Dynamic Field Estimation Using Wireless Sensor Networks: Tradeoffs Between Estimation Error and Communication Cost

Haotian Zhang, *Member, IEEE*, José M. F. Moura, *Fellow, IEEE*, and Bruce Krogh, *Fellow, IEEE*

***Abstract*—This paper concerns the problem of estimating a spatially distributed, time-varying random field from noisy measurements collected by a wireless sensor network. When the field dynamics are described by a linear, lumped-parameter model, the classical solution is the Kalman–Bucy filter (KBF). Bandwidth and energy constraints can make it impractical to use all sensors to estimate the field at specific locations. Using graph-theoretic techniques, we show how reduced-order KBFs can be constructed that use only a subset of the sensors, thereby reducing energy consumption. This can lead to degraded performance, however, in terms of the root mean squared (RMS) estimation error. Efficient methods are presented to apply Pareto optimality to evaluate the tradeoffs between communication costs and RMS estimation error to select the best reduced-order KBF. The approach is illustrated with simulation results.**

***Index Terms*—Communication cost, estimation error, field estimation, Kalman–Bucy filter, Pareto optimality, tradeoffs, wireless sensor networks.**

I. Introduction

WIRELESS sensor networks (WSNs) are promising for large-scale environmental monitoring and control applications, such as temperature regulation in office buildings [1] or data centers [2], soil moisture monitoring [3], and habitat monitoring [4]. These applications call for estimating correlated dynamical fields, often at locations other than where the sensors are placed. This motivates the following problem: Estimate the values of a dynamic field at specific locations of interest $R = \{r_1, \ldots, r_q\}$ based on measurements at sensor locations $S = \{s_1, \ldots, s_p\}$, where R is not necessarily contained in S.

The field of interest is a physical quantity, say temperature, over a space that we discretize into voxels (or pixels). The field dynamics are described by a linear, lumped-parameter model, whose state collects the field values at all the voxels. The field correlation is captured by the state intracorrelation, i.e., the correlation among the components of the state vector. A Kalman-Bucy filter (KBF) provides optimal estimates of the field in minimal root-mean-squared (RMS) error sense. Bandwidth and energy constraints can preclude the direct centralized implementation of the centralized KBF for a large-scale field, however. We propose an alternative based on deriving reduced-order input-output models using graph-theoretic techniques. These reduced-order models are tuned to describe the field dynamics at the locations of interest. The proposed procedure identifies state variables that can be eliminated and other state variables that are reassigned as input variables in the reduced models. This makes it possible to construct reduced-order KBFs that estimate the field at the desired locations using fewer sensors and lower-dimensional state vectors, thereby reducing the amount of communication needed to acquire the data from the WSN. The penalty paid is that the quality of the estimates may also be reduced, when compared to the full-order KBF that uses all of the sensor values. We show that the quality of the estimation is not determined strictly by the number of sensors that are used; it is also influenced by the locations of the sensors employed by the KBF relative to the locations at which the field is to be estimated.

The above approach usually leads to many possible reduced-order KBFs. This raises the problem of how to select one KBF from among the many alternatives. We choose estimation error and communication cost as performance metrics to compare different reduced-order KBFs and study tradeoffs among them. We apply Pareto optimality to balance between these two competing goals: minimize the RMS estimation error in the field estimation and minimize the communication cost required by the estimator. Simulation results illustrate these tradeoffs under Pareto optimality.

We organize the paper as follows. The following section presents the problem formulation and defines the KBF required to estimate the values of the field at the locations specified in R. Section III presents the graph-based approach to constructing reduced-order KBFs. Section IV focuses on the RMS estimation error for different KBFs and presents an easily computable approximation to the RMS error. Section V considers the communication cost associated with the KBFs and derives from discrete event simulations an easily computable approximation to the communication cost. Section VI studies by Pareto optimality the tradeoffs between estimation accuracy and communication cost and illustrates the approach with simulation results. Finally, Section VII summarizes the

Manuscript received November 22, 2006; accepted December 23, 2008. First published February 10, 2009; current version published May 15, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Qing Zhao. Research supported by NSF Grants ECS-0225449 and CNS-0428404.

