indicates sensor selection; $U_i = 1$ if sensor $i$ transmits; otherwise, $U_i = 0$. |
| $\mathbf{U} = [U_1, U_2, \ldots U_N]$ | Random variables for transmission policy ($\mathbf{u} = [u_1, u_2, \ldots u_N]$ is a specific realization). |
| $\mathbf{H}[n]$ | Historical information available at the fusion center FC at time instant $n$. This includes information received by FC *before* time $n$. $\mathbf{H}[n] = [\mathbf{H}[n-1], X_i[n]$ s.t. $u_i = 1]$. |
| $\delta_{\mathbf{u}} = \Pr(\mathbf{U} = \mathbf{u} \vert \mathbf{H}[n])$ | Conditional probability of a particular transmission policy $\mathbf{u}$ being selected given the historical measurements. The optimal policy could be randomized, which is given by $\delta_{\mathbf{u}}$. Note that $\sum_{\mathbf{u}} \delta_{\mathbf{u}} = 1$. |
| $M[n] = \max(X_1[n], X_2[n], \ldots X_N[n])$ | Maximum value of the measurements from all sensors. This refers to the best possible tracking. |
| $F[n] = \max(X_i[n]$ s.t. $U_i = 1)$ | Maximum value of the measurements received from the sensors who were tasked to transmit by the algorithm and is a function of $\mathbf{X}[n]$ and $\mathbf{U}$. $F_{\mathbf{u}}[n]$ denotes the measured maximum value if the sensors were following transmission policy $\mathbf{u}$. |
| $\mathcal{E}$ | Energy consumed by a sensor when it transmits. For the sake of simplicity we assume equal transmission power. It would be straightforward to extend it to a case where the transmission power depends on the distance to FC. |
| $k(\mathbf{u})$ | Number of ones in the vector $\mathbf{u}$ (denotes the number of sensors transmitting) |

be minimized but the energy consumption would be high. On the contrary, if too few sensors transmit, the energy consumed would be low, but we would run the risk of erroneous tracking. Therefore, our goal is to develop an algorithm that minimizes both the tracking error and the energy using the measured spatio-temporal characteristics of the sensed field. If we know the spatio-temporal correlation structure, current measurements can be used to predict future values and then a decision can be made by the algorithm about which sensors should transmit.

In this paper we consider both centralized and distributed versions of the problem. In the centralized version of the problem, FC keeps history of all measurements from all sensors that it has received so far. At time instant $n$, FC decides which of the $N$ sensors need to transmit after examining the previously received data, and *tasks* those selected sensors to transmit in the next epoch. Upon receiving the measurements from the chosen sensors in the next epoch, FC determines the maximum and repeats the process. In the distributed version of the problem, FC does not make a decision about which sensor should transmit; instead the sensor nodes themselves make a decision about whether to transmit after overhearing transmissions from other nodes and exploiting the spatio-temporal correlations in the sensed field.

We introduce some useful notation in Table I for the description of our algorithms.

## IV. CENTRALIZED TRACKING ALGORITHM

We assume that each sensor can transmit once in an information gathering epoch. The epochs are assumed to be equal in time duration. At the beginning of each information gathering epoch $n$, FC uses the previously received measurements (in the $n-1^{st}$ epoch) to make a decision about which sensors should transmit their measurements in the current epoch. Upon receiving this information, the requisite sensors transmit to FC.

We assume a slotted TDMA style MAC protocol in operation. FC uses a broadcast slot to communicate the information about "which sensors transmit next" to all sensors listening in that slot. Along with this information, FC also specifies a slot schedule for "which sensors transmits when" in the current epoch. Such a contention-free scheme allows sensors to put their transceivers to sleep at all times except when it is their turn to transmit. In addition to mitigation of packet losses due to collisions, this generally results in tremendous energy savings because of the reduction in idle listening. FC, however, listens to all ongoing transmissions in the network.

For this analysis we assume a perfect channel with no transmission errors or data loss due to fading or time asynchrony due to clock skews. FC chooses the maximum among the received values and denotes it as the maximum of the entire field. FC also updates its history $\mathbf{H}[n]$ with the received values.

