Signal Processing 128 (2016) 369–381

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Spatio-temporal sensor management for environmental field estimation ☆

Venkat Roy [a,*], Andrea Simonetto [b], Geert Leus [a]

[a] Delft University of Technology, Delft, The Netherlands
[b] Université Catholique de Louvain, Louvain-la-Neuve, Belgium

ARTICLE INFO

Article history:
Received 28 December 2015
Received in revised form
15 April 2016
Accepted 7 May 2016
Available online 10 May 2016

Keywords:
Sensor placement
Sparsity
Wireless sensor network
Environment monitoring

ABSTRACT

We develop sparsity-enforcing spatio-temporal sensor management methods for environmental field monitoring applications. Leveraging the space–time stationarity, an environmental field can be estimated with a desired spatio-temporal resolution based on recorded measurements. If the field is non-stationary, it can be monitored dynamically based on the collected measurements and predictions made through a state model, if known a priori. We develop algorithms to implement sparse sensing, i.e., sensing only the most informative locations in space and time for both spatio-temporally stationary and non-stationary field monitoring applications. The selected sensing locations form an underdetermined measurement model which can be used to estimate the field based on the prior knowledge regarding the space–time variability of the field. The task of locating the most informative sensing locations can be performed for both multiple snapshots and a single snapshot based on the availability of prior knowledge (space–time correlation and dynamics) regarding the field, available computing power and the application. Centralized sensor placement problems for the estimation of both stationary and non-stationary fields are formulated as relaxed convex optimization problems, constrained by static or dynamic performance criteria. Finally, an iterative sparsity-enhancing saddle point method is formulated to solve both of these sensor placement problems.

1. Introduction

Monitoring an environmental field, e.g., rainfall, surface temperature, pollution concentration, humidity over an area, is generally performed by a network of dedicated sensors deployed in an intelligent constellation. The sensitivity of the estimation performance strongly depends on the sensor deployment (static) or movement (dynamic) strategies. In a centralized paradigm, due to the resource-related constraints of the sensors (e.g., bandwidth and life-time), it is always desirable to use a limited number of sensors to perform the sensing task with a required accuracy. Sensor selection promoting sparsity both for linear and non-linear measurement models is extensively studied for field estimation [1], localization [2], and tracking [3] problems. Specifically for spatial field estimation, sparsity-aware kriging [4] and correlation-aware sensor placement [5] promoting sparsity are also proposed. The problem of performance-aware sensor selection constrained by the number of resources has been well-formulated as a convex problem in [6], for statistical A, D, and E optimality criteria [7]. Also a distributed implementation of the sparsity inducing sensor selection problem is presented in [8]. The problem of sensor placement for field estimation has also been solved using tools from network and information theory. In [9], information theoretic approaches are adopted for placing sensors to estimate Gaussian processes, where the submodularity of the mutual information between the sensor locations is exploited. To monitor a dynamic process, periodic sensor activation and deactivation is optimal for an infinite time horizon [10], and it is shown to be approximately optimal for a finite time window [11]. In [12], a sparsity-enforcing sensor scheduling method is presented and applied to monitor a dynamic field.

Computational complexity is an issue for realistic sensor placement problems, where the service area is quite large. As mentioned earlier, the accuracy-constrained sensor selection problem can be formulated as a convex problem and solved using off the shelf solvers like CVX [13] and SeDuMi [14]. But to select sensing locations over a large service area and/or multiple time snapshots these solvers can be computationally inefficient. In [1,12], alternating direction method of multipliers (ADMM) and accelerated proximal gradient method (APGM) algorithms are used to improve the speed of the sensor selection process.

In this work, we select the most informative sensing locations for the estimation of a general class of environmental fields. The main difference of this work with the standard sensor selection literature

☆This work has been supported by the SHINE project, the flagship project of DIRECT (Delft Institute for Research on ICT at the Delft University of Technology).
* Corresponding author.
E-mail address: v.roy@tudelft.nl (V. Roy).

http://dx.doi.org/10.1016/j.sigpro.2016.05.011
0165-1684/

(like [2,6,8,15]) is the primary measurement model, which we consider to be underdetermined. Resorting to the Bayesian philosophy, we exploit the available prior statistical knowledge regarding the unknown field. In the first case, we model the field as a *spatio-temporally stationary* stochastic process. The spatio-temporal covariance structure is considered to be known as prior information. In the second case, the field is considered to be *non-stationary*, where the prior knowledge comes from the known dynamics. For a non-stationary field, we specifically model the spatio-temporal evolution using a state model incorporating some common physical phenomena present in many environmental processes like diffusion and advection [16].

The estimation of the field intensities with a prescribed resolution can be performed offline based on recorded measurements at different locations over multiple snapshots if the field is spatio-temporally stationary. If the field is non-stationary, then first and second order statistics can be computed multiple snapshots ahead based on the available prior statistics and the dynamics of the field. This allows for dynamic estimation of the non-stationary field multiple snapshots ahead.

In both of these scenarios, it is always useful to know the best time/place to deploy the sensors in order to reduce the number of sensors to economize the overall processing time and power. We next briefly elucidate the importance of sensor placement for stationary and non-stationary environmental field estimation applications.

One plausible application of sensor placement for stationary field estimation could be the deployment of rain gauges in an area for long term precipitation monitoring, where stationarity is a valid assumption [17]. A dynamic deployment of sensors is needed for spatio-temporal field tracking applications like robotic sensor networks, social sensing, and mobile sensor networks (sensors deployed on vehicles, bikes) for environment monitoring as well as disaster management. For many such applications, it is essential to know the locations where to deploy/move the sensors in the next snapshots. For the aforementioned applications, a single snapshot or multiple snapshots ahead sensor placement method can be applied for the sensor deployment over upcoming snapshots, if the dynamics for the future snapshots are known a priori.

The main contributions of the paper are,

- We formulate sensor placement problems for both spatio-temporally stationary and non-stationary environmental field estimation as convex optimization problems with similar structures but with different Bayesian performance metrics.
- We develop a mathematical framework to efficiently utilize the spatial/temporal correlation information of the environmental field to optimize the required number of sensing locations.
- We propose a first-order iterative sparsity-enhancing saddle-point method to solve the sensor placement problems.

To enforce sparsity in selecting the optimal sensing locations and time instances, we follow the iterative reweighted ℓ_1 minimization technique [18]. Simulations are carried out to select the optimal sensing locations for different stationary and non-stationary environmental field models.

Organization: We organize the paper in the following way. In Section 2, we describe the measurement model, the main problem statement, and the statistical characterizations of the environmental field. The mean square error (MSE) matrix of a linear minimum mean square error (LMMSE) estimator is derived in Section 3 for both stationary and non-stationary field estimation problems. In Section 4, we formally address the sensor placement problems for both stationary and non-stationary field estimation applications. An iterative sparsity-enhancing saddle point method is formulated in Section 5 to solve the proposed sensor placement problems. Simulation results are presented in Section 6. The final conclusions are drawn in Section 7.

Notations: Matrices are in upper case bold while column vectors are in lower case bold. The notation $[\mathbf{X}]_{ij}$ is the (i,j)-th entry of the matrix $\mathbf{X}$, $[\mathbf{x}]_i$ is the i-th entry of the vector $\mathbf{x}$, and $\mathrm{tr}[\mathbf{X}]$ denotes the trace of $\mathbf{X}$, i.e., the sum of the diagonal elements of $\mathbf{X}$. The notation $\mathrm{supp}(\mathbf{x})$ is defined as the set of indices of the non-zero entries of $\mathbf{x}$, while $\mathrm{diag}(\mathbf{x})$ and $\mathrm{diag}(\mathbf{X})$ are the diagonal matrix with diagonal $\mathbf{x}$ and the main diagonal of the matrix $\mathbf{X}$, respectively. An identity matrix of size $N \times N$ is denoted by $\mathbf{I}_N$. The notation $(\cdot)^T$ is the transpose operator, $\hat{\mathbf{x}}$ is the estimate of $\mathbf{x}$, and $\| \mathbf{x} \|_p = (\sum_{i=0}^{N-1} |[\mathbf{x}]_i|^p)^{1/p}$ is the ℓ_p norm of $\mathbf{x}$. The notation $\triangleq$ defines an entity. Vectors of all zeros and ones of length N are denoted by $\mathbf{0}_N$ and $\mathbf{1}_N$, respectively. An all zero matrix of size $N \times N$ is given by $\mathbf{0}_{N\times N}$. The set of symmetric matrices of size $N \times N$ and the set of symmetric positive-definite matrices of size $N \times N$ are denoted by $\mathbb{S}^N$ and $\mathbb{S}^N_{++}$, respectively.

2. Signal model and problem statement

2.1. Measurement model

We assume a finite uniform pixelation of the entire service area of interest into N pixels, where we would like to estimate the field intensities. The field intensity at N pixels at time index $t = 1, 2, \ldots$ can be represented by $\mathbf{u}_t \in \mathbb{R}^N$. It is assumed that the field intensities are the same everywhere within a pixel. The elements of $\mathbf{u}_t$ are given by $[\mathbf{u}_t]_j = u_t(\mathbf{x}_j)$, for $j = 1, \ldots, N$, where $u_t(\mathbf{x})$ is the continuous function representing the field at time t at any arbitrary position $\mathbf{x} \in \mathbb{R}^2$ and $\mathbf{x}_j \in \mathbb{R}^2$ is the centroid of the j-th pixel.

The measurements are given by $\mathbf{y}_t \in \mathbb{R}^{M_t}$, collected from M_t sensing locations (pixels) of the aforementioned service area. Only a single dimension of $\mathbf{u}_t$ is measured by a sensor deployed at any of the N pixels. The model is compressive as $M_t < N$. The *time-varying linear compressive* measurement model can be constructed as

$$\mathbf{y}_t = \mathbf{C}_t \mathbf{u}_t + \mathbf{e}_t, \tag{1}$$

where the *compressive measurement matrix* $\mathbf{C}_t \in \{0, 1\}^{M_t \times N}$ maps M_t measurements from N pixels in $\mathbf{y}_t$. The measurement matrix can be constructed by $\mathbf{C}_t = \mathrm{diag}_X(\mathbf{w}_t)$, where $\mathbf{w}_t = [w_{t1}, \ldots, w_{tN}]^T \in \{0, 1\}^N$ is the *sensor location selection vector* for time t, and $\mathrm{diag}_X(\mathbf{w}_t)$ removes the zero rows from $\mathrm{diag}(\mathbf{w}_t)$. So, if we have $[\mathbf{w}_t]_j = 1(0)$, then the j-the field location is selected (not selected) for sensor deployment at time t. The measurement matrix $\mathbf{C}_t$ is related to the sensor location selection vector $\mathbf{w}_t$ by the relations

$$\mathbf{C}_t^T \mathbf{C}_t = \mathrm{diag}(\mathbf{w}_t); \quad \mathbf{C}_t \mathbf{C}_t^T = \mathbf{I}_{M_t}. \tag{2}$$

The M_t measurements are corrupted by additive spatio-temporally white Gaussian noise $\mathbf{e}_t \sim \mathcal{N}(\mathbf{0}, \sigma_e^2 \mathbf{I}_{M_t})$, where $\sigma_e{}^2$ is the noise variance. Further, we also assume that $\mathbf{e}_t$ is uncorrelated with $\mathbf{u}_t$.

Any spatio-temporal distribution of the field, i.e., the field intensities at the N pixels for any observation window of N_s snapshots, can be represented by the vector $\tilde{\mathbf{u}}_t = [\mathbf{u}_t^T, \ldots, \mathbf{u}_{t+N_s-1}^T]^T \in \mathbb{R}^{NN_s}$. In this case, the overall measurement model to estimate the field at N locations over any N_s snapshots can be expressed as

$$\tilde{\mathbf{y}}_t = \tilde{\mathbf{C}}_t \tilde{\mathbf{u}}_t + \tilde{\mathbf{e}}_t, \tag{3}$$

where $\tilde{\mathbf{C}}_t = \mathrm{blkdiag}(\mathbf{C}_t, \ldots, \mathbf{C}_{t+N_s-1}) \in \{0, 1\}^{\tilde{M}_t \times NN_s}$, with $\tilde{M}_t = \sum_{\tau=0}^{N_s-1} M_{t+\tau}$ and $\mathrm{blkdiag}(\cdot)$ denoting a block diagonal matrix. The measurements are given as $\tilde{\mathbf{y}}_t = [\mathbf{y}_t^T, \ldots, \mathbf{y}_{t+N_s-1}^T]^T$ of length $\tilde{M}_t$ and the noise components at all snapshots are represented by the vector $\tilde{\mathbf{e}}_t = [\mathbf{e}_t^T, \ldots, \mathbf{e}_{t+N_s-1}^T]^T$ of the same length as $\tilde{\mathbf{y}}_t$. The noise vector $\tilde{\mathbf{e}}_t$ is spatio-temporally white and characterized by $\tilde{\mathbf{e}}_t \sim \mathcal{N}(\mathbf{0}_{\tilde{M}_t}, \sigma_e^2 \mathbf{I}_{\tilde{M}_t})$.

By using the relation (2), we obtain

$$\tilde{\mathbf{C}}_t^T\tilde{\mathbf{C}}_t = \text{blkdiag}(\text{diag}(\mathbf{w}_t), \ldots, \text{diag}(\mathbf{w}_{t+N_s-1})) \tag{4}$$

$$\tilde{\mathbf{C}}_t^T\tilde{\mathbf{C}}_t = \text{diag}(\tilde{\mathbf{w}}_t), \tag{5}$$

where $\tilde{\mathbf{w}}_t = [\mathbf{w}_t^T, \ldots, \mathbf{w}_{t+N_s-1}^T]^T$ is the sensor location selection vector for the N locations in all the N_s snapshots.