H. Zhang was with the Department of Electrical and Computer Engineering Carnegie Mellon University, Pittsburgh, PA 15213 USA. He is now with LSI Corporation, Longmont, CO 80501 USA (e-mail: haotian@ieee.org).

J. M. F. Moura and B. Krogh are with the Department of Electrical and Computer Engineering Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: moura@ece.cmu.edu; krogh@ece.cmu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2009.2015110

contributions of the paper and discusses directions for future research. A preliminary version of the graph-based method presented in Section III appeared in [5].

II. Problem Formulation

We consider the problem of estimating a real-valued, time-varying random field $F_t(\mathbf{s})$ defined over a planar region $D \subset \mathbb{R}^2$, where t and $\mathbf{s} \in D$ are the time and spatial variables indexing the field, respectively. The field is monitored at sampling times $k = 0, 1, \ldots$ by p sensors at locations in the set $S = \{\mathbf{s}_1, \ldots, \mathbf{s}_p\} \subset D$. The measurements at these locations at time k are given by the vector

$$z(k) = [F_k(\mathbf{s}_1), \ldots, F_k(\mathbf{s}_p)]^T + \eta(k) \tag{1}$$

where $\eta(k) \in \mathbb{R}^p$ denotes zero-mean, white Gaussian random measurement noise. Our objective is to use the sensor measurements to estimate the values of the dynamic field at q locations in the set

$$R = \{\mathbf{r}_1, \ldots, \mathbf{r}_q\} \subset D \tag{2}$$

where R is not necessarily a subset of S.

To be concrete, we take $F_t(\mathbf{s})$ to be a physical field, like a temperature, heat, or light intensity field. With continuous indexes (time and space), the field is commonly modeled by a partial differential equation (PDE), e.g., a parabolic equation like the diffusion heat equation (first-order partial derivative in t and second-order partial derivatives in the space variables). The following informal discussion motivates the models we use to develop KBFs. For example, a linear PDE model is given by

$$\frac{\partial F_t(\mathbf{s})}{\partial t} = \triangle F_t(\mathbf{s}) + w_t(\mathbf{s}) \tag{3}$$

where $w_t(\mathbf{s})$ is a forcing input, and $\triangle$ is the Laplacian operator

$$\triangle(\cdot) = \frac{\partial^2(\cdot)}{\partial s_1^2} + \frac{\partial^2(\cdot)}{\partial s_2^2}.$$

The PDE model is completed with an initial condition $\forall \mathbf{s} \in D$, $F_{t_0}(\mathbf{s})$ at $t = t_0$, and a boundary condition $\forall t \geq t_0$, $F_t(\mathbf{s}) = \phi_t(\mathbf{s}), \mathbf{s} \in \partial D$, where ∂D is the boundary of the domain D.

Rather than working with continuous indices, it is common to discretize the continuous indexes model and work with PDEs. For simplicity, discretize the domain so that D is a regular finite lattice. Take

$$t = k\delta_t, s_1 = i\delta_s, \text{ and } s_2 = j\delta_s, k \geq 0, 1 \leq i, j \leq N$$

and define

$$F_k(i,j) = F_{k\delta_t}(i\delta_s, j\delta_s).$$

Standard discretization of the PDE model, e.g., central or forward Euler difference, leads to (for a linear PDE, assumed in the sequel)

$$F_{k+1}(i,j) = \sum_{(l,m)\in\Omega_{i,j}} \alpha_{k,l,m} F_k(l,m) + w_{k,l,m} \tag{4}$$

where $k \geq 0$, $1 \leq i,j \leq N$, $\{\Omega_{i,j}\}_{ij}$ defines the *neighborhood* structure of the discrete pde resulting from the stencil (i.e., difference scheme) used, and $\alpha_{k,l,m}$ are the coefficients coming from the discretization of the PDE operator. Notice that usually the discretizing lattice is assumed to be finite.