**Objective Function:** At the beginning of every epoch, FC depending on the history gathered so far, chooses the sensors that ensure minimum deviation from an estimate of the actual maximum in the epoch. If all sensors transmit, the tracking error would be zero but we should choose the minimum number of sensors possible for reducing energy consumption. We capture the competing goals of minimizing both energy consumption and the tracking accuracy by using the following linear cost function as the objective function for the optimization problem ($\lambda$ is a multiplier that weighs the relative importance of energy efficiency and tracking error):

$$\lambda \times \text{Energy consumed} + (1-\lambda) \times \text{Tracking Error}$$

Since FC may not have access to the measurements ($\mathbf{X}$) before making a decision, we attempt to minimize the expected value of the deviation to account for all possible realizations of $\mathbf{X}$. Also the optimal policy $\mathbf{U}$ may be probabilistic. So we take the expectation over $\mathbf{U}$ as well. Our optimization problem then becomes the following:

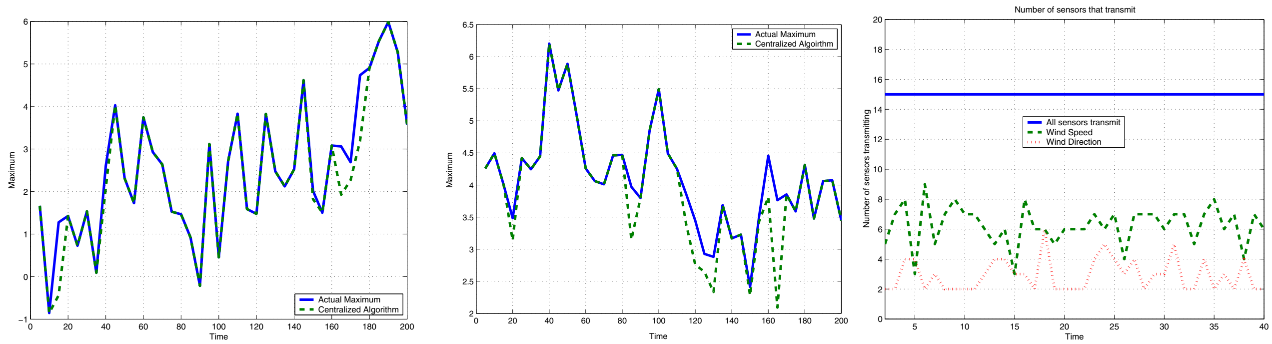


Fig. 2. Tracking the Maximum using the Centralized Algorithm: (a) Wind Speed; (b) Wind Direction; (c) Energy consumption

$$\min_{\delta} \{\lambda \times \mathbb{E}_{\mathbf{U},\mathbf{X}}(E_{total}|\mathbf{H}[n]) + (1-\lambda) \times \mathbb{E}_{\mathbf{U},\mathbf{X}}(M[n] - F[n]|\mathbf{H}[n])\} \quad (1)$$

subject to constraint $\sum_{\mathbf{u}} \delta_{\mathbf{u}} = 1$, where $E_{total}$ denotes the total energy consumed by all sensors and $\mathbb{E}_{\mathbf{U},\mathbf{X}}$ denotes the expectation over all possible realizations of $\mathbf{U}$ and $\mathbf{X}$ given the history $\mathbf{H}[n]$.

**Theorem 1.** *The cost function given in Equation 1 is minimized by a deterministic policy given by*

$$\begin{aligned} \delta_{\mathbf{u}} &= 1, \text{ if } \mathbf{u} = \mathbf{u}_{opt} \\ &= 0, \text{ otherwise} \end{aligned} \quad (2)$$

*where*

$$\begin{aligned} \mathbf{u}_{opt} &= \arg\min_{\mathbf{u}} \{\lambda \times k(\mathbf{u}) \times \mathcal{E} \\ &\quad + (1-\lambda) \times \mathbb{E}_{\mathbf{U},\mathbf{X}}(M[n] - F_{\mathbf{u}}[n]|\mathbf{H}[n])\} \end{aligned} \quad (3)$$

*$F_{\mathbf{u}}[n]$ is the maximum among the readings of the sensors which transmit as indicated by $\mathbf{u}$.*

*Proof:* Consider the first term

$$\lambda \times \mathbb{E}_{\mathbf{U},\mathbf{X}}(E_{total}|\mathbf{H}[n]) = \lambda \times \sum_{\mathbf{u}} \delta_{\mathbf{u}} \times k(\mathbf{u}) \times \mathcal{E}$$

$\mathbb{E}_{\mathbf{U},\mathbf{X}}(M[n]|\mathbf{H}[n])$ is independent of $\mathbf{U}$ and hence does not depend on $\delta_{\mathbf{u}}$.