2.2. Problem statement

The optimal placement of the sensors at the informative locations can be formulated as a *sensor location selection* problem, i.e., the design of a selection vector $\tilde{\mathbf{w}}_t \in \{0, 1\}^{NN_s}$. However, generally choosing the best subset of sensing locations achieving some desired estimation performance is a combinatorially complex problem. A standard approach to tackle this problem is to relax it into a convex problem, which can be efficiently solved in polynomial time [6,8,2]. In this case, a sparsity-enforcing, performance-constrained design of $\tilde{\mathbf{w}}_t$ can be obtained by solving

$$\hat{\tilde{\mathbf{w}}}_t = \underset{\tilde{\mathbf{w}}_t\in[0,1]^{NN_s}}{\arg\min}\left\{\| \tilde{\mathbf{w}}_t \|_1,\ \text{s.t.}\ g(\tilde{\mathbf{w}}_t) \le \gamma\right\}, \tag{6}$$

where $g(\tilde{\mathbf{w}}_t)$ is a performance metric expressed as a function of the selection vector, and γ is the desired threshold on the performance. After solving (6), we obtain $\tilde{\mathbf{w}}_t \in [0, 1]^{NN_s}$. To generate a Boolean selection vector $\tilde{\mathbf{w}}_t \in \{0, 1\}^{NN_s}$ from $\tilde{\mathbf{w}}_t \in [0, 1]^{NN_s}$, we can adopt the randomized rounding technique of [2] or a simple thresholding. The randomization is done by simply generating random realizations of $\tilde{\mathbf{w}}_t$ with the probability that $[\tilde{\mathbf{w}}_t]_k = 1$ specified by $[\hat{\tilde{\mathbf{w}}}_t]_k$, where $k = 1, \ldots, NN_s$. The realizations satisfying $g(\tilde{\mathbf{w}}_t) \le \gamma$ are selected and the minimum ℓ_0 norm realization is picked up, whose support denotes the sparsest optimal sensor placement scheme.

2.3. Statistical characterization of $\mathbf{u}_t$

In this paper, we consider two statistical characterizations of the field vector $\mathbf{u}_t$.

2.3.1. Stationary field

In the first case, we consider the elements of $\mathbf{u}_t$, i.e., $[\mathbf{u}_t]_j = u_t(\mathbf{x}_j)$ for $j = 1, \ldots, N$, to be Gaussian random variables. We further assume that they are realizations of a spatio-temporally (second-order) stationary isotropic process [16] with mean $\mathbb{E}[u_t(\mathbf{x}_j)] = \mu_s$ for all t and $j = 1, \ldots, N$. The spatio-temporal covariance matrix is derived from a space–time separable exponential covariance function. For any temporal lag τ, i.e., the time difference between the snapshots $\mathbf{u}_t$ and $\mathbf{u}_{t-\tau}$, and any two spatial locations $\mathbf{x}_i, \mathbf{x}_j$, with $d_{ij} \triangleq \| \mathbf{x}_i - \mathbf{x}_j \|_2$, the elements of the spatial covariance matrix for lag τ, which is denoted as $\mathbf{\Gamma}_\tau$, are modeled as

$$\begin{aligned}[\mathbf{\Gamma}_\tau]_{ij} &= \left[\mathbb{E}\left[\left(\mathbf{u}_t - \mu_s\mathbf{1}_N\right)(\mathbf{u}_{t-\tau} - \mu_s\mathbf{1}_N)^T\right]\right]_{ij} \\ &= \sigma_u^2 \exp\left[-\frac{1}{s_h}d_{ij} - \frac{1}{s_\tau}|\tau|\right].\end{aligned} \tag{7}$$

Here, $\mathbb{E}[(u_t(\mathbf{x}_j) - \mu_s)^2] = \sigma_u^2$ and s_h, s_τ are the positive scaling parameters for space and time, respectively. In this work, we assume that the parameters $\sigma_u{}^2$, s_h and s_τ are all known a priori. The nature of the covariance function of (7) with different lags over space and time is shown in Fig. 2a. The overall spatio-temporal covariance matrix can be expressed as $\tilde{\mathbf{\Gamma}}_t = \mathbb{E}[(\tilde{\mathbf{u}}_t - \tilde{\boldsymbol{\mu}}_t)(\tilde{\mathbf{u}}_t - \tilde{\boldsymbol{\mu}}_t)^T] \in \mathbb{S}_{++}^{NN_s}$, where $\tilde{\boldsymbol{\mu}}_t = \mathbb{E}[\tilde{\mathbf{u}}_t] = \mu_s\mathbf{1}_{NN_s}$. The diagonal and off-diagonal blocks of $\tilde{\mathbf{\Gamma}}_t$ are given by

$$\tilde{\mathbf{\Gamma}}_t = \begin{bmatrix} \mathbf{\Gamma}_0 & \cdots & \mathbf{\Gamma}_{-N_s+1} \\ \vdots & \ddots & \vdots \\ \mathbf{\Gamma}_{N_s-1} & \cdots & \mathbf{\Gamma}_0 \end{bmatrix}. \tag{8}$$

It should be noted that, if the field is spatio-temporally uncorrelated then the spatio-temporal covariance matrix is simply given as $\tilde{\mathbf{\Gamma}}_t = \sigma_u^2\mathbf{I}_{NN_s}$.

2.3.2. Non-stationary field

In the second case, we consider $\mathbf{u}_t$ to be a non-stationary environmental field. The spatio-temporal evolution of the environmental field can be described by an integro-difference equation (IDE) [16]. For a specific sampling interval T_s (i.e., the time duration between two consecutive time indices) the discrete time IDE can be represented as,

$$u_t(\mathbf{x}) = \int_{\mathcal{R}_s} f(\mathbf{x}, \mathbf{x}')u_{t-1}(\mathbf{x}')d\mathbf{x}' + q_t(\mathbf{x}), \tag{9}$$

where $\mathcal{R}_s \subset \mathbb{R}^2$ is the service area of interest. The spatio-temporal evolution of $u_t(\mathbf{x})$ is modeled by the function $f(\mathbf{x}, \mathbf{x}')$. Here, $q_t(\mathbf{x})$ is the Gaussian process noise which can be spatially colored but is temporally white. The space–time interaction function $f(\mathbf{x}, \mathbf{x}')$ can be modeled as a time-varying parameterized kernel function $f(\mathbf{x}, \mathbf{x}') = \nu h_t(\mathbf{x}, \mathbf{x}', \boldsymbol{\theta}_t)$, where the parameters of the function, i.e., $\boldsymbol{\theta}_t$, can be deterministic or random. Here ν is a positive scaling parameter to ensure the stability of the process [19]. In this case, the IDE is given as

$$u_t(\mathbf{x}) = \nu\int_{\mathcal{R}_s} h_t(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}_t)u_{t-1}(\mathbf{x}')d\mathbf{x}' + q_t(\mathbf{x}). \tag{10}$$

Note that, the state models in (9) and (10) are infinite dimensional. One way to approximate these to finite dimensional models is by using a spectral representation of $f(\cdot)$ or $h_t(\cdot)$, and $u_t(\cdot)$ using a known orthonormal basis and selecting only the K dominant coefficients [20].

However, here we have considered a finite uniform spatial discretization of the field into N pixels. So, the process model of (10) can be discretized over N pixels by

$$u_t(\mathbf{x}_i) = \nu\sum_{j=1}^{N} h_t(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta}_t)u_{t-1}(\mathbf{x}_j) + q_t(\mathbf{x}_i), \tag{11}$$

where $i = 1, \ldots, N$. Here, we assume that the parameters of the kernel function, i.e., $\boldsymbol{\theta}_t$, are perfectly known and deterministic. It should be noted that the parameters of the kernel function depend upon the temporal sampling interval T_s.

Spatial phenomena like advection and diffusion can be modeled by changing the translation and dilation parameters of a Gaussian kernel [19,21]. More specifically, we consider a time-varying 2D Gaussian kernel

$$h_t(\mathbf{x}_i, \mathbf{x}_j) = \exp[-(\mathbf{x}_i - \mathbf{x}_j - \mathbf{a}_t)^T\mathbf{D}_t^{-1}(\mathbf{x}_i - \mathbf{x}_j - \mathbf{a}_t)], \tag{12}$$

where the translation parameter $\mathbf{a}_t \in \mathbb{R}^2$ and the dilation parameter $\mathbf{D}_t \in \mathbb{S}_{++}^2$ model the advection and the isotropic/anisotropic diffusion, respectively. In this case, the parameter vector of the kernel function, i.e, $\boldsymbol{\theta}_t$ in (11), contains the elements of $\mathbf{a}_t$ and $\mathbf{D}_t$. Here, we model $\mathbf{a}_t$ as the time-varying displacement and $\mathbf{D}_t$ as the time-varying diffusion in every T_s seconds. Note that, $\mathbf{a}_t$ and $\mathbf{D}_t$ can also be varied over space in order to model more complicated dynamics of the field [19]. The directions of anisotropy of diffusion can be incorporated through $\mathbf{D}_t$. Otherwise, isotropic diffusion can be modeled as $\mathbf{D}_t = \kappa_t\mathbf{I}_2$, with $\kappa_t > 0$. For example, like (12), modeling the function $h_t(\cdot)$ as a *Gaussian dispersal kernel* can be used for short term rainfall prediction [21]. The above approach can be generalized to describe the dynamics of many environmental phenomena such as the distribution of pollutants,

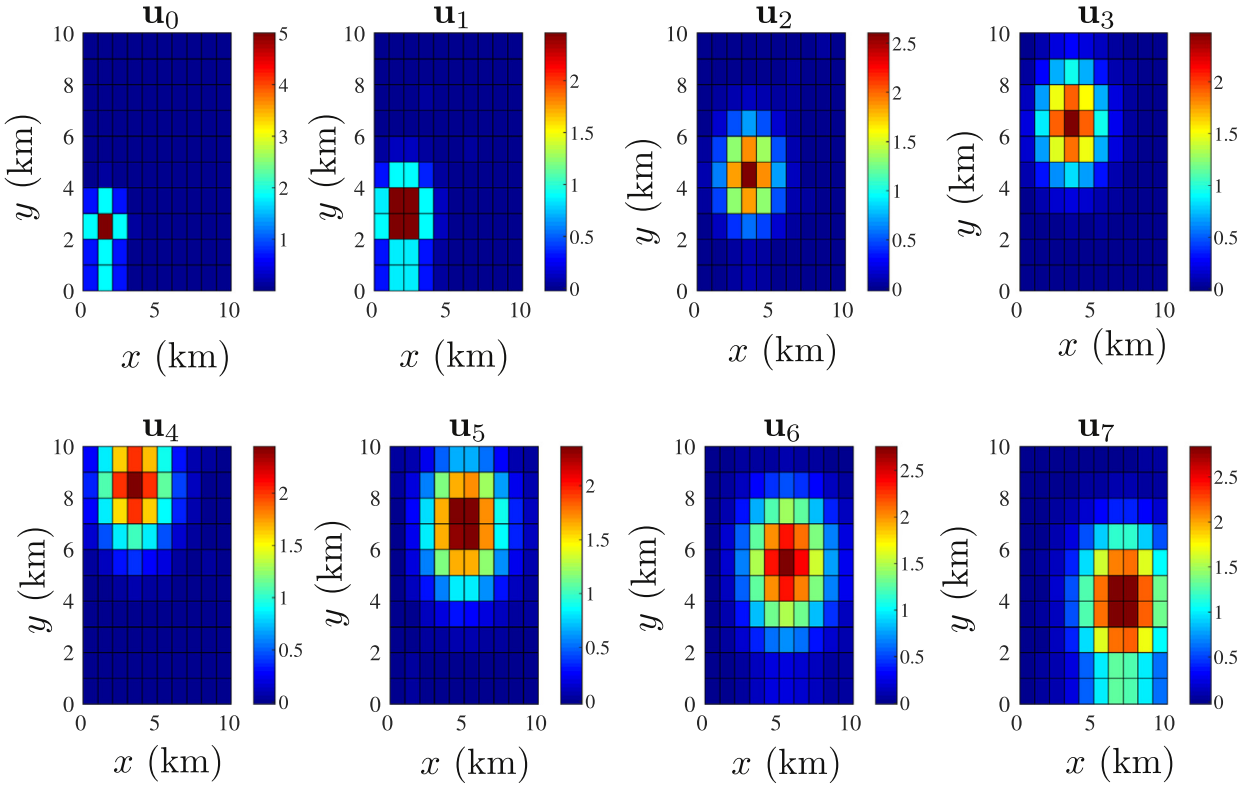

Fig. 1. Spatio-temporal evolution of the field in 10×10 square km area; spatial resolution: 1×1 square km; $\mathbf{D}_t = \mathbf{I}_2$; $\nu = 0.4$; $\mathbf{q}_t \sim \mathcal{N}(\mathbf{0}_{100}, 10^{-3}\mathbf{I}_{100})$; displacement due to advection, i.e., $\mathbf{a}_t$ on every snapshot is given by $[0.5, 0.5]^T$, $[1.5, 1.5]^T$, $[0, 2]^T$, $[0, 2]^T$, $[1.5, -1.5]^T$, $[0.5, -1.5]^T$, $[1.5, -1.5]^T$ where $t = 1, \ldots, 7$ min .

movement of aerosols, vapor concentrations, etc. that possess properties like advection, diffusion, etc.