To get a compact notation, we stack the values of the field at a given time k over all sites of the lattice (pixels, or voxels) in a vector $x(k)$, referred to as the *state* of the field. If lexicographic order (scan from top to bottom, left to right) is used, the state would be the N^2 dimensional vector

$$x(k) = [F_k(1,1) \cdots F_k(1,N) \cdots F_k(N,N)].$$

Below we will assume a different scanning order. In the sequel, we take $n = N^2$, so that $x(k) \in \mathbb{R}^n$. The model (4) now becomes a lumped parameter, discrete-time linear (LTI) state equation of the form

$$x(k+1) = Ax(k) + Bu(k) + w(k). \tag{5}$$

Equation (5) introduces $u(k) \in \mathbb{R}^m$ that accounts for m input variables at locations where there are sources driving the field. The vector $w(k)$ collects the noise inputs. In this paper, we assume that the model is linear time invariant (LTI). Note that here we discussed a model derived from basic principles (from a PDE model). More generally, and for particular applications, the LTI model might be obtained from sensor data using system identification techniques [5]. Also, the underlying model might be nonlinear, e.g., in fluid dynamics these models derive from highly nonlinear Navier–Stokes equations under appropriate simplifying assumptions, in which case (5) results from dynamical linearization of the nonlinear model, a common technique in applying Kalman filtering to nonlinear models. We have used this approach in data assimilation studies in ocean circulation, for example, [6] and [7]. For additional examples, see [8].

Based on the above motivation, we proceed with the formulation of the problem considered in this paper. As stated in (1), the field is measured by p sensors. In this section, we assume that all input variables $u(k)$ are included in the noisy measurements, which implies $m \leq p$. In practice this is not necessarily the case; Section III describes how to handle unmeasurable inputs. Without loss of generality, we assume the inputs are at locations $\{\mathbf{s}_1, \ldots, \mathbf{s}_m\}$. Thus, the first m elements of the measurement vector $z(k)$, denoted $\widetilde{u}(k)$, correspond to noisy measurements of $u(k)$; that is,

$$u(k) = \widetilde{u}(k) - \widetilde{\eta}(k) \tag{6}$$

where $\widetilde{\eta}(k)$ denotes the first m elements of the measurement noise vector $\eta(k)$. Substituting this expression for $u(k)$ into (5) gives the system $\mathbf{S}$

$$\mathbf{S}: x(k+1) = Ax(k) + B\widetilde{u}(k) + \widetilde{w}(k) \tag{7}$$

where $\widetilde{w}(k) = w(k) - B\widetilde{\eta}(k)$. Note that (7) is a standard LTI state equation and that the input $\widetilde{u}(k)$ is measured by the sensors.

We assume the state vector includes the values of the field at the remaining $p - m$ sensor locations plus the values of the field at any locations in R that are not in S. Without loss of generality, we let the first $p-m$ components of $x(k)$ correspond to the last $p-m$ components of $z(k)$ and define the output vector $y(k)$ as

$$y(k) = Cx(k) + v(k) \tag{8}$$

where $C = [I_{(p-m)\times(p-m)} | 0_{(p-m)\times(n-p+m)}]$ and $v(k)$ is defined as the last $p-m$ elements of $\eta(k)$. Finally, without loss of generality, we assume the last q elements of the state vector give the values of the field at the q locations in R, the points at which the field is to be estimated, and let

$$x^r = [x_{n-q+1}, \dots, x_n]^T \tag{9}$$

denote the vector of these state variables.

Remark: Although the model (7) and (8) only exhibit explicitly the time dependence k and not the space dependence, we notice that the state collects the field values at the voxels (spatial sites) of interest in the domain D. The spatial structure of the field, i.e., the spatial correlation among field values at different pixels, is encapsulated by the system matrices A and B. Also, given a particular location $r \in R$, there will be sensors that will be closer to r than others. So, although not directly apparent in the model equations, there is a notion of physical proximity between components of the state vector.