$$\begin{aligned} \mathbb{E}_{\mathbf{U},\mathbf{X}}(F[n]|\mathbf{H}[n]) &= \sum_{\mathbf{u}} \int_{\mathbf{x}} F[n]\, p(\mathbf{u},\mathbf{x}|\mathbf{H}[n])\, d\mathbf{x} \\ &= \sum_{\mathbf{u}} \int_{\mathbf{x}} F[n]\, p(\mathbf{x}|\mathbf{H}[n])\, \delta(\mathbf{u}|\mathbf{H}[n])\, d\mathbf{x} \\ &= \sum_{\mathbf{u}} \delta_{\mathbf{u}}\, \mathbb{E}_{\mathbf{X}}(F_{\mathbf{u}}[n]|\mathbf{H}[n]) \end{aligned}$$

This can be formulated as a linear programming problem of the form: Minimize $\sum_{\mathbf{u}} \delta_{\mathbf{u}} C_{\mathbf{u}}$, subject to $\sum_{\mathbf{u}} \delta_{\mathbf{u}} = 1$, where $C_{\mathbf{u}} = \lambda \times k(\mathbf{u}) \times \mathcal{E} + (1-\lambda) \times \mathbb{E}(M[n] - F_{\mathbf{u}}[n]|\mathbf{H}[n])$. By the Fundamental Theorem of Linear Programming, the solution to this optimization problem is attained at the end points of the $2^N - 1$ simplex formed by the $\delta_{\mathbf{u}}$'s [6]. Hence Equation 2 yields the optimal solution. ■

**Exploiting the Correlation Model:** We assume that the measured data are jointly zero-mean gaussian r.v.'s with the following correlation structure:

$$\mathbb{E}(X_i[n_1]X_j[n_2]) = \sigma^2 e^{-B|n_1-n_2|} e^{-Ad_{ij}} \quad (4)$$

where $n_1$ and $n_2$ denote discrete instants of time, $d_{ij}$ is the distance between sensor $i$ and $j$, and $\sigma^2$ is the variance. $A$ and $B$ control the degree of correlation between the measured data samples – higher $A$ implies low spatial correlation and vice-versa. Similarly, $B$ controls temporal correlation. If the measured data is spatially uncorrelated, (e.g., wind direction),

$$\mathbb{E}(X_i(n_1)X_j(n_2)) = \sigma^2 e^{-B|n_1-n_2|} \delta_{ij} \quad (5)$$

where $\delta_{ij}$ is the kronecker delta, then it is sufficient to keep the most recently received values in the history. We use this recent history information to estimate the conditional mean and the variance of the next measurement. Since the measurements are spatially uncorrelated, any measurements at sensor 1 (without loss of generality) from other sensors do not affect the conditional mean or the variance at sensor 1. This can be shown easily. For current time instant $n$ and historical time instants $n_1 > n_2 > \ldots n_N$, we have:

$$\begin{aligned} &\mathbb{E}(X_1[n]|\text{all history from all sensors}) \\ &= \mathbb{E}(X_1[n]|X_1[n_1], X_1[n_2], \ldots X_1[n_N]) \quad (6) \\ &= \mathbb{E}(X_1[n]|X_1[n_1]) \quad (7) \end{aligned}$$

Equation 6 follows because the measurements are spatially uncorrelated. Equation 7 follows due to the fact that for each sensor, the temporal measurements form a 1-D Gauss Markov Random Process (as given by correlation structure of Eq. 5).

We now explain how we use historical information to affect the expected value of the highest order statistic. Given two jointly gaussian random variables $\mathbf{X}$ and $\mathbf{Y}$ we have:

$$\begin{aligned} \mathbb{E}(\mathbf{X}|\mathbf{Y}) &= \Sigma_{\mathbf{XY}} \Sigma_{\mathbf{YY}}^{-1} \mathbf{Y} \quad (8) \\ \text{Var}(\mathbf{X}|\mathbf{Y}) &= \Sigma_{\mathbf{XX}} - \Sigma_{\mathbf{XY}} \Sigma_{\mathbf{YY}}^{-1} \Sigma_{\mathbf{YX}} \quad (9) \end{aligned}$$

where $\Sigma_{\mathbf{XX}}, \Sigma_{\mathbf{XY}}, \Sigma_{\mathbf{YX}}$ and $\Sigma_{\mathbf{YY}}$ are covariance matrices.