Let us now assume a state transition matrix $\mathbf{H}_t \in \mathbb{R}^{N\times N}$ which is modeled using a simple 2D Gaussian kernel whose elements are given by $[\mathbf{H}_t]_{ij} = \nu h_t(\mathbf{x}_i, \mathbf{x}_j)$. After proper vectorization of the field intensities and the process noise for the N pixels, the overall state model can be represented as

$$\mathbf{u}_t = \mathbf{H}_t\mathbf{u}_{t-1} + \mathbf{q}_t. \tag{13}$$

Here, $\mathbf{q}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t)$ is the spatially colored yet temporally white Gaussian process noise. In Fig. 1, an example of the spatio-temporal evolution of the field is shown with a time-invariant isotropic diffusion and a time-varying advection. The initial state of $\mathbf{u}_t$, i.e., the state at $t=0$ is generated by a simple Gaussian function. In Fig. 1, it is seen that the field is isotropically diffused as well as shifted in different directions given by the advection vector $\mathbf{a}_t$ changing with time t.

In this case, the field is statistically characterized by the dynamics as

$$p(\mathbf{u}_t|\mathbf{u}_{t-1}) \sim \mathcal{N}(\mathbf{H}_t\mathbf{u}_{t-1}, \mathbf{Q}_t). \tag{14}$$

The N_s snapshots ahead first and second order statistics of the field can be derived using the state model. In this case, the N_s snapshots ahead mean and covariance matrix, i.e., $\tilde{\boldsymbol{\mu}}_t = \mathbb{E}[\tilde{\mathbf{u}}_t]$ and $\tilde{\boldsymbol{\Gamma}}_t = \mathbb{E}[(\tilde{\mathbf{u}}_t - \tilde{\boldsymbol{\mu}}_t)(\tilde{\mathbf{u}}_t - \tilde{\boldsymbol{\mu}}_t)^T]$ can be computed in the following way. Using (13), the mean can be computed as

$$\tilde{\boldsymbol{\mu}}_t = \begin{bmatrix} \mathbf{H}_t\mathbb{E}[\mathbf{u}_{t-1}] \\ \mathbf{H}_{t+1}\mathbf{H}_t\mathbb{E}[\mathbf{u}_{t-1}] \\ \vdots \\ \mathbf{H}_{t+N_s-1}\mathbf{H}_{t+N_s-2}\ldots\mathbf{H}_t\mathbb{E}[\mathbf{u}_{t-1}] \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu}_t \\ \boldsymbol{\mu}_{t+1} \\ \vdots \\ \boldsymbol{\mu}_{t+N_s-1} \end{bmatrix}. \tag{15}$$

The time-dependent covariance matrix for any N_s snapshots, i.e., $\tilde{\boldsymbol{\Gamma}}_t$ is given by

$$\tilde{\boldsymbol{\Gamma}}_t = \tilde{\mathbf{R}}_t - \tilde{\boldsymbol{\mu}}_t\tilde{\boldsymbol{\mu}}_t^T, \tag{16}$$

where the correlation matrix $\tilde{\mathbf{R}}_t = \mathbb{E}[\tilde{\mathbf{u}}_t\tilde{\mathbf{u}}_t^T]$ is given by

$$\tilde{\mathbf{R}}_t = \begin{bmatrix} \mathbb{E}[\mathbf{u}_t\mathbf{u}_t^T] & \ldots & \mathbb{E}[\mathbf{u}_t\mathbf{u}_{t+N_s-1}^T] \\ \vdots & \ddots & \vdots \\ \mathbb{E}[\mathbf{u}_{t+N_s-1}\mathbf{u}_t^T] & \ldots & \mathbb{E}[\mathbf{u}_{t+N_s-1}\mathbf{u}_{t+N_s-1}^T] \end{bmatrix}. \tag{17}$$

The diagonal blocks of $\tilde{\mathbf{R}}_t$ are given as

$$\mathbf{R}_{t+\tau} = \mathbb{E}[\mathbf{u}_{t+\tau}\mathbf{u}_{t+\tau}^T] = \mathbf{H}_{t+\tau}\mathbf{R}_{t+\tau-1}\mathbf{H}_{t+\tau}^T + \mathbf{Q}_{t+\tau}, \tag{18}$$

where $\tau = 0, \ldots, N_s - 1$. The general form of the right and left off-diagonal blocks of $\tilde{\mathbf{R}}_t$ can be given for any two temporal lags τ_m and τ_n, where $\tau_m = 0, \ldots, N_s - 1$, $\tau_n = 0, \ldots, N_s - 1$ and $\tau_m \neq \tau_n$. The right off-diagonal blocks $(\tau_m < \tau_n)$ are given as

$$\mathbb{E}[\mathbf{u}_{t+\tau_m}\mathbf{u}_{t+\tau_n}^T] = \mathbf{R}_{t+\tau_m}\mathbf{H}_{t+\tau_m+1}^T\ldots\mathbf{H}_{t+\tau_n-1}^T\mathbf{H}_{t+\tau_n}^T, \tag{19}$$

and the left off-diagonal blocks $(\tau_m > \tau_n)$ are given as

$$\mathbb{E}[\mathbf{u}_{t+\tau_m}\mathbf{u}_{t+\tau_n}^T] = \mathbf{H}_{t+\tau_m}\mathbf{H}_{t+\tau_m-1}\ldots\mathbf{H}_{t+\tau_n+1}\mathbf{R}_{t+\tau_n}, \tag{20}$$

where $\mathbf{R}_{t+\tau_m}$ and $\mathbf{R}_{t+\tau_n}$ can be computed by the recursive relationships of (18). Substituting (17) into (16) and using the expression of (15) only the diagonal blocks of $\tilde{\boldsymbol{\Gamma}}_t$ can be recursively represented as

$$\boldsymbol{\Gamma}_{t+\tau} = \mathbb{E}[(\mathbf{u}_{t+\tau} - \boldsymbol{\mu}_{t+\tau})(\mathbf{u}_{t+\tau} - \boldsymbol{\mu}_{t+\tau})^T] \tag{21}$$

$$\boldsymbol{\Gamma}_{t+\tau} = \mathbf{H}_{t+\tau}\left(\mathbf{R}_{t+\tau-1} - \boldsymbol{\mu}_{t+\tau-1}\boldsymbol{\mu}_{t+\tau-1}^T\right)\mathbf{H}_{t+\tau}^T + \mathbf{Q}_{t+\tau} \tag{22}$$

$$\boldsymbol{\Gamma}_{t+\tau} = \mathbf{H}_{t+\tau}\boldsymbol{\Gamma}_{t+\tau-1}\mathbf{H}_{t+\tau}^T + \mathbf{Q}_{t+\tau}, \tag{23}$$

where $\tau = 0, \ldots, N_s - 1$. Note that, at any time t, the N_s *snapshots*

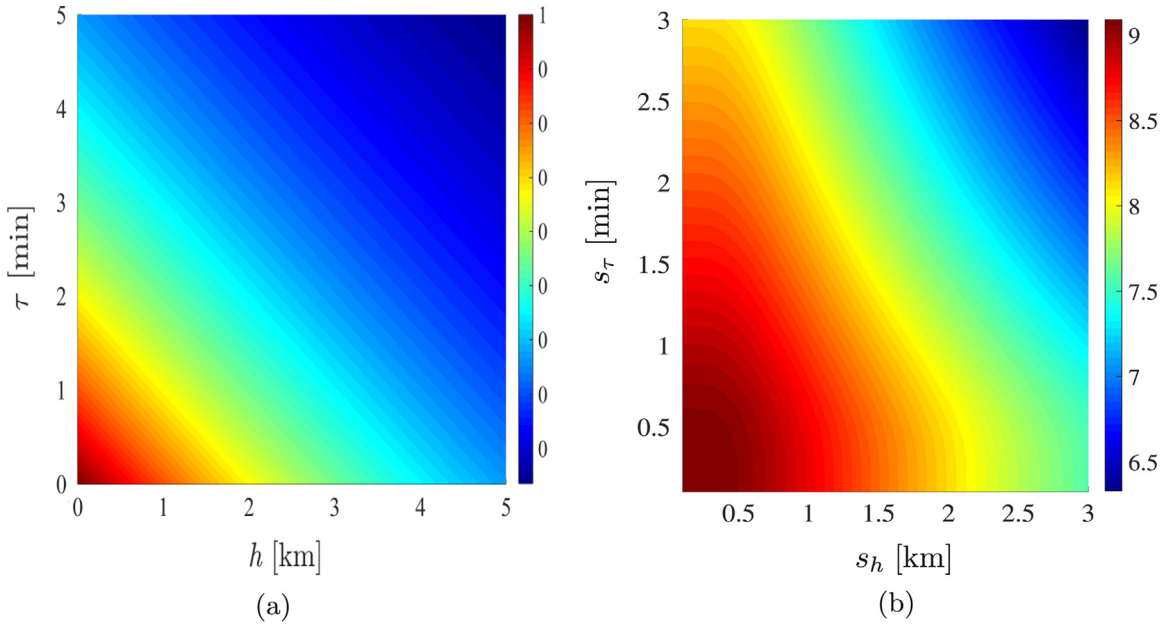

Fig. 2. Effect of the parameters of the covariance function on the MSE (a) plot of the space–time variation of the covariance function ($s_h = 5$; $s_\tau = 5$, $\sigma_u^2 = 1$) and (b) MSE variation with s_h and s_τ ($N = 25$, $N_s = 4$, $\tilde{\mathbf{C}}_t = \text{blkdiag}(\mathbf{I}_{25}, \mathbf{I}_{25}, \mathbf{I}_{25}, \mathbf{I}_{25})$, $\sigma_u^2 = 1$, $\sigma_e^2 = 0.1$).

ahead first and second order statistics of the field can be pre-computed if $\mathbb{E}[\mathbf{u}_{t-1}]$, $\mathbf{\Gamma}_{t-1}$, $\mathbf{H}_{t+\tau}$, and $\mathbf{Q}_{t+\tau}$ for $\tau = 0, \ldots, N_s - 1$ are all known a priori. One way to estimate the first and second order statistics of $\mathbf{u}_{t-1}$ is to use a "sequential minimum mean square error (MMSE) estimator", i.e., a standard Kalman filter [22] that uses the previous measurements up to time $t-1$. Let us assume that this estimate is given by $\hat{\mathbf{u}}_{t-1}$, with an estimation error covariance matrix $\mathbf{\Sigma}_{t-1}$, i.e., $\mathbf{\Sigma}_{t-1} = \mathbb{E}[(\mathbf{u}_{t-1} - \hat{\mathbf{u}}_{t-1})(\mathbf{u}_{t-1} - \hat{\mathbf{u}}_{t-1})^T]$. We use these as the first and second order statistics of the state $\mathbf{u}_{t-1}$, i.e., $\mathbf{u}_{t-1} \sim \mathcal{N}(\hat{\mathbf{u}}_{t-1}, \mathbf{\Sigma}_{t-1})$.

3. Performance metrics

In this section, we derive the performance metric, i.e., $g(\tilde{\mathbf{w}}_t)$ as mentioned in (6) for both stationary and non-stationary field estimation problems. In the space–time measurement model of (3), the unknown parameter $\tilde{\mathbf{u}}_t$ is statistically characterized by $\tilde{\mathbf{u}}_t \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_t, \tilde{\mathbf{\Gamma}}_t)$. The mean and the covariance matrix can be computed for both the stationary and the non-stationary fields as mentioned in the previous section.

The unknown parameter $\tilde{\mathbf{u}}_t$ can be estimated using an LMMSE estimator [22], i.e., $\hat{\tilde{\mathbf{u}}}_t = \tilde{\boldsymbol{\mu}}_t + \sigma_e^{-2}(\tilde{\mathbf{\Gamma}}_t^{-1} + \sigma_e^{-2}\tilde{\mathbf{C}}_t^T\tilde{\mathbf{C}}_t)\tilde{\mathbf{C}}_t^T(\tilde{\mathbf{y}}_t - \tilde{\mathbf{C}}_t\tilde{\boldsymbol{\mu}}_t)$. The MSE matrix, i.e., $\mathbb{E}[(\tilde{\mathbf{u}}_t - \hat{\tilde{\mathbf{u}}}_t)(\tilde{\mathbf{u}}_t - \hat{\tilde{\mathbf{u}}}_t)^T]$ is then given by

$$\mathbf{\Sigma}_t(\mathbf{w}_t, \ldots, \mathbf{w}_{t+N_s-1}) = (\tilde{\mathbf{\Gamma}}_t^{-1} + \sigma_e^{-2}\tilde{\mathbf{C}}_t^T\tilde{\mathbf{C}}_t)^{-1}. \tag{24}$$

We mention that (24) is considered as the generalized expression (for both stationary and non-stationary field estimation) of the MSE matrix in this work. The performance metric to estimate the field at N locations over N_s snapshots is quantified as $\text{tr}[\mathbf{\Sigma}_t(\mathbf{w}_t, \ldots, \mathbf{w}_{t+N_s-1})]$ or $\text{tr}[\mathbf{\Sigma}_t(\tilde{\mathbf{w}}_t)]$. By using the relation of (5), the performance metric in (24) can be written as

$$\mathbf{\Sigma}_t(\mathbf{w}_t, \ldots, \mathbf{w}_{t+N_s-1}) = (\tilde{\mathbf{\Gamma}}_t^{-1} + \sigma_e^{-2}\text{diag}(\tilde{\mathbf{w}}_t))^{-1}. \tag{25}$$

Here, we assume that $\tilde{\mathbf{\Gamma}}_t$ is well-conditioned and accurately invertible. We will come back to this issue later on in this section.