Using (7) and (8), the values of the field at the locations corresponding to the state variables, and hence at the locations in R, can be estimated using a standard steady-state Kalman–Bucy filter (KBF) [9] given by

$$\begin{cases} \bar{x}(k) = A\widehat{x}(k-1) + B\widetilde{u}(k-1) \\ \widehat{x}(k) = \bar{x}(k) + \mathcal{K}(y(k) - C\bar{x}(k)) \end{cases} \tag{10}$$

where $\bar{x}(k)$ is the *a priori* state estimate at time step k before processing the current measurement $y(k)$, $\widehat{x}(k)$ is the *a posteriori* state estimate at step k, and $\mathcal{K}$ is the steady-state Kalman gain matrix.

Equation (10) is a centralized KBF that uses all of the sensors to estimate the field at the locations in R. In large-scale applications where there are many sensors covering a large area D, bandwidth and time constraints may limit the ability to collect data from every sensor at every sampling time. If the points of interest in R are localized in a small region within D, it would be expected that not all sensor values from locations in S are needed to estimate the field at the locations of interest in R. This presents an opportunity to reduce the amount of communication needed to perform state estimation by possibly sacrificing the accuracy of the field estimates if fewer sensors are used. This leads to the following problems addressed in the remainder of the paper: 1) how to construct reduced-order KBFs (with lower dimensional states) that use fewer sensors and 2) how to select the reduced-order KBF that achieves the "best" tradeoff between the accuracy of the field estimates and the cost of communication. The following section deals with the first problem and the remaining sections deal with the second problem.

III. Graph-Based Approach to Model Reduction

To obtain estimates of the field at the locations in R using a subset of the sensors at the locations in S, it is reasonable to select sensors that are in the vicinity of the locations in R (see the Remark in Section II regarding physical proximity of sensors to site locations). In this section we present a method for choosing subsets of sensors for which a dynamic model can be constructed that is equivalent to the original LTI model (7) in the sense that the model generates exactly the same field values at the locations in R when there is no process noise. Our approach is based on the concept of *system digraphs* for LTI systems from [10], which we modify slightly for our purposes and rename as *structure digraphs*.

Definition 1 (Structure Digraph): Let the LTI system $\mathbf{S}$ be described by

$$x(k+1) = Ax(k) + Bu(k) \tag{11}$$

where $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ and $B = [b_{ij}] \in \mathbb{R}^{n\times m}$, and, at time k, the input and state variables are $U = \{u_1, u_2, \dots, u_m\}$ and $X = \{x_1, x_2, \dots, x_n\}$, respectively. The *structure digraph* of (11) is a tuple $\mathbf{D} = (V, E)$ with *vertex set* $V = U \cup X$ and *edge set* $E \subseteq V \times V$, where $(v, v') \in E$ iff: $(v, v') = (x_j, x_i) \in X \times X$ and $a_{ij} \neq 0$, or $(v, v') = (u_j, x_i) \in U \times X$ and $b_{ij} \neq 0$. □

We refer to the structure digraph vertices in U as *input vertices* and the vertices in X as *state vertices*. The directed arcs into each state vertex indicate which state variables and input variables influence the next value of that state variable in the state transition (11). There are no arcs into input vertices.

To reduce the dimensionality of the system and handle unmeasured inputs we introduce the concept of *cut-point sets*. We partition the vertex set V into the *measurable vertex set*, denoted V_m, and the *unmeasurable vertex set*, denoted V_u. Each of these sets can contain both input and state vertices. Let $\overline{V'}$ denote the complement of a subset of vertices $V' \subseteq V$ with respect to V; that is, $\overline{V'} \triangleq V - V'$. Given these definitions and notation, the concept of a cut-point set is defined as follows.