The determination of the optimal policy requires the evaluation of the expected value of the highest order statistic conditioned on the history. The bound on the expected order statistic can then be evaluated by calculating the conditional mean and conditional variance of the random variables, conditioned on the history. We can determine this by using Equations 8 and 9, where $\mathbf{Y} = \mathbf{H}[n]$ and $\mathbf{X} = X_i[n]$ for all $i$. We use known order statistics bounds to compute the expected value of the maximum [3].

Note that for the case where the measured data is spatio-temporally correlated, from a theoretical point of view, only the most recently received data is not sufficient for the history. However, from a practical point of view, we observed from our simulation experiments (presented later) that using the most recent history alone yields good tracking performance.

**Tracking Algorithm at FC:** The algorithm for tracking the maximum of a spatio-temporally correlated sensor field can broken down into the start-up phase and subsequent phases.

Initialization phase ($n = 0$):

1) All sensors transmit their measurements to FC
2) $F(0) = \max(X_1[n], X_2[n], \ldots X_N[n])$
3) $\mathbf{H}[0] = [X_1[0], X_2[0], \ldots X_N[0]]$

Subsequent phases ($n > 0$):

1) FC determines which sensors have to transmit according to the optimal policy $\mathbf{u}_{opt}$ given $\mathbf{H}[n]$ (Equation 3).
2) $F[n] = \max(X_i[n] \text{ s.t. } u_i = 1)$
3) $\mathbf{H}[n+1] = [\mathbf{H}[n]\,,\, X_i[n] \text{ s.t. } u_i = 1]$

**Simulation Results:** In the simulations of the centralized algorithms, 15 sensors are distributed randomly in a $80m \times 80m$ square. They sense a zero-mean jointly Gaussian field with the correlation structure of Equation 4. We assume that each sensor requires unit energy to transmit to FC. The energy consumed in every epoch is therefore the number of sensors transmitting in that epoch.

We used the partial space time autocorrelation functions determined by the experimental data presented in Section II. $\phi_{10}$ is the correlation coefficient for a temporal lag of one unit and no spatial lag. $\phi_{11}$ is the correlation coefficient for a temporal lag of one unit and a spatial lag of one. Hence, from Equation 5, we have: $e^{-B\Delta n} = \phi_{10}$ and $e^{-Ad_{12}}e^{-B\Delta n} = \phi_{11}$, where $\Delta n$ is the time difference between the samples and $d_{12}$ is the distance between the two sensors.

Wind speed data has both spatial and temporal correlation; we have determined that $A = 0.0107$ and $0.0922$ respectively. We set $\lambda = 0.5$ and $\sigma^2 = 5$ for these simulations since we wanted to put equal imporance on energy conservation and tracking accuracy. In Figure 2(a), the actual maximum, $M[n]$, and the maximum determined by the centralized algorithm, $F[n]$, is plotted against time. The centralized algorithm tracks the maximum with very little deviation. Our simulations showed energy savings of 62.67% over the time period of 200s in comparison with the policy of all sensors tranmsitting in every time epoch.

For less spatially correlated data like wind direction, $A = 0.044$ and $B = 0.0095$ were respectively derived from the space-time partial autocorrelation functions. In this simulation we chose $\lambda = 0.45$ and $\sigma^2 = 5$. We plotted the actual maximum and the maximum as determined by the centralized algorithm in Figure 2(b). For this 200s simulation run, we achieved an energy savings of nearly 79%.

Higher energy savings can be expected when sensing highly temporally correlated data – if a sensor has sensed the maximum at a particular instant of time, it is likely that in the next epoch the same sensor would continue to report the maximum value. This is because the measurement of this sensor is unlikely to be affected by that of the neighbor due to low spatial correlation.

We now analyze the ability of the algorithm to select the appropriate sensors to transmit. We have used the simulation data of the parameters obtained for the wind direction data.

Figure 2(c) illustrates how the energy consumption of the centralized algorithm varies over time for tracking both wind speed and direction. We observe that, on average, only 6 sensors (out of 15) need to transmit for tracking the maximum wind speed and even fewer number of sensors are needed to track maximum wind direction. This is because the latter phenomenon has a very high degree of temporal correlation, whereas the former phenomenon has much less temporal correlation although it does have some spatial correlation.

## V. Distributed Tracking Algorithm

The centralized algorithm presented in Section IV suffers from the problem of exponential time complexity since it uses a brute force approach to optimization over $2^N$ transmission policies. Hence, we propose a suboptimal distributed algorithm which is a simple extension of our centralized algorithm. In the distributed algorithm, FC does not make a decision about the sensors that should be transmitting. Instead, each sensor node makes that decision locally.