Remark (*Recursive performance metric*): When the field is non-stationary as mentioned in the previous section, we can use $\mathbf{u}_{t-1} \sim \mathcal{N}(\hat{\mathbf{u}}_{t-1}, \mathbf{\Sigma}_{t-1})$ to compute the N_s snapshots ahead first and second order statistics, i.e, $\tilde{\boldsymbol{\mu}}_t$ and $\tilde{\mathbf{\Gamma}}_t$. The expression of the MSE matrix of (25) can be evaluated by substituting $\mathbf{u}_{t-1} \sim \mathcal{N}(\hat{\mathbf{u}}_{t-1}, \mathbf{\Sigma}_{t-1})$ in the recursive relationship of (23). After the aforementioned substitutions, for $N_s = 1$, the performance metric of (25), i.e., $\mathbf{\Sigma}_t(\tilde{\mathbf{w}}_t)$, becomes $\mathbf{\Sigma}_t(\mathbf{w}_t)$. This is given as

$$\begin{aligned}\mathbf{\Sigma}_t\Big(\mathbf{w}_t\Big) &= [(\mathbf{H}_t\mathbf{\Sigma}_{t-1}\Big(\mathbf{w}_{t-1}\Big)\mathbf{H}_t^T + \mathbf{Q}_t)^{-1} + \sigma_e^{-2}\mathbf{C}_t^T\mathbf{C}_t]^{-1} \\ &= [(\mathbf{H}_t\mathbf{\Sigma}_{t-1}\Big(\mathbf{w}_{t-1}\Big)\mathbf{H}_t^T + \mathbf{Q}_t)^{-1} + \sigma_e^{-2}\text{diag}\Big(\mathbf{w}_t\Big)]^{-1}.\end{aligned} \tag{26}$$

This expression is the same as the single snapshot ahead update of the state error covariance matrix of a simple Kalman filter as a function of the selection vectors at time index t and $t-1$.

We can see that, for a large service area (large N) and/or high N_s, the computations of $\tilde{\boldsymbol{\mu}}_t$ and $\tilde{\mathbf{\Gamma}}_t$ as derived in (15) and (16), respectively, can be cumbersome from the *real time monitoring* perspective. However, in the simulation section we solve both the single snapshot and the multiple snapshots ahead sensor placement problems.

3.1. Effect of spatio-temporal correlation

The parameters s_h and s_τ control the strength of the spatial and temporal correlations, respectively. Increasing these values, the field becomes more correlated over space and time. Also for a fixed noise power, the MSE with all the candidate locations equipped with sensors, i.e., $\text{tr}[\mathbf{\Sigma}_t(\mathbf{1}_N, \ldots, \mathbf{1}_N)]$, reduces as s_h and s_τ jointly increase as shown in Fig. 2b. From the aforementioned analysis, it can be said that to achieve a desired estimation performance, less sensors are required to estimate a highly space–time correlated field than to estimate a lightly correlated field.

3.2. Highly correlated fields

For highly space–time correlated fields the spatio-temporal covariance matrix can be ill-conditioned [23], meaning that $\tilde{\mathbf{\Gamma}}_t$ in (25) is close to singular. In that case, we propose alternative formulations of the MSE matrix $\mathbf{\Sigma}_t$. In this paper, we leverage the matrix inversion lemma (MIL) and a special regularization

parameter to remove the ill-conditioning, as follows. By using the MIL, the error expression in (24) can be equivalently expressed as

$$\begin{aligned}\boldsymbol{\Sigma}_t(\mathbf{w}_t, \ldots, \mathbf{w}_{t+N_s-1}) &= (\tilde{\boldsymbol{\Gamma}}_t^{-1} + \sigma_e^{-2}\tilde{\mathbf{C}}_t^T\tilde{\mathbf{C}}_t)^{-1} \\ &= \tilde{\boldsymbol{\Gamma}}_t - \tilde{\boldsymbol{\Gamma}}_t\tilde{\mathbf{C}}_t^T(\tilde{\mathbf{C}}_t\tilde{\boldsymbol{\Gamma}}_t\tilde{\mathbf{C}}_t^T + \sigma_e^2\mathbf{I}_{\tilde{M}_t})^{-1}\tilde{\mathbf{C}}_t\tilde{\boldsymbol{\Gamma}}_t. \end{aligned} \tag{27}$$

Assuming a nonzero scalar constant $\beta \in \mathbb{R}$, the ill-conditioned matrix $\tilde{\boldsymbol{\Gamma}}_t$ can be regularized to a well-conditioned matrix $\mathbf{S}$ as

$$\mathbf{S} = \tilde{\boldsymbol{\Gamma}}_t + \beta\mathbf{I}_{NN_s}. \tag{28}$$

We now substitute $\tilde{\boldsymbol{\Gamma}}_t = \mathbf{S} - \beta\mathbf{I}_{NN_s}$ in the middle inverse of the right-most term of (27) and using the fact that $\tilde{\mathbf{C}}_t\tilde{\mathbf{C}}_t^T = \mathbf{I}_{\tilde{M}_t}$, we obtain

$$\boldsymbol{\Sigma}_t(\mathbf{w}_t, \ldots, \mathbf{w}_{t+N_s-1}) = \tilde{\boldsymbol{\Gamma}}_t - \tilde{\boldsymbol{\Gamma}}_t\tilde{\mathbf{C}}_t^T(\tilde{\mathbf{C}}_t\mathbf{S}\tilde{\mathbf{C}}_t^T + (\sigma_e^2 - \beta)\mathbf{I}_{\tilde{M}_t})^{-1}\tilde{\mathbf{C}}_t\tilde{\boldsymbol{\Gamma}}_t. \tag{29}$$

Using the MIL we can write

$$(\mathbf{S}^{-1} + (\sigma_e^2 - \beta)^{-1}\tilde{\mathbf{C}}_t^T\tilde{\mathbf{C}}_t)^{-1} = \mathbf{S} - \mathbf{S}\tilde{\mathbf{C}}_t^T(\tilde{\mathbf{C}}_t\mathbf{S}\tilde{\mathbf{C}}_t^T + (\sigma_e^2 - \beta)\mathbf{I}_{\tilde{M}_t})^{-1}\tilde{\mathbf{C}}_t\mathbf{S}. \tag{30}$$

Using (30), we have the following matrix identity

$$\begin{aligned}&\tilde{\mathbf{C}}_t^T\left(\tilde{\mathbf{C}}_t\mathbf{S}\tilde{\mathbf{C}}_t^T + (\sigma_e^2 - \beta)\mathbf{I}_{\tilde{M}_t}\right)^{-1}\tilde{\mathbf{C}}_t \\ &= \mathbf{S}^{-1}\left[\mathbf{S} - (\mathbf{S}^{-1} + (\sigma_e^2 - \beta)^{-1}\tilde{\mathbf{C}}_t^T\tilde{\mathbf{C}}_t)^{-1}\right]\mathbf{S}^{-1}. \end{aligned} \tag{31}$$

Substituting (31) into (29), the error expression of $\boldsymbol{\Sigma}_t(\mathbf{w}_t, \ldots, \mathbf{w}_{t+N_s-1})$ can be viewed as a function of the space–time sensor location selection vectors given as

$$\begin{aligned}&\boldsymbol{\Sigma}_t\left(\mathbf{w}_t, \ldots, \mathbf{w}_{t+N_s-1}\right) \\ &= \tilde{\boldsymbol{\Gamma}}_t - \tilde{\boldsymbol{\Gamma}}_t\mathbf{S}^{-1}\left[\mathbf{S} - (\mathbf{S}^{-1} + (\sigma_e^2 - \beta)^{-1}\tilde{\mathbf{C}}_t^T\tilde{\mathbf{C}}_t)^{-1}\right]\mathbf{S}^{-1}\tilde{\boldsymbol{\Gamma}}_t \\ &= \tilde{\boldsymbol{\Gamma}}_t - \tilde{\boldsymbol{\Gamma}}_t\mathbf{S}^{-1}\tilde{\boldsymbol{\Gamma}}_t + \tilde{\boldsymbol{\Gamma}}_t\mathbf{S}^{-1}\left[\mathbf{S}^{-1} + (\sigma_e^2 - \beta)^{-1}\tilde{\mathbf{C}}_t^T\tilde{\mathbf{C}}_t\right]^{-1}\mathbf{S}^{-1}\tilde{\boldsymbol{\Gamma}}_t \\ &= \mathbf{A} + \mathbf{B}^T\left[\mathbf{S}^{-1} + (\sigma_e^2 - \beta)^{-1}\mathrm{diag}\left(\tilde{\mathbf{w}}_t\right)\right]^{-1}\mathbf{B}, \end{aligned} \tag{32}$$

where the matrices $\mathbf{A} = \tilde{\boldsymbol{\Gamma}}_t - \tilde{\boldsymbol{\Gamma}}_t\mathbf{S}^{-1}\tilde{\boldsymbol{\Gamma}}_t$ and $\mathbf{B} = \mathbf{S}^{-1}\tilde{\boldsymbol{\Gamma}}_t$ are independent of the selection vectors, and therefore known a priori.

Here we comment that the regularization in (28) is valid for any nonzero $\beta \in \mathbb{R}$ if $\mathbf{S}$ is only desired to be invertible. However, to maintain the positive definiteness of $\mathbf{S}$ as well as $[\mathbf{S}^{-1} + (\sigma_e^2 - \beta)^{-1}\mathrm{diag}(\tilde{\mathbf{w}}_t)]$, we specifically choose $0 < \beta < \sigma_e^2$.

We notice that the MSE matrix $\boldsymbol{\Sigma}_t(\mathbf{w}_t, \ldots, \mathbf{w}_{t+N_s-1})$ computed as in (32) does not involve any inversion of the possible ill-conditioned $\tilde{\boldsymbol{\Gamma}}_t$. The only inversions are of the regularized, and thus well-conditioned matrix $\mathbf{S}$. The expression (32) therefore offers a better alternative than (27) for highly correlated fields.

3.3. Uncorrelated fields

In this work, we mainly target the application of spatio-temporal monitoring of environmental fields like pollutant concentrations in the atmosphere, concentrations of some hazardous gas, rainfall, ground layer ozone, humidity, etc. Generally, these fields are spatio-temporally correlated. But in some scenarios the spatial/temporal correlation may be very small. In these cases, the off-diagonal elements of $\tilde{\boldsymbol{\Gamma}}_t$ are close to 0. In such cases, $\tilde{\boldsymbol{\Gamma}}_t$ can be modeled as $\tilde{\boldsymbol{\Gamma}}_t = \sigma_u^2\mathbf{I}_{NN_s}$. Then the MSE matrix is given by

$$\begin{aligned}\boldsymbol{\Sigma}_t\left(\mathbf{w}_t, \ldots, \mathbf{w}_{t+N_s-1}\right) &= \left(\sigma_u^{-2}\mathbf{I}_{NN_s} + \sigma_e^{-2}\tilde{\mathbf{C}}_t^T\tilde{\mathbf{C}}_t\right)^{-1} \\ &= \left(\sigma_u^{-2}\mathbf{I}_{NN_s} + \sigma_e^{-2}\mathrm{diag}\left(\tilde{\mathbf{w}}_t\right)\right)^{-1}. \end{aligned} \tag{33}$$

Note that, if the field is uncorrelated the estimation error is mainly characterized by the signal to noise ratio (SNR) of the system, i.e., σ_u^2/σ_e^2. In this case, the term $\sigma_u^{-2}\mathbf{I}_{NN_s}$ acts both as a regularization term ensuring the computability of (33) and as a scaling term for the MSE. For the current measurement model of (3), to estimate an uncorrelated field (with the same $\sigma_u{}^2$ over space and time) the number of sensors is more relevant for estimation performance than their constellation as long as the MSE is considered to be the performance criterion.