Definition 1 (Cut-Point Set): Given the system $\mathbf{S}$ (11) with structure digraph $\mathbf{D} = (V, E)$, and a set of state vertices $X_s \subseteq X$, the *cut-point set P_c for the set of state vertices X_s* is a set of measurable vertices $P_c \subseteq (V_m - X_s)$ for which there exists an *extended state vertex set* $X'_s \supseteq X_s$ such that:

1) if $v \in P_c$, there exists $v' \in X'_s$ such that $(v, v') \in E$;
2) if $v \in \overline{P_c \cup X'_s}$, there is no $v' \in X'_s$ such that $(v, v') \in E$. □

The following example illustrates the concepts of structure digraph and cut-point sets.

Example 1: Suppose that the system matrices A and B in (11) are

$$A = \begin{bmatrix} 0 & a_{12} & 0 & 0 & 0 & 0 & 0 & 0 & a_{19} \\ a_{21} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{38} & 0 \\ 0 & 0 & 0 & a_{44} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{57} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{67} & a_{68} & 0 \\ 0 & 0 & 0 & a_{74} & a_{75} & a_{76} & 0 & 0 & 0 \\ 0 & a_{82} & a_{83} & 0 & 0 & a_{86} & 0 & 0 & 0 \\ a_{91} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

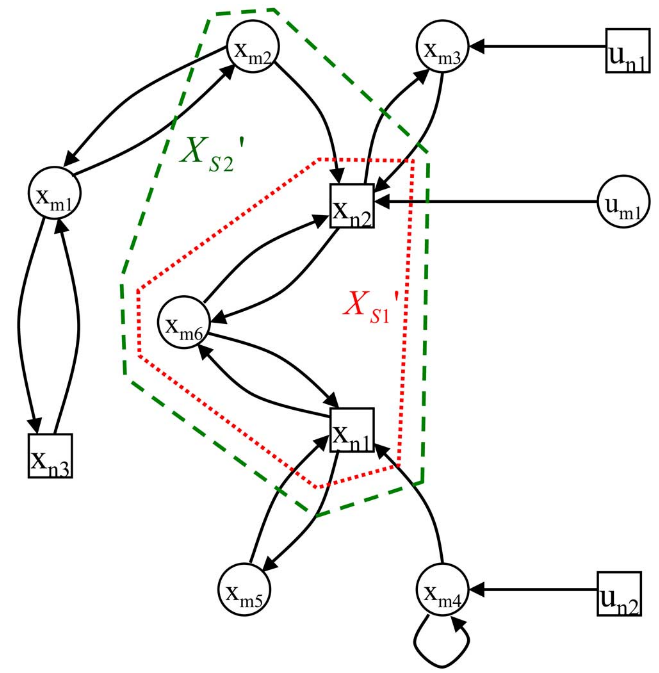

Fig. 1. Structure digraph for Example 1: circle vertices are measurable; square vertices are unmeasurable; $x_{mi}(k)$—measurable state vertex; $x_{ni}(k)$—unmeasurable state vertex; $u_{mi}(k)$—measurable input vertex; $u_{ni}(k)$—unmeasurable input vertex. The dotted and dashed lines indicate two cut-point sets for $X_s = \{x_{n1}(k), x_{m6}(k)\}$.

and

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & b_{32} & 0 \\ 0 & 0 & b_{43} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ b_{81} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

where the parameters a_{ij}, b_{ij}, denote nonzero entries in the matrices A and B, respectively. Assume that the state vector is $x(k) = [x_{m_1}(k), \ldots, x_{m_6}(k), x_{n_1}(k), x_{n_2}(k), x_{n_3}(k)]^T$, where $x_{m_i}(k)$ and $x_{n_j}(k)$ are measurable and unmeasurable states, respectively, and the input vector is $u(k) = [u_{m_1}(k), u_{n_1}(k), u_{n_2}(k)]^T$, where $u_{m_i}(k)$ and $u_{n_j}(k)$ are measurable and unmeasurable inputs, respectively. Fig. 1 shows the digraph for this system. The circle nodes represent measurable vertices and the square nodes represent unmeasurable vertices. A directed edge (v_i, v_j) shows the dependence of vertex v_j on vertex v_i. For example, there is a directed edge $(u_{n_1}(k), x_{m_3}(k))$ in the system digraph in Fig. 1 since b_{32} is nonzero, implying that the update of the state $x_{m_3}(k)$ depends in part on the value of the input $u_{n_1}(k)$.