The sensors overhear transmissions and update their local history. We assume that the time is slotted as in Section IV and FC distributes transmission slot-schedules *a priori* to all nodes. Each node knows which slot (in each epoch) to transmit its measurements in. At every time step, each sensor tries to locally minimize the cost function according to the policy defined by Equation 3. Each sensor $S_i$ follows the following rule: if last-received values from certain other sensors $S_{i_1}, \ldots, S_{i_k}$ in $S_i$'s history are greater than $S_i$'s value, then $S_i$ assumes that $S_{i_1}, \ldots, S_{i_k}$ will transmit. If the decision is to transmit, $S_i$ senses and transmits its measurement to FC.

In this distributed algorithm, each sensor node keeps awake in other slots as well in order to overhear the transmissions from other sensors. If the decision is not to transmit, that sensor sleeps for that time epoch. However, if the sensor sleeps for a long period, it may lose track of the state of the field. Hence, we propose that all sensors wake up, listen, and transmit at *a priori* fixed periodic intervals of duration $T$. More formally, when $n = 0$ or $n \bmod T = 0$:

1) All sensors transmit to FC
2) Each sensor updates its history about other sensors after hearing their transmissions to FC
3) FC determines the maximum $M[n]$

In subsequent phases ($n > 0$ and $n \bmod T \neq 0$ ), for every sensor $i$, the following steps are taken:

1) $\mathbf{H}_i[n]$ is the vector denoting history available at sensor $i$ at time $n$. In formal terms, $\mathbf{H}_i[n] = [H_i^1, H_i^2, \ldots, H_i^N]$, where $H_i^j$ is the last received value from sensor $j$ at sensor $i$.

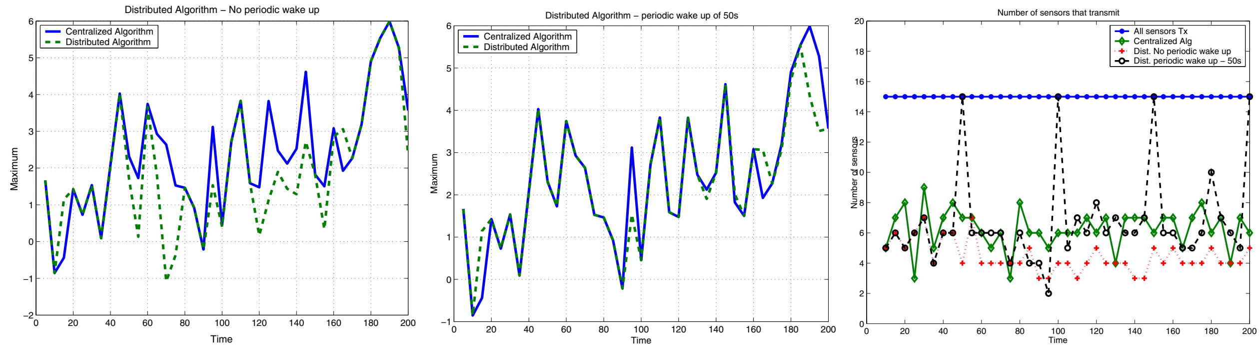


Fig. 3. Tracking the Maximum using the Distributed Algorithm: (a) No periodic wake up; (b) Periodic wakeup (50s); (c) Energy consumption

2) Let $u_j = 1$ if $H_i^j \geq H_i^i$
3) Define $\mathbf{u}_i^1 = [u_1, \ldots, u_{i-1}, 1, u_{i+1}, \ldots, u_N]$ and $\mathbf{u}_i^0 = [u_1, \ldots, u_{i-1}, 0, u_{i+1}, \ldots, u_N]$
4) Every sensor $i$ calculates:

$$\lambda \times k(\mathbf{u}_i^0) \times \mathcal{E} + (1-\lambda) \times \mathbb{E}(M[n] - F_{\mathbf{u}_i^0} | \mathbf{H}_i[n]) \quad (10)$$

$$\lambda \times k(\mathbf{u}_i^1) \times \mathcal{E} + (1-\lambda) \times \mathbb{E}(M[n] - F_{\mathbf{u}_i^1} | \mathbf{H}_i[n]) \quad (11)$$

5) If expession 10 $>$ expression 11, then sensor $i$ transmits.
6) If it transmits, it also listens to the transmissions of the other sensors and updates $\mathbf{H}_i[n+1]$, else it sleeps until the beginning of the next time epoch.