4. Sensor placement problem

A generalized performance metric to estimate both stationary and non-stationary fields can be formulated as $\mathrm{tr}[\boldsymbol{\Sigma}_t(\tilde{\mathbf{w}}_t)]$, where $\boldsymbol{\Sigma}_t(\tilde{\mathbf{w}}_t)$ is the generalized MSE matrix (24). Following the main optimization problem of (6), an offline selection of sensing locations from N locations over N_s snapshots can be performed by solving the following optimization problem:

$$\hat{\tilde{\mathbf{w}}}_t = \underset{\tilde{\mathbf{w}}_t \in [0,1]^{NN_s}}{\arg\min} \left\{ \| \tilde{\mathbf{w}}_t \|_1, \ \text{s.t. } \mathrm{tr}\left[\boldsymbol{\Sigma}_t(\tilde{\mathbf{w}}_t)\right] \le \gamma' \right\}, \tag{34}$$

where γ' is a threshold on the estimation performance. An extra set of affine constraints can be added to the problem of (34), to restrict the minimum number of sensing locations to be selected at every time index t. This is given as

$$\| \mathbf{w}_{t+\tau} \|_1 \ge p, \quad \tau = 0, \ldots, N_s - 1. \tag{35}$$

This constraint enforces at least p sensors to be selected at every snapshot. This is an optional design constraint to efficiently utilize every available snapshot.[1] Considering the general form of the performance metric, i.e., (32), the optimization problem of (34) can be formulated as a semidefinite programming (SDP). For the N candidate sensing locations, the performance constraint can be expressed as N linear matrix inequalities (LMI) [7,6]. If the column vectors of the matrix $\mathbf{B}$ are given by $\mathbf{b}_j$, where $j = 1, \ldots, N$, then the SDP is given by

$$\begin{aligned}\hat{\tilde{\mathbf{w}}}_t = &\underset{\tilde{\mathbf{w}}_t \in [0,1]^N, \mathbf{v} \in \mathbb{R}^N}{\arg\min} \ \| \tilde{\mathbf{w}}_t \|_1, \\ &\text{s. t.} \ \begin{bmatrix} v_j & \mathbf{b}_j^T \\ \mathbf{b}_j & \mathbf{S}^{-1} + (\sigma_e^2 - \beta)^{-1}\mathrm{diag}(\tilde{\mathbf{w}}_t) \end{bmatrix} \succcurlyeq 0, \quad j = 1, \ldots, N \end{aligned} \tag{36}$$

$$\mathbf{1}^T\mathbf{v} \le \gamma' - \mathrm{tr}(\mathbf{A}), \tag{37}$$

where we use the auxiliary variable $\mathbf{v} = [v_1, \ldots, v_N]^T$. The set of N LMIs in (36) signify the fact that $v_j \ge \mathbf{b}_j^T[\mathbf{S}^{-1} + (\sigma_e^2 - \beta)^{-1}\mathrm{diag}(\mathbf{w}_t)]^{-1}\mathbf{b}_j$, where $j = 1, \ldots, N$ (using the Schur complement of the block $\mathbf{S}^{-1} + (\sigma_e^2 - \beta)^{-1}\mathrm{diag}(\mathbf{w}_t)$).

The solution of the aforementioned optimization problem gives the sensor placement patterns achieving the desired estimation performance γ'. It is clear that lowering γ', i.e., putting a tighter threshold on the performance, more sensing locations need to be selected.

In practical scenarios, the performance threshold can generally be derived from the application, i.e., the nature of the field to be estimated, required resolution, etc. In the present work, we calculate the thresholds by scaling the best case, i.e., sensors are deployed in all N candidate locations. In other words, we consider $\gamma' = \zeta\ \mathrm{tr}[\boldsymbol{\Sigma}_t(\mathbf{1}_{NN_s})]$, where $\zeta > 1$ is a positive scaling parameter.

[1] Spatial selection of sensing locations can be performed in a more efficient manner by employing a structural constraint on $\mathbf{w}_{t+\tau}$ like group sparsity. The evolution of $\mathbf{w}_t$ can also be controlled by using a smoothing penalty in the cost function of (34), where the sensing locations are selected on a single snapshot basis [24].

5. Sensor placement algorithm using iterative saddle-point method

From the above discussions, the structure of the general optimization problem, i.e., (34) with the performance metric (32), can be formulated as,

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in[0,1]^L}\left\{ \| \mathbf{w} \|_1 \ \text{s.t. tr}\left[\mathbf{B}^T(\mathbf{S}^{-1} + (\sigma_e^2 - \beta)^{-1}\text{diag}\big(\mathbf{w}\big))^{-1}\mathbf{B}\right] - \gamma'' \le 0\right\} \tag{38}$$

where $\gamma'' = \gamma - \text{tr}(\mathbf{A})$, and the matrices $\mathbf{A}$ and $\mathbf{B}$ are independent of $\mathbf{w}$. In this section, we use $\mathbf{w}$ instead of $\tilde{\mathbf{w}}_t$ for the sake of notational simplicity. Note that, the aforementioned problem can be used to select sensing locations for both the stationary and the non-stationary scenarios. The length of the selection vector is given as $L = NN_s$. In this work, we consider the fact that the spatio-temporal covariance matrix is accurately invertible, i.e, we take $\beta = 0$, $\mathbf{B} = \mathbf{I}_L$, and $\mathbf{A} = \mathbf{0}_{L\times L}$. Using these the optimization problem of (38) can be given as

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in[0,1]^L}\left\{ \| \mathbf{w} \|_1 \ \text{s.t. tr}[\mathbf{Z}^{-1} + \sigma_e^{-2}\text{diag}\big(\mathbf{w}\big)]^{-1} - \gamma \le 0\right\}, \tag{39}$$

where the matrix $\mathbf{Z}$ is the spatio-temporal covariance matrix. We define a function $h(\mathbf{w})$ as

$$h(\mathbf{w}) = \text{tr}[\mathbf{Z}^{-1} + \sigma_e^{-2}\text{diag}(\mathbf{w})]^{-1} - \gamma. \tag{40}$$

However, as mentioned earlier, the convex problem (39) can be easily formulated as a semidefinite programming (SDP) problem and solved for $\mathbf{w}$ using off-the-shelf solvers like CVX [13] and SeDuMi [14]. But for a large service area and/or many snapshots the number of unknowns (L), i.e., the number of LMIs becomes increasingly high. In this case, SDP based approaches using standard solvers can be time consuming. In this section, we propose an alternative approach to solve the optimization problem (39) directly.

5.1. Primal–dual iterations

We use an iterative saddle-point method [25], to solve the optimization problem (39). We adopt first-order methods rather than Newton's method because Newton's method requires the expression of the Hessian and its inverse, which increases the computational complexity and leads to storage issues. First of all, we define the dual variable associated with the inequality constraint (39) as λ. Under convexity and Slater's condition (which holds for (39), given the choice of γ), we can prove that the dual variable λ lives in a bounded compact set $[0, \lambda_{\max}]$ [25, Lemma 3]. The value of $\lambda_{\max} > 0$ is easily computable a priori, given any Slater's vector. Let us now define the compact constraint sets, $\mathbb{X} \in [0, 1]^L$ and $\mathbb{D} \in [0, \lambda_{\max}]$ for the primal and the dual variables $\mathbf{w} \in \mathbb{X}$, and $\lambda \in \mathbb{D}$, respectively. The Lagrangian function $\mathcal{L}(\mathbf{w}, \lambda)$: $\mathbb{X} \times \mathbb{D} \to \mathbb{R}$, for the optimization problem (38) is given by,

$$\mathcal{L}(\mathbf{w}, \lambda) = \mathbf{1}^T\mathbf{w} + \lambda\, h(\mathbf{w}). \tag{41}$$

The primal–dual iterations for $i \ge 0$ can be given as,

$$\hat{\mathbf{w}}^{i+1} = \mathcal{P}_{\mathbb{X}}[\mathbf{w}^i - \alpha\nabla_{\mathbf{w}}\,\mathcal{L}(\mathbf{w}^i, \lambda^i)], \tag{42}$$

$$\hat{\lambda}^{i+1} = \mathcal{P}_{\mathbb{D}}[\lambda^i + \alpha\nabla_{\lambda}\,\mathcal{L}(\mathbf{w}^i, \lambda^i)], \tag{43}$$

where $\mathcal{P}_{\mathbb{X}}$ and $\mathcal{P}_{\mathbb{D}}$ are the projection operators onto the sets $\mathbb{X}$ and $\mathbb{D}$, respectively. The scalar $\alpha > 0$ is the step size. We define $\nabla_{\mathbf{w}}(\cdot)$ and $\nabla_{\lambda}(\cdot)$ as the gradients w.r.t. $\mathbf{w}$ and λ, respectively. Note that, the primal–dual iterations actually minimize $\mathcal{L}(\mathbf{w}, \lambda)$ w.r.t. $\mathbf{w}$ and maximize it w.r.t. λ in order to achieve the saddle point $(\hat{\mathbf{w}}, \hat{\lambda})$ [25], which satisfies

$$\mathcal{L}(\hat{\mathbf{w}}, \lambda) \le \mathcal{L}(\hat{\mathbf{w}}, \hat{\lambda}) \le \mathcal{L}(\mathbf{w}, \hat{\lambda}), \quad \forall\, \mathbf{w} \in \mathbb{X}, \lambda \in \mathbb{D}. \tag{44}$$

The expressions of the gradients $\nabla_{\mathbf{w}}(\cdot)$ and $\nabla_{\lambda}(\cdot)$ are computed in the Appendix. They are given as

$$\nabla_{\mathbf{w}}\,\mathcal{L}(\mathbf{w}^i, \lambda^i) = \mathbf{1}_L + \lambda^i\sigma_e^{-2}\text{diag}(-[\mathbf{Z}^{-1} + \sigma_e^{-2}\text{diag}(\mathbf{w}^i)]^{-2}) \tag{45}$$

$$\nabla_{\lambda}\,\mathcal{L}(\mathbf{w}^i, \lambda^i) = \text{tr}[\mathbf{Z}^{-1} + \sigma_e^{-2}\text{diag}(\mathbf{w}^i)]^{-1} - \gamma. \tag{46}$$

Due to compactness of the sets $\mathbb{X}$ and $\mathbb{D}$ and the invertibility of $\mathbf{Z}$, it can be proven that the gradients are bounded. And, in particular, we can write

$$\max\{\|\nabla_{\mathbf{w}}\,\mathcal{L}(\mathbf{w}, \lambda)\|, \|\nabla_{\lambda}\,\mathcal{L}(\mathbf{w}, \lambda)\|\} \le C, \quad \forall\, \mathbf{w} \in \mathbb{X}, \lambda \in \mathbb{D}, \tag{47}$$

where $C > 0$ is a constant. With this in place, due to Propositions 1 and 2 of [25], the iterates $\{\mathbf{w}^i, \lambda^i\}$ converge weakly (in the ergodic mean sense) to a neighborhood of the saddle point of the Lagrangian (41). The size of the neighborhood (i.e., the asymptotical error bound) is proportional to αC^2. In addition, convergence goes as $O(1/i\alpha)$, i being the iteration counter. A similar result is also valid for the amount of constraint violation. In practice, in the simulation results, we will select the step size α to trade-off convergence speed and asymptotical error. The stopping criteria will be based either on a maximum number of iterations or on a required tolerance on the value of $\| h(\mathbf{w}^i)\|$.

5.2. Iterative reweighted ℓ_1 algorithm to improve sparsity

The well-known convex approximation of the non-convex ℓ_0 norm is the sought-after ℓ_1 norm. However, there are better functions to model a sparsity-promoting cost like a sum of logarithms or a sum of inverse Gaussians. Unfortunately both of these functions are non-convex. For example, in the optimization problem of (6), the objective function can be replaced by a sparsity-promoting non-convex cost, i.e., $\sum_{l=1}^{L}\ln(\epsilon + [\mathbf{w}]_l)$. Here, $\epsilon > 0$ is used to maintain the stability of the sum of the logarithm cost. As mentioned in [18], such a log-concave function can be well approximated by its first order linear approximation. This means that minimizing $\sum_{l=1}^{L}\ln(\epsilon + [\mathbf{w}]_l)$ can be approximated by iteratively minimizing its linear approximation, i.e.,

$$\arg\min_{\mathbf{w}} \sum_{l=1}^{L}\frac{[\mathbf{w}]_l}{\epsilon + [\hat{\mathbf{w}}^j]_l}, \tag{48}$$

where $\hat{\mathbf{w}}^j$ is the estimate of $\mathbf{w}$ in the j-th iteration [18]. Following the derivation of [18], the optimization problem (6), can be formulated as the iterative reweighted ℓ_1 minimization given by

- **Initialize** j=0, weight vector $\mathbf{z}^0 = \mathbf{1}_L$, ϵ, and maximum number of iterations J
- **for** $j = 0, \ldots, J$

$$\hat{\mathbf{w}}^j = \arg\min_{\mathbf{w}\in[0,1]^L}\left\{ (\mathbf{z}^j)^T\mathbf{w}, \ \text{s.t. } g(\mathbf{w}) \le \gamma\right\}$$

- $[\mathbf{z}^{j+1}]_l = \frac{1}{\epsilon + [\hat{\mathbf{w}}^j]_l}$, for every $l = 1, \ldots, L$
- **end**;
- **set** $\hat{\mathbf{w}} = \hat{\mathbf{w}}^J$.

The aforementioned algorithm is envisaged to avoid the dependence of $\hat{\mathbf{w}}$ on the magnitude of its elements. Using this iterative approach, a higher weight is put on the smaller elements of $\mathbf{w}$ to push them towards 0, enhancing the sparsity in $\mathbf{w}$. On the other

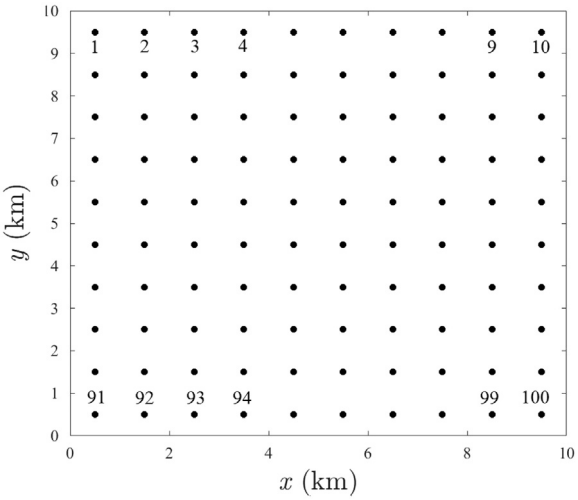

Fig. 3. Service area with the candidate sensing locations.

hand, it maintains the magnitude of the larger elements by putting a smaller weight. However, after this "sparsity-enhancing" iterative algorithm we still have $\hat{\mathbf{w}} \in [0, 1]^L$. After the computation of $\hat{\mathbf{w}}$ from the above iterative algorithm we compute $\mathbf{w} \in \{0, 1\}^L$ using randomized rounding or thresholding.