We now explain cut-point sets with reference to the system digraph in Fig. 1. Let the unmeasurable state vertex set $X_s = \{x_{n_1}, x_{m_6}\}$. One of the cut-point sets for X_s is $P_{c1} = \{x_{m_2}, x_{m_3}, x_{m_4}, x_{m_5}, u_{m_1}\}$, with the corresponding extended state vertex set $X'_{s1} = \{x_{n_1}, x_{n_2}, x_{m_6}\}$ encircled by the dotted line in Fig. 1. The other cut-point set for X_s is $P_{c2} = \{x_{m_1}, x_{m_3}, x_{m_4}, x_{m_5}, u_{m_1}\}$, with the corresponding extended state vertex set $X'_{s1} = \{x_{n_1}, x_{n_2}, x_{m_2}, x_{m_6}\}$ encircled by the dashed line in Fig. 1. P_{c1} and P_{c2} are the only two cut-point sets for X_s. We can see the following for either cut-point set P_c. There is at least one edge pointing to vertices in the corresponding extended state vertex set X'_s; all the edges pointing to vertices in X'_s are from vertices in P_c or other vertices in X'_s. These facts reflect the conditions in the definition of cut-point sets. □

Reduced-order models for a given LTI system $\mathbf{S}$ with state model (11) with structure digraph $D_{\mathbf{S}} = (V, E)$ are constructed as follows. Let $X_s \subset X$ be a given subset of state variables from $\mathbf{S}$, and, for a given cut-point set P_c for X_s with a corresponding extended state vertex set X'_s, define the graph $D_{\mathbf{S}'} = (V', E')$, where $V' = X'_s \cup P_c$ and $E' = V' \times V' \cap E$. Thus, $D_{\mathbf{S}'}$ is the subgraph of D with vertices in V'. We construct the reduced-order system $\mathbf{S}'$ with state variables $X' = X'_s$ and input variables $U' = P_c$ by letting x' and u' be column vectors of the variables in X'_s and P_c, and defining the state model for $\mathbf{S}'$ as

$$x'(k+1) = A'x'(k) + B'u'(k) \tag{12}$$

where the elements of the matrices $A' = [a'_{ij}]$ and $B' = [b'_{ij}]$ are given by

$$a'_{ij} = \begin{cases} a_{qr} & \text{if } (x'_j, x'_i) = (x_r, x_q) \in E' \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

and

$$b'_{ij} = \begin{cases} a_{qr} & \text{if } (u'_j, x'_i) = (x_r, x_q) \in E' \\ b_{qr} & \text{if } (u'_j, x'_i) = (u_r, x_q) \in E' \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

As stated in the following theorem, the trajectories of the state variables for a reduced-order model $\mathbf{S}'$ constructed as described above are identical to the trajectories of the corresponding states in the original system when the input sequence for the reduced-order system (which may include some of the original state variables) are identical to the corresponding input and state variable sequences in the original system.

Theorem 1 [5], [11]: Consider the LTI system $\mathbf{S}$ with: 1) state model given by (11) and structure digraph $\mathbf{D} = (V, E)$; 2) set of state variables $X_s \subset V$; 3) a cut-point set P_c for X_s with a corresponding extended state vertex set X'_s; 4) initial state and input sequence for $\mathbf{S}$ given by $x(0)$ and $u(0), u(1), \ldots$, respectively; and 5) $x(0), x(1), \ldots$, the resulting state trajectory for $\mathbf{S}$. Further, construct a reduced order system $\mathbf{S}'$ as described above and define the initial state $x'(0)$ and input sequence $u'(0), u'(1), \ldots$ by the values of the corresponding state and input variables in $x(0)$ and $u(0), u(1), \ldots$

Then, the values of the state variables in the state resulting trajectory $x'(0), x'(1), \ldots$ for $\mathbf{S}'$ are identical to the values of the corresponding state variables in the state trajectory $x(0), x(1), \ldots$, for $\mathbf{S}$.