The above distributed algorithm involves computation of only two values (Expressions 10 and 11) at each sensor; therefore, it is much more computationally efficient in comparision with its centralized counterpart which has to make $2^N$ computations at each time epoch.

**Simulation Results:** We simulated the distributed algorithm with the same data (wind direction) that was used for the centralized algorithm, and compared their performance. From Figure 3, we can observe that the distributed algorithm performs almost as well as the centralized algorithm although the total energy consumed is $\approx 25\%$ more than the latter (for periodic wakeup of $T = 25s$). The mean squared error is $0.1026$ which is lower in comparison with $0.1773$ for the centralized algorithm because more sensors transmit in the case of the former. The distributed algorithm is efficient in its operation since it just involves the computation of two values before making a decision about whether to transmit. With minimal extra energy expenditure, we obtain very good tracking performance.

We observe from Figure 3(c) that the distributed algorithm is able to conserve energy almost as significantly as its centralized counterpart. However, the exact energy efficiency depends on the value of wakeup period $T$. For the case where there is no wakeup at all, the energy consumption is quite low (lower than centralized) at the cost of moderate tracking error. For wake up period $T = 50s$, we observe that energy consumption shoots up occasionally but is low overall. Therefore, we conclude that a distributed scheme with a low frequency periodic wakeup has decent energy efficiency and good extremum tracking performance.

## VI. Conclusion

In this paper we show that a single hop wireless sensor network that is continuously tracking the extrema of time-varying phenomena can intelligently exploit the underlying spatio-temporal correlations in order to save precious battery energy and thus extend network lifetime. In particular, we proposed both centralized and distributed versions of an optimization algorithm that simultaneously attempt to minimize the energy consumption as well as the tracking error. We showed that both the centralized and the distributed algorithms could track the maximum of a spatio-temporally correlated field remarkably well over varying degrees of correlation while spending little energy. The algorithms were also adaptive to detect sudden changes in the phenomena.

## References


[1] I. F. Akyildiz, M. C. Vuran, and O. B. Akan, "On Exploiting Spatial and Temporal Correlation in Wireless Sensor Networks," *Proc. WiOpt*, 2004.

[2] M. C. Alexiadis, P. S. Dokopoulos and H. S. Sahsamanoglou, "Wind speed and power forecasting based on spatial correlation models," *IEEE Transactions on Energy Conversion,* September 1999.

[3] D. Berstimas, K. Natarajan and C. Teo, "Tight bounds on Expected Order Statistics," *Probability in the Engineering and Informational Sciences*, 2006.

[4] A. Boulis, S. Ganeriwal and M. Srivatsava, "Aggregation in Sensor Networks: An Energy Accuracy Tradeoff," *Proc. Sensor Network Protocols and Applications*, 2003.

[5] T. Fuhrmann and J. Widmer, "Extremum Feedback with Partial Knowledge," *Lecture Notes in Computer Science*, Springer, 2004.

[6] D. G. Luenberger, "Linear and Non Linear Programming," 2nd Edition, Addison-Wesley, 1989.

[7] G. Mergen, Q. Zhao, and L. Tong, "Sensor Networks with Mobile Access: Energy and Capacity Considerations," *IEEE Transactions on Communications*, vol. 54, no. 11, pp 2033-2044, Nov., 2006.

[8] S. Pattem, B. Krishnamachari, R. Govindan," The impact of spatial correlation on routing with compression in wireless sensor networks", *Symposium on Information Processing in Sensor Networks (IPSN)*, 2004.

[9] P. E. Pfeifer, S. J. Deutsch, "A Three Stage Iterative Procedure for Space-Time Modeling," *Technometrics*, Vol. 22, No. 1, Feb 1980.

[10] A. D. Sahin and Z. Sen, "First-order Markov chain approach to wind speed modelling," *Journal of Wind Engineering and Industrial Aerodynamics*, Volume 89, Issues 3-4, March 2001.

[11] Y. Sung, S. Misra, L. Tong and A. Ephremides, " Cooperative Routing for Signal Detection in Large Sensor Networks," *IEEE JSAC Special Issue on Cooperative Communications and Networking*, vol. 25, no. 2, pp. 471–483, Feb. 2007.

[12] J. Widmer and T. Fuhrmann, "Extremum Feedback for Very Large Multicast Groups," *Lecture Notes in Computer Science*, Springer, 2001.