5.3. Primal–dual iterations with the iterative reweighted ℓ_1 minimization

The sparsity-enhancing iterative algorithm mentioned in the previous section can be implemented in combination with the saddle-point method. In this case, the Lagrangian can be formulated as

$$\mathcal{L}(\mathbf{w}^j, \lambda) = (\mathbf{z}^j)^T\mathbf{w}^j + \lambda\, h(\mathbf{w}^j), \tag{49}$$

where $\mathbf{z}^j$ is the weighting vector of the j-th iteration of the iterative algorithm. As before, we can compute the saddle-point iterates with this new Lagrangian, which (for the same reasons as mentioned in the Section 5.1) will converge weakly to a saddle point up to a bounded error.

In Algorithm 1, the saddle-point iterations for the reweighted ℓ_1 minimization are presented. The overall algorithm is implemented using two nested loops, where the inner loop (indexed by i) is used for the primal–dual iterations and the outer loop (indexed by j) is used for the iterative reweighted ℓ_1 algorithm.

To place the sensors dynamically every snapshot, the same Algorithm 1 is implemented for $t = 1, \ldots, T$ snapshots with $N_s = 1$, i.e., $L = N$. The estimation error is initialized as $\mathbf{\Sigma}_0$ at $t = 0$. After estimating $\mathbf{w}_t$ at any t, the estimation error, i.e., $\mathbf{\Sigma}_t(\mathbf{w}_t)$ is updated based on the recursive relation of (26). We refer to this algorithm as Dynamic Iterative Sparsity-Enhancing Sensor Placement (DISESP).

Algorithm 1. Saddle point iterations enhancing sparsity.

1: **Initialize**: $j = 0$, weight vector $\mathbf{z}^0 = \mathbf{1}_L$, α, I, J, tol, and ϵ.
2: **for** $j = 0, \ldots, J$
3: **solve** the saddle point iterations
4: **while** $i < I$ or $|h(\mathbf{w})| \geq$ tol

$$\hat{\mathbf{w}}^{i+1,j} = \mathcal{P}_{\mathbb{X}}[\mathbf{w}^{i,j} - \alpha\nabla_{\mathbf{w}}[(\mathbf{z}^j)^T\mathbf{w}^{i,j} + \lambda^{i,j}\, h(\mathbf{w}^{i,j})]],$$
$$\hat{\lambda}^{i+1,j} = \mathcal{P}_{\mathbb{D}}[\lambda^{i,j} + \alpha\nabla_{\lambda}[(\mathbf{z}^j)^T\mathbf{w}^{i,j} + \lambda^{i,j}\, h(\mathbf{w}^{i,j})]]$$

5: **end while**
6: **update** the weight vector by $[\mathbf{z}^{j+1}]_l = \frac{1}{\epsilon + [\hat{\mathbf{w}}^j]_l}$, where $l = 1, \ldots, L$.
7: **end for**
8: $\hat{\mathbf{w}} = \hat{\mathbf{w}}^J$.

6. Simulation results

In this section, we present simulation results for both stationary and non-stationary field estimation applications using the developed sensor placement method. Let us assume a service area of 10×10 square km which is discretized into $N = 100$ pixels of size 1 square km. The service area and the centroids of the pixels are shown in Fig. 3. We assume that all of these centroids are candidate sensing locations. They are row-wise indexed from top to bottom as shown in Fig. 3.

6.1. Sensor placement for stationary field estimation

In the first case, we assume that the field is spatio-temporally stationary. We consider to have $N_s = 3$ snapshots. In this case, the size of the spatio-temporal covariance matrix ($\tilde{\mathbf{\Gamma}}_t$) is 300×300. The temporal lags are $\tau = 0, 1, 2$ as $N_s = 3$. The diagonal and off-diagonal blocks of $\tilde{\mathbf{\Gamma}}_t$ are given by $\mathbf{\Gamma}_0 \in \mathbb{S}_{++}^{100}$ and $\mathbf{\Gamma}_2, \mathbf{\Gamma}_1, \mathbf{\Gamma}_{-1}, \mathbf{\Gamma}_{-2} \in \mathbb{S}_{++}^{100}$, respectively. The elements of these matrices are generated by the exponential covariance function mentioned in (7) (Section 2.3.1) with parameters $\sigma_u^2 = 1$, $s_h = 5$, $s_\tau = 2$. The d_{ij} parameters are computed from the distance matrix (matrix of all possible pairwise Euclidean distances) of the pixel centroids as shown in Fig. 3. Based on these, $\tilde{\mathbf{\Gamma}}_t$ is a symmetric, positive-definite and block-toeplitz matrix.

The measurement noise variance is assumed to be $\sigma_e^2 = 1$. The performance threshold γ' is computed by scaling the best case MSE (i.e. $\mathrm{tr}[\mathbf{\Sigma}_t(\mathbf{1}_{300})]$) by $\zeta = 2$. The parameters of the sparsity-enforcing iterative algorithm are $\epsilon = 10^{-8}$ and $J = 5$. A constant step-size of $\alpha = 0.1/(NN_s)$ is used in the saddle point algorithm and the algorithm is iterated until a desired tolerance level (tol) or the maximum number of iterations (I) is achieved. Here, we take tol $= 10^{-4}$ and $I = 300NN_s$.

The estimated sensor location selection vectors for $N_s = 3$ snapshots, i.e., $\hat{\mathbf{w}}_1$, $\hat{\mathbf{w}}_2$, $\hat{\mathbf{w}}_3$ before and after the randomized rounding are shown in Fig. 4a and b, respectively. In the next case, we keep the same γ' but assume that the field is spatio-temporally more correlated than the last time. In this case, we use $s_h = 7$ and $s_\tau = 3$. The resulting selected sensing locations, i.e., $\hat{\mathbf{w}}_1$, $\hat{\mathbf{w}}_2$, $\hat{\mathbf{w}}_3$ before and after the randomized rounding are shown in Fig. 5a and b, respectively.

It is observed that less sensors are needed to achieve a desired estimation performance, when the field is highly correlated over space and/or time. In Fig. 5a and b, we see that less sensing locations are selected than in Fig. 4a and b. This observation is consistent with the fact that the Bayesian MSE is reduced as the correlation over space/time is increased, as shown in Fig. 2b.

We study space–time sensor placement patterns for a simple exponential covariance function (uniformly decaying). It can be conjectured that for such a covariance function the optimal sensor placement is more or less uniform over space and time. However, different sensor placement patterns can be observed for different spatio-temporal covariance matrices, i.e., $\tilde{\mathbf{\Gamma}}_t$.

6.2. Sensor placement for non-stationary field estimation

In the second case, we consider that the field is non-stationary. We consider that the dynamics, i.e, $\mathbf{H}_t$ for $t = 1, \ldots, 7$ snapshots (minutes) are assumed to be known a priori. We present two

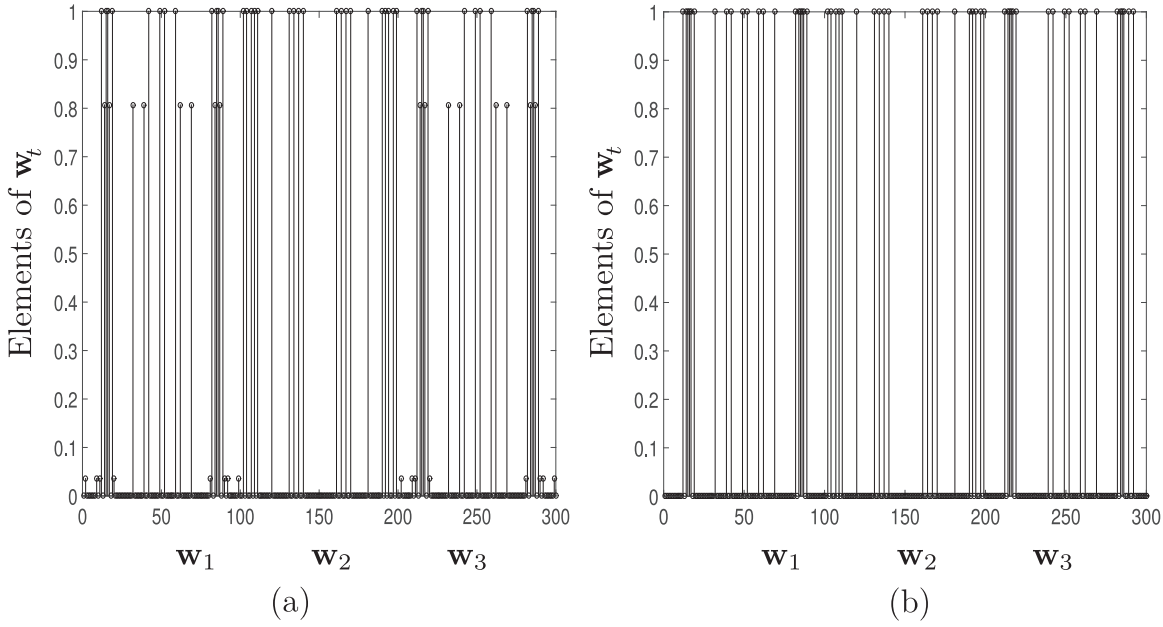

Fig. 4. Sensor placement pattern ($s_h = 5$; $s_\tau = 2$) (a) before randomization and (b) after randomization.

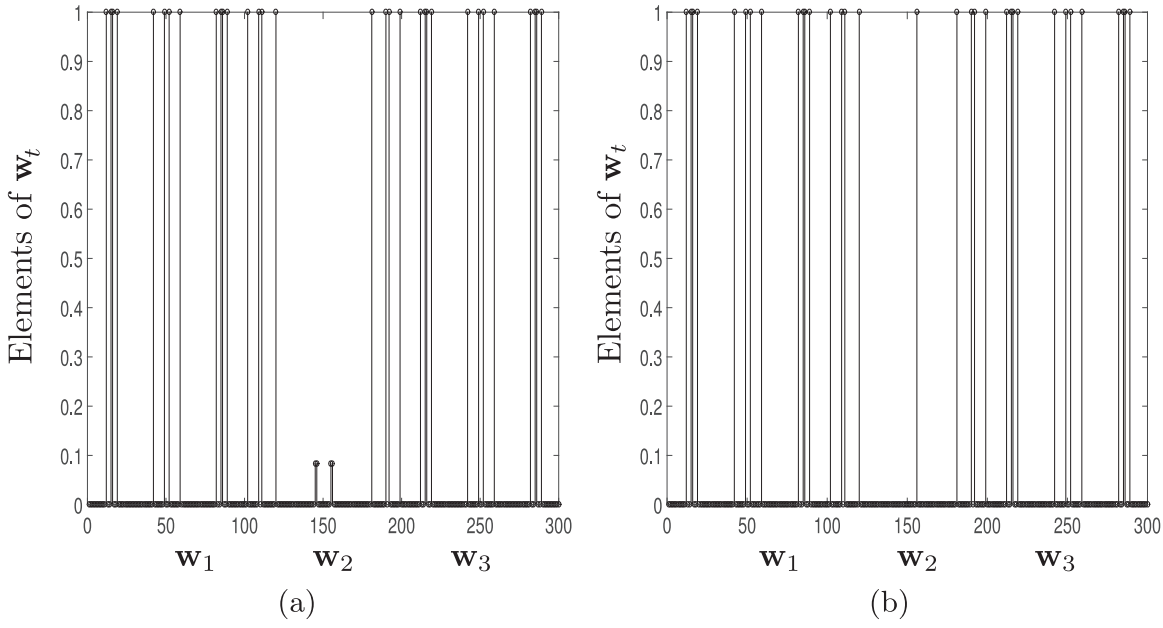

Fig. 5. Sensor placement pattern ($s_h = 7$; $s_\tau = 3$) (a) before randomization and (b) after randomization.

scenarios. First, we solve the multiple snapshots ahead sensor placement problem. We solve this only once without any updating although this could be considered as well. In the next case, we solve the single snapshot ahead sensor placement problem, where the performance metric is updated every snapshot. We consider the same service area as shown in Fig. 3.

6.2.1. Multiple snapshots ahead sensor placement

We consider $N_s = 3$ snapshots. The parameters of the state transition matrix are given by $\nu = 0.4$, $\mathbf{D}_t = \mathbf{D} = \mathbf{I}_2$, which is an isotropic diffusion, and $\mathbf{a}_t = \mathbf{a} = [0.5, 0.5]^T$ for $\mathbf{H}_1$, $\mathbf{H}_2$, $\mathbf{H}_3$. We assume that the initial distribution of the field is given as $\mathbf{u}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Gamma}_0)$, where we take $\boldsymbol{\mu}_0 = \mathbf{1}_{100}$ and the elements of $\boldsymbol{\Gamma}_0 \in \mathbb{S}_{++}^{100}$ are given by (7) with parameters $\sigma_u^2 = 1$, $s_h = 1$, $s_\tau = 0$, i.e, $[\boldsymbol{\Gamma}_0]_{ij} = \exp[-d_{ij}]$ for $i, j = 1, \ldots, 100$. The process noise is characterized by $\mathbf{q}_t \sim \mathcal{N}(\mathbf{0}_{100}, 0.001\mathbf{I}_{100})$ for all t. The diagonal, right and left off-diagonal blocks of $\tilde{\mathbf{R}}_t$ are computed using (18), (19), and (20), respectively. Finally, the overall space–time covariance matrix is computed by $\tilde{\boldsymbol{\Gamma}}_t = \tilde{\mathbf{R}}_t - \tilde{\boldsymbol{\mu}}_t \tilde{\boldsymbol{\mu}}_t^T$, where $\tilde{\boldsymbol{\mu}}_t$ and $\tilde{\mathbf{R}}_t$ are computed using the expressions of (15) and (17), respectively. The measurement noise variance is assumed to be same as before, i.e., $\sigma_e^2 = 1$. The parameters of the iterative saddle point algorithm are also maintained to be the same as before. In this case, we again adopt the performance threshold γ' by scaling the best MSE by $\zeta = 2$. The sensor location selection vectors, i.e., $\hat{\mathbf{w}}_1$, $\hat{\mathbf{w}}_2$, $\hat{\mathbf{w}}_3$ (before and after randomization) are shown in Fig. 6a and b, respectively.