Proof: The theorem follows by induction on the time index k. For $k = 0$, the state variables in $x'(0)$ equal the values of the corresponding state variables in $x(0)$ by definition of $x'(0)$. Now, suppose for arbitrary $k > 0$ that the state variables in $x'(k)$ equal the values of the corresponding state variables in $x(k)$, and consider element i of $x'(k+1)$, denoted by $x'_i(k+1)$. The ith state variable for $\mathbf{S}'$ corresponds to the rth state variable for $\mathbf{S}$ for some $1 \leq q \leq n$. We need to show that $x'_i(k+1) =$

$x_q(k+1)$. From the definition of the elements of A' and B', we have

$$\begin{aligned} x'_i(k+1) &= \sum_{\{r \mid \exists\, j \ni (x'_j = x_r \,\vee\, u'_j = x_r) \,\wedge\, (x_r, x_q) \in E'\}} a_{qr} x_r(k) \\ &+ \sum_{\{r \mid \exists\, j \ni u'_j = x_r \,\wedge\, (u_r, x_q) \in E'\}} b_{qr} u_r(k). \end{aligned} \tag{15}$$

From the definition of the cutset P_c and the structure digraph D' for $\mathbf{S}'$, (15) for $x'_i(k+1)$ is precisely the equation for $x_q(k+1)$ in $\mathbf{S}$. Therefore, the values of the state variables in $x'(k+1)$ are equal to the elements of the corresponding state variables in $x(k+1)$. □

The result in Theorem 1 uses the known property that the physical field arising from a PDE or pde and described by the lumped parameter model $\mathbf{S}$ in (11) is Markov.

To construct a reduced-order KBF for the field estimation problem, X_s is chosen as the state variables corresponding to the points for which the field values are to be estimated. If none of these state variables are measurable, at least one measurable state variable is added to X_s. We then construct a cut-point set for X_s and identify the corresponding extended state vertex set X'_s. Regarding the measurable state variables in X'_s as the outputs of the system, a KBF can be constructed as described in Section II to estimate the values of the field at the locations in R. Thus, each cut-point set results in a feasible KBF. Since there are usually multiple cut-point sets for a given set of desired locations X_s with corresponding extended state vertex sets X'_s, especially for large-scale WSNs, we have multiple choices of KBFs to implement the real-time estimation at the desired locations.

IV. RMS Estimation Error

In field estimation applications, it is clear that the accuracy of the estimates is an important performance goal. We let X_s collect the states at the locations where it is desired to estimate the field but there are no sensors; we consider that there is at least one location with a sensor. For simplicity, we assume a single location $R = \{r_0\}$ where the field is to be estimated. Therefore, a single state variable is of interest, which from (9) is given as

$$x^r = x_n.$$

Generalizing for the case when it is desired to estimate the field at more than a single location is straightforward. In practice, if these desired locations are not in close vicinity, it may be preferable to consider each location separately, i.e., to design a reduced-order model and the corresponding KBF to estimate the field at each individual location. This reduces the order of the model and of the KBF, which reduces the amount of communication (and therefore the energy) needed to perform state estimation.

The RMS estimation error $\mathcal{E}_{\text{RMS}}$ is usually taken as the measure of accuracy for the error performance of the KBF:

$$\mathcal{E}_{\text{RMS}} = \sqrt{E\{(\widehat{x}^r - x^r)^2\}} \tag{16}$$

where $\widehat{x}^r$ is the estimate of the field at r_0 provided by the KBF.