It is observed that when the field is non-stationary, the selected sensing locations are less uniformly distributed than for the stationary field case. It is also seen that when a non-stationary field is to be estimated jointly using the measurements from multiple snapshots then measurements from alternate snapshots are more informative than measurements from consecutive snapshots. This

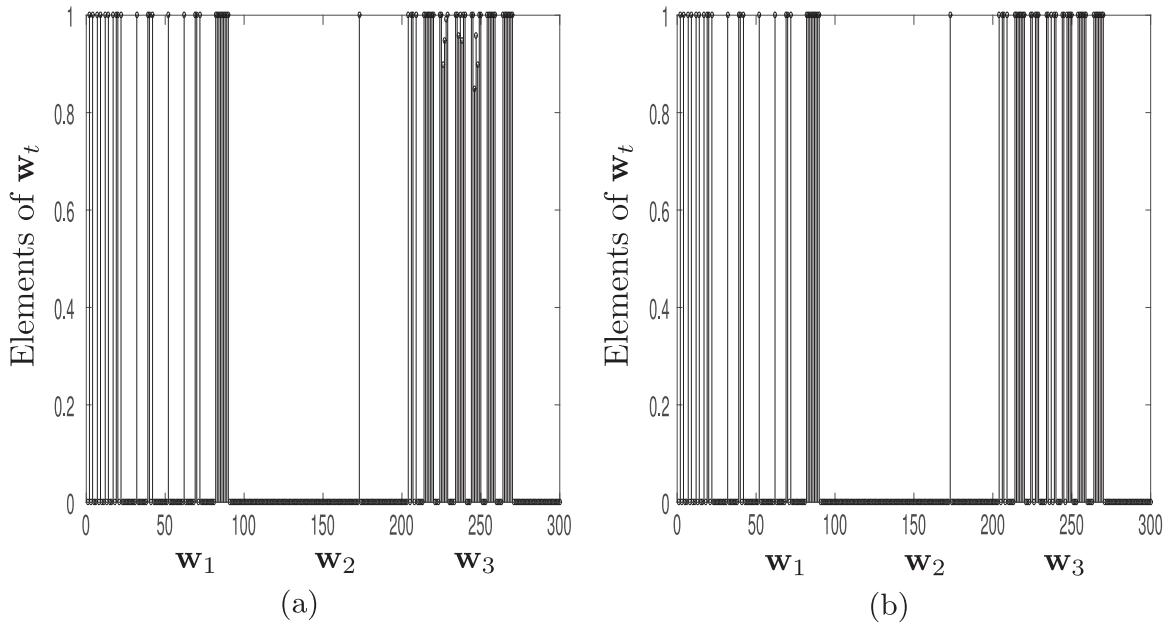

Fig. 6. Sensor placement pattern (non-stationary field) (a) before randomization and (b) after randomization.

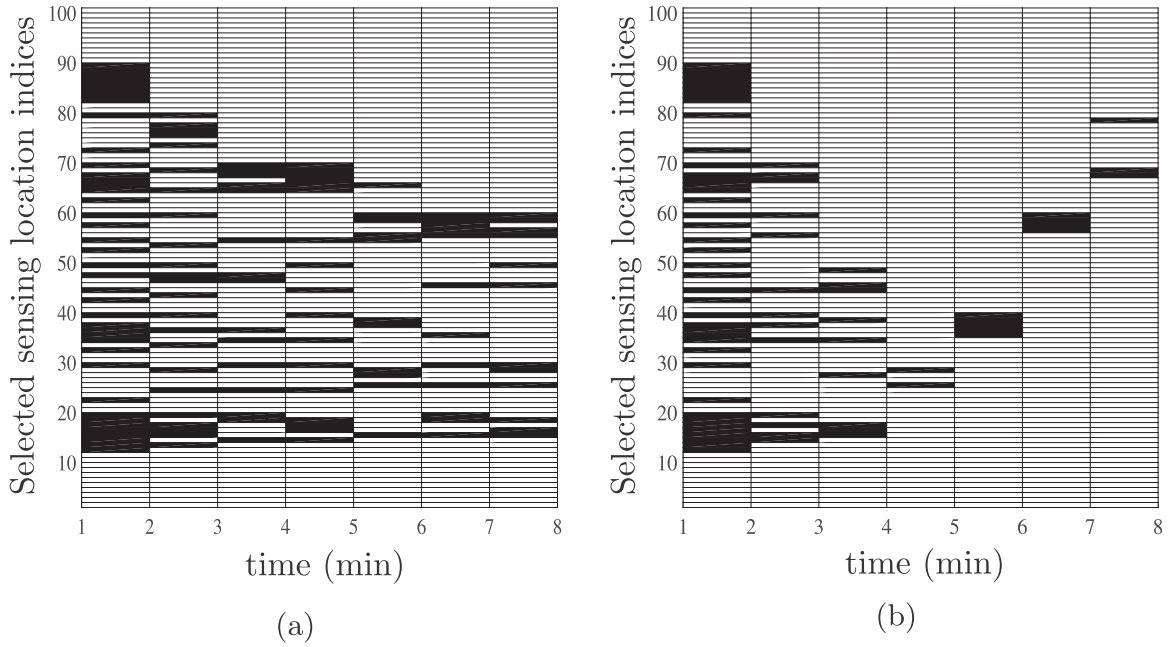

Fig. 7. (a) Sensor deployment pattern for $\mathbf{H}_t = \mathbf{H}$ and (b) sensor deployment pattern for a time-varying advection parameter $\mathbf{a}_t$.

makes sense as sensors in alternate snapshots are less correlated and the values for the middle snapshot can be easily predicted. The dependence on the dynamics is even more clearly observed in the next case, where we update the performance metric every snapshot.

6.2.2. Single snapshot ahead sensor placement

In this case, we select the sensing location for every snapshot, i.e., we consider $N_s = 1$. To avoid the computation of $\check{\mathbf{\Gamma}}_t$ we update the performance metric every snapshot based on (26), i.e., we use the developed DISESP approach mentioned in Section 5.3.

Considering the same service area shown in Fig. 3, we would like to choose the sensing locations every snapshot dynamically. Note that, for every t we compute $\mathbf{w}_t$ (with the prior knowledge of $\mathbf{H}_t$, $\mathbf{Q}_t$, and the estimation error covariance of the previous snapshot, i.e., $\mathbf{\Sigma}_{t-1}$) whose support gives the locations where to move/place the sensors to estimate the field for the current snapshot. The scaling and diffusion parameters of the state transition matrix are given by $\nu = 0.4$ and $\mathbf{D}_t = \mathbf{D} = \mathbf{I}_2$, which is same as before. For the advection, we consider two scenarios. In the first case, we assume that $\mathbf{a}_t$ is fixed for all t, i.e., we have $\mathbf{H}_t = \mathbf{H}$. In the next case, we change $\mathbf{a}_t$ every minute. In that case, the values of $\mathbf{a}_t$ for $t = 1, \ldots, 7$ are given as $[0.5, 0.5]^T$, $[1.5, 1.5]^T$, $[0, 2]^T$, $[0, 2]^T$, $[1.5, -1.5]^T$, $[0.5, -1.5]^T$, $[1.5, -1.5]^T$. Here, we mention that the parameters of the matrix $\mathbf{H}_t$ are chosen in such a way that the maximum eigenvalue of $\mathbf{H}_t$ is always less than 1, in order to assure the stability of the model. The measurement noise variance is assumed to be $\sigma_e^2 = 1$, for all t. The process noise is chosen to be

the same as before. i.e., $\mathbf{q}_t \sim \mathcal{N}(\mathbf{0}_{100}, 0.001\mathbf{I}_{100})$ for all t. We assume that at time $t=0$ the estimation error covariance is $\mathbf{\Sigma}_0 = \mathbf{I}_{100}$. The performance threshold γ' is dynamically computed by scaling the best case MSE, i.e., $\mathrm{tr}[\mathbf{\Sigma}_t(\mathbf{1}_{100})]$, by $\zeta = 1.3$ every t. We perform $J=3$ iterations of the iterative algorithm at every snapshot t. In order to improve the speed, the selected sensing locations are computed by thresholding w.r.t. 0 (i.e., setting the non-zero elements to 1) rather than performing randomization on every snapshot.

The resulting sensor placement patterns for the fixed and time-varying $\mathbf{H}_t$ are shown in Fig. 7a and b, respectively. The y-axis represents the indices of the selected sensing locations indexed as shown in Fig. 3. The x-axis represents the time in minutes. In Fig. 7a, it is seen that more or less the same subset of sensing locations are selected with increasing time as the state error covariance converges to the steady state. The number of required sensing locations for individual snapshots is also decreasing with time due to the reduction of the state error.

On the contrary, in Fig. 7b different sensing locations are selected with time due to the time-varying state transition matrix $\mathbf{H}_t$. It is seen that the number and position of the optimal sensing locations, achieving a prescribed estimation performance are guided by the dynamics of the field, as well as the required performance (which is also dependent on the noise level) of the system.

6.3. Analysis of the performance metric

In this section, we compare the performance of the developed sensor placement algorithm with random sensor placement in terms of their respective mean square errors for different measurement noise variances (σ_e^2). For every noise variance, 100 random realizations of the selection vector $\tilde{\mathbf{w}}_t \in \{0, 1\}^{NN_s}$ are generated with the same number of 1 s generated by the proposed approach. The average MSE of all these realizations are compared with the achieved MSE using the proposed algorithm.

We consider two scenarios. In the first case, we consider the spatial sensor placement problem, i.e, $N_s = 1$, to estimate a stationary field in the service area shown in Fig. 3, i.e., $N=100$. The elements of the spatial covariance matrix $\tilde{\mathbf{\Gamma}}_t$ are generated using the exponential covariance function mentioned in (7) with $s_h=5$. The comparison of the MSE for the proposed approach and the average MSE of random sensor placement for different noise variances is shown in Fig. 8. The standard deviation of the MSEs for different realizations of the random placement are also shown for every σ_e^2. In the second case, we consider the field to be non-stationary with the same $\mathbf{H}_t = \mathbf{H}$ as mentioned in Section 6.2.1. In this case, we consider $N=25$ (5×5 square km service area with 25 pixels), $N_s = 3$ and solve for $\tilde{\mathbf{w}}_t \in \{0, 1\}^{75}$, i.e, in a multiple snapshot ahead fashion. The comparison of the MSE with random placement is shown in Fig. 9a. In the last two cases, we consider $\zeta = 2$ and $\zeta = 1.5$, respectively.

In the third case, we consider the single snapshot ahead sensor

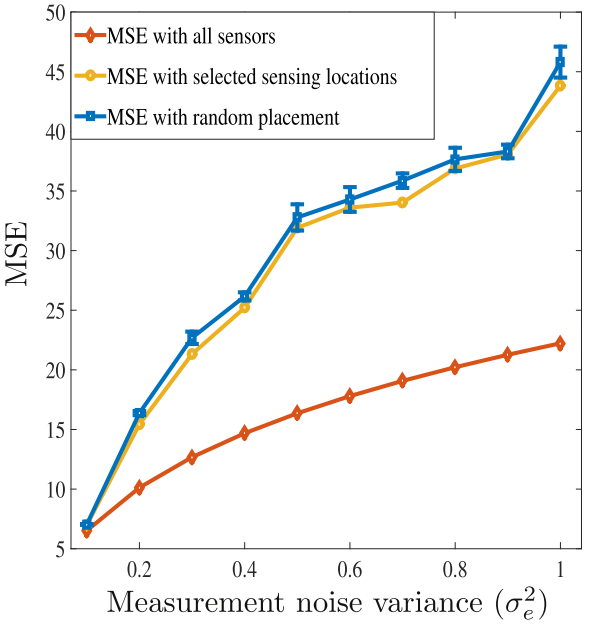

Fig. 8. MSE comparison with random sensor placement (stationary field; $N = 100$, $N_s = 1$).

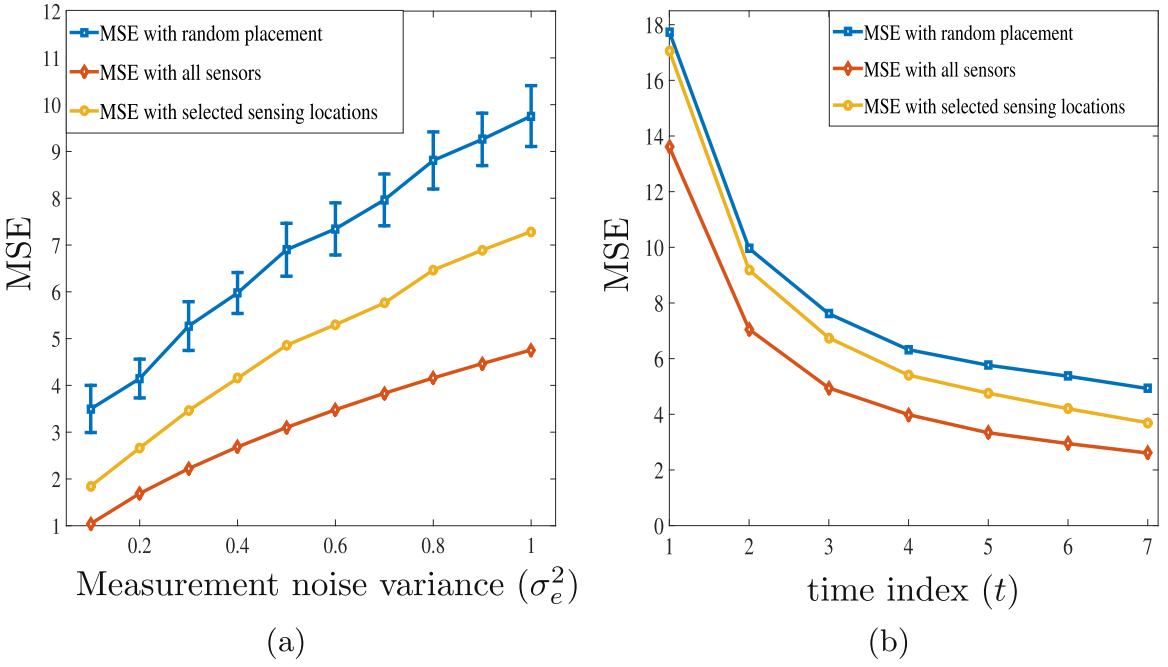

Fig. 9. (a) MSE comparison with random sensor placement (non-stationary field; $N=25$, $N_s = 3$) and (b) MSE comparison with random sensor placement (non-stationary field; $N=100$, $\sigma_e^2 = 1$).

placement problem, i.e., the performance metric is updated every iteration. We fix $\sigma_e^2 = 1$ and $\zeta = 2$. Every t, the MSE of the proposed approach and the average MSE for 100 different realizations of $\mathbf{w}_t \in \{0, 1\}^N$ (with the same number of 1 s generated by the proposed approach) are compared. We consider the same setup as mentioned in Section 6.2.2. The performance comparison is shown in Fig. 9b.

First of all, it is observed that the achieved MSE using the proposed approach is lower than randomly placing the sensors for a given number of available sensing locations. It is also seen that when the field is stationary with a smoothly varying spatial covariance function, on average random placement performs well. The reason behind this is that a uniform placement is close to optimal in order to estimate a stationary field as observed in Figs. 4b and 5b. So, the performance gap is not significant in this case, as long as the average MSE of the uniformly at random realizations is concerned. But there could be some realizations producing a high MSE, as seen by the standard deviation plot.

But when the field is non-stationary, the optimal sensor placement patterns are non-uniform over space and time as shown in Figs. 6a, b, and 7a, b, for multiple and single snapshot ahead sensor placements, respectively. They are mainly guided by the dynamics of the field. In this case, the performance gap between the proposed approach and a random placement is significant.

7. Conclusion

We have presented sparsity-enforcing sensor placement methods for the estimation of both stationary and non-stationary spatio-temporal environmental fields. The developed methodologies can be used for both offline and online field estimation applications. They exploit the space–time correlation information as well as the dynamics of the field to deploy sensors at the most informative locations in space and time. We have also developed a sparsity-enforcing iterative first order approach to select the sensing locations that achieve a prescribed estimation accuracy in terms of the MSE. We further compared the performance of the developed sensor placement approach with random sensor placement.

It is observed that for an exponentially decaying stationary covariance function, the higher the spatio-temporal correlation the less sensing locations are needed. For a non-stationary field, the number and the position of the selected sensing locations are controlled by the dynamics of the field, the required estimation accuracy, and the noise level. If the dynamics are not changing with time then the same set of sensors are selected with time once the posterior error covariance reaches a steady state.

Appendix

In this section, we compute the derivatives of the Lagrangian which are used in (42) and (43). The derivatives w.r.t. $\mathbf{w}$ and λ are given as

$$\nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}, \lambda) = \mathbf{1}_L + \lambda \nabla_{\mathbf{w}} h(\mathbf{w}), \quad \text{and} \tag{50}$$

$$\nabla_{\lambda} \mathcal{L}(\mathbf{w}, \lambda) = h(\mathbf{w}), \tag{51}$$

respectively. To compute $\nabla_{\mathbf{w}} h(\mathbf{w})$ we use the following identities for the differentiation of a scalar function of matrix and vector variables [26]. They are

$$\frac{\partial f(\mathbf{X}(\mathbf{w}))}{\partial [\mathbf{w}]_l} = \mathrm{tr}\left[\left(\frac{\partial f(\mathbf{X}(\mathbf{w}))}{\partial \mathbf{X}(\mathbf{w})} \right)^T \frac{\partial \mathbf{X}(\mathbf{w})}{\partial [\mathbf{w}]_l} \right], \quad \text{and}$$

$$\frac{\partial \,\mathrm{tr}[\mathbf{X}^{-1}]}{\partial \mathbf{X}} = -(\mathbf{X}^{-2})^T,$$

where $l = 1, \ldots, L$. Now using the above identities we can compute,

$$\nabla_{\mathbf{w}} h\Big(\mathbf{w}\Big) = \nabla_{\mathbf{w}} \Big[\mathrm{tr}\Big[\mathbf{Z}^{-1} + \sigma_e^{-2} \mathrm{diag}(\mathbf{w}) \Big]^{-1} \Big]$$
$$= \sigma_e^{-2} \mathrm{diag}\Big(-\Big[\mathbf{Z}^{-1} + \sigma_e^{-2} \mathrm{diag}(\mathbf{w}) \Big]^{-2} \Big). \tag{52}$$

Extended derivation: We consider $f(\mathbf{X}(\mathbf{w})) = h(\mathbf{w})$, where $\mathbf{X}(\mathbf{w}) = [\mathbf{Z}^{-1} + \sigma_e^{-2} \mathrm{diag}(\mathbf{w})]$. Now, we compute the elements of the vector $\frac{\partial f(\mathbf{X}(\mathbf{w}))}{\partial \mathbf{w}} = \frac{\partial h(\mathbf{w})}{\partial \mathbf{w}} = \nabla_{\mathbf{w}} h(\mathbf{w})$ by,

$$\frac{\partial f(\mathbf{X}(\mathbf{w}))}{\partial [\mathbf{w}]_1} = \mathrm{tr}\left[\left(\frac{\partial f(\mathbf{X}(\mathbf{w}))}{\partial \mathbf{X}(\mathbf{w})} \right)^T \frac{\partial \mathbf{X}(\mathbf{w})}{\partial [\mathbf{w}]_1} \right]$$
$$= \mathrm{tr}\left[-[\mathbf{Z}^{-1} + \sigma_e^{-2} \mathrm{diag}\Big(\mathbf{w}\Big)]^{-2} \sigma_e^{-2} \mathbf{A}_1 \right],$$

where $\mathbf{A}_1$ is an $L \times L$ matrix with only one non-zero element at $(1, 1)$ given as $[\mathbf{A}_1]_{11} = 1$. Similarly,

$$\frac{\partial f(\mathbf{X}(\mathbf{w}))}{\partial [\mathbf{w}]_L} = \mathrm{tr}\left[\left(\frac{\partial f(\mathbf{X}(\mathbf{w}))}{\partial \mathbf{X}(\mathbf{w})} \right)^T \frac{\partial \mathbf{X}(\mathbf{w})}{\partial [\mathbf{w}]_L} \right] \tag{53}$$

$$\frac{\partial f(\mathbf{X}(\mathbf{w}))}{\partial [\mathbf{w}]_L} = \mathrm{tr}\left[-[\mathbf{Z}^{-1} + \sigma_e^{-2} \mathrm{diag}\Big(\mathbf{w}\Big)]^{-2} \sigma_e^{-2} \mathbf{A}_L \right], \tag{54}$$

where $\mathbf{A}_L$ is an $L \times L$ matrix with only one non-zero element at (L, L) given as $[\mathbf{A}_L]_{LL} = 1$. So, the overall gradient is computed as

$$\nabla_{\mathbf{w}} h(\mathbf{w}) = \left[\frac{\partial f(\mathbf{X}(\mathbf{w}))}{\partial [\mathbf{w}]_1}, \ldots, \frac{\partial f(\mathbf{X}(\mathbf{w}))}{\partial [\mathbf{w}]_L} \right]^T. \tag{55}$$

Substituting the elements we have $\nabla_{\mathbf{w}} h(\mathbf{w}) = \sigma_e^{-2} \mathrm{diag}\left(-[\mathbf{Z}^{-1} + \sigma_e^{-2} \mathrm{diag}(\mathbf{w})]^{-2} \right)$.

References

[1] S. Liu, A. Vempaty, M. Fardad, E. Masazade, P. Varshney, Energy-aware sensor selection in field reconstruction, IEEE Signal Process. Lett. 21 (12) (2014) 1476–1480.

[2] S. Chepuri, G. Leus, Sparsity-promoting sensor selection for non-linear measurement models, IEEE Trans. Signal Process. 63 (3) (2015) 684–698.

[3] E. Masazade, M. Fardad, P.K. Varshney, Sparsity-promoting extended Kalman filtering for target tracking in wireless sensor networks, IEEE Signal Process. Lett. 19 (12) (2012) 845–848.

[4] S. Liu, E. Masazade, M. Fardad, P.K. Varshney, Sparsity-aware field estimation via ordinary kriging, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2014, pp. 3948–3952.

[5] V. Roy, G. Leus, Correlation-aware sparsity-enforcing sensor placement for spatio-temporal field estimation, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, 2015, pp. 340–343.

[6] S. Joshi, S. Boyd, Sensor selection via convex optimization, IEEE Trans. Signal Process. 57 (2) (2009) 451–462.

[7] S. Boyd, S. Vandenberghe, Convex Optimization, Cambridge University Press, 2009.

[8] H. Jamali-Rad, A. Simonetto, G. Leus, Sparsity-aware sensor selection: centralized and distributed algorithms, IEEE Signal Process. Lett. 21 (2) (2014) 217–220.

[9] A. Krause, A. Singh, C. Guestrin, Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies, J. Mach. Learn. Res. 9 (2008) 235–284.

[10] E. Feron, C. Olivier, Targets, sensors and infinite-horizon tracking optimality, in: Proceedings of the 29th IEEE Conference on Decision and Control, IEEE, 1990, pp. 2291–2292.

[11] W. Zhang, M.P. Vitus, J. Hu, A. Abate, C.J. Tomlin, On the optimal solutions of the infinite-horizon linear sensor scheduling problem, in: Proceedings of the

49th IEEE Conference on Decision and Control, IEEE, 2010, pp. 396–401.
[12] S. Liu, M. Fardad, P.K. Varshney, E. Masazade, Optimal periodic sensor scheduling in networks of dynamical systems, IEEE Trans. Signal Process. 62 (12) (2014) 3055–3068.
[13] C. Research Inc., CVX: Matlab Software for Disciplined Convex Programming, version 2.0, ⟨http://cvxr.com/cvx⟩, August 2012.
[14] J.F. Sturm, Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones, Optim. Methods Softw. 11 (1–4) (1999) 625–653.
[15] J. Ranieri, A. Chebira, M. Vetterli, Near-optimal sensor placement for linear inverse problems, IEEE Trans. Signal Process. 62 (5) (2014) 1135–1146.
[16] N. Cressie, K. Wikle, Statistics for Spatio-Temporal Data, John Wiley & Sons, 2011.
[17] I. Rodriguez-Iturbe, M. Marani, P. D'Odorico, A. Rinaldo, On space–time scaling of cumulated rainfall fields, Water Resour. Res. 34 (12) (1998) 3461–3469.
[18] E. Candes, M. Wakin, S. Boyd, Enhancing sparsity by reweighted ℓ_1 minimization, J. Fourier Anal. Appl. 14 (5–6) (2008) 877–905.
[19] C.K. Wikle, A kernel-based spectral model for non-Gaussian spatio-temporal processes, Stat. Modell. 2 (4) (2002) 299–314.
[20] C.K. Wikle, N. Cressie, A dimension-reduced approach to space–time Kalman filtering, Biometrika 86 (4) (1999) 815–829.
[21] F. Sigrist, H.R. Künsch, W.A. Stahel, A dynamic nonstationary spatio-temporal model for short term prediction of precipitation, Ann. Appl. Stat. 6 (4) (2012) 1452–1477.
[22] S.M. Kay, Fundamentals of Statistical Signal Processing: Estimation Theory, PTR Prentice-Hall, Englewood Cliffs, NJ, 1993.
[23] R. Ababou, A.C. Bagtzoglou, E.F. Wood, On the condition number of covariance matrices in kriging, estimation, and simulation of random fields, Math. Geol. 26 (1) (1994) 99–133.
[24] S.P. Chepuri, G. Leus, Sparsity-promoting adaptive sensor selection for nonlinear filtering, in: International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2014, pp. 5080–5084.
[25] A. Nedić, A. Ozdaglar, Subgradient methods for saddle-point problems, J. Optim. Theory Appl. 142 (1) (2009) 205–228.
[26] K.B. Petersen, M.S. Pedersen, The Matrix Cookbook, vol. 7, Technical University of Denmark, 2008, p. 15.