In large physical areas monitored by sensor networks, there may be an excessively large number of possible cut-point sets P_c associated with a given X_s and so there are many feasible KBFs that can be used to compute $\widehat{x}^r$. To study the tradeoffs among these, we need to evaluate the RMS error performance for each KBF, which may be too costly to compute directly. In this section, we present an alternative measure of performance that is simple to compute *a priori*, i.e., before actually deploying the sensor network in the field, and can serve as a surrogate measure for the RMS error. To motivate the discussion, we consider the following simulated scenario.

Spatial Distribution of Sensors: We consider that the field being monitored by the sensor network covers a large area $\mathcal{A}$. The sensors are deployed randomly over this area. To be specific, we assume that the sensors are distributed in $\mathcal{A}$ according to a Poisson distribution with parameter λ. This parameter represents the average number of sensors per unit area. The Poisson probability distribution has been adopted by many others as the distribution model for sensors in WSNs [12]–[14]. It is a reasonable model when the number of randomly deployed sensors is large [15]. With this model, the probability of l sensors being in a region $R_e \subset \mathcal{A}$ depends only on the area A of the region R_e and is given by

$$Pr[l \text{ sensors in } R_e] = \frac{e^{-\lambda A}(\lambda A)^l}{l!}. \tag{17}$$

The average number of sensors in an arbitrary shaped region of area A is λA.

For simplicity, we assume that the region of interest $\mathcal{A}$ is tesselated by a uniform grid and that the locations of the sensors and the locations where the field is to be estimated lie on the nodes of this grid. This assumption greatly simplifies our simulation study, but the resulting analysis and conclusions are valid for generic sensor networks where the sensors are arbitrarily placed on the plane, not necessarily on the vertices of a uniform grid.

We adopt as a motivating application the problem of air temperature monitoring from [11] in a region $\mathcal{A}$ consisting of 10×10 nodes on the 2-D coordinate plane.[1] The grid nodes, referred to as "pixels," are located at integer coordinates, as shown in Fig. 2. In this figure, the sensors, represented by circles, have been deployed according to the Poisson distribution with $\lambda = 0.5$. In this example, the total number of sensors is 53. We could consider several locations where to estimate the field; we focus here on the single location r_0, the pixel with coordinate $(6, 4)$ represented by the square in Fig. 2. This position was randomly selected.

To estimate the field at r_0, we select a state set X_s that includes r_0 and at least one sensor location, i.e., a measurable vertex. This will guarantee that any resulting extended state vertex set X'_s will include at least one sensor measurement. We can associate with X_s several cut-point sets. Fig. 3 shows an example of one of these cut-point sets, where the overlapped

[1] This example is presented to motivate the need to evaluate the trade off between estimation accuracy and communication cost. The details of the dynamic model and how the parameters can be estimated from experimental data are discussed in [5] and [11].

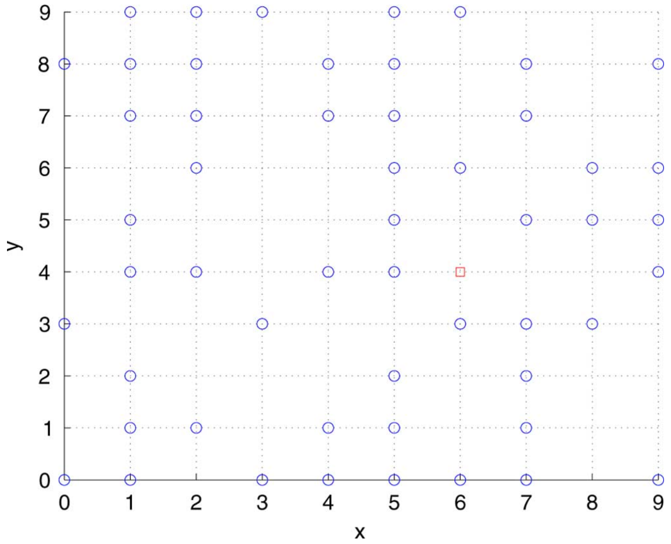

Fig. 2. Sensor distribution and the location to estimate: sensors—circles; location to estimate—square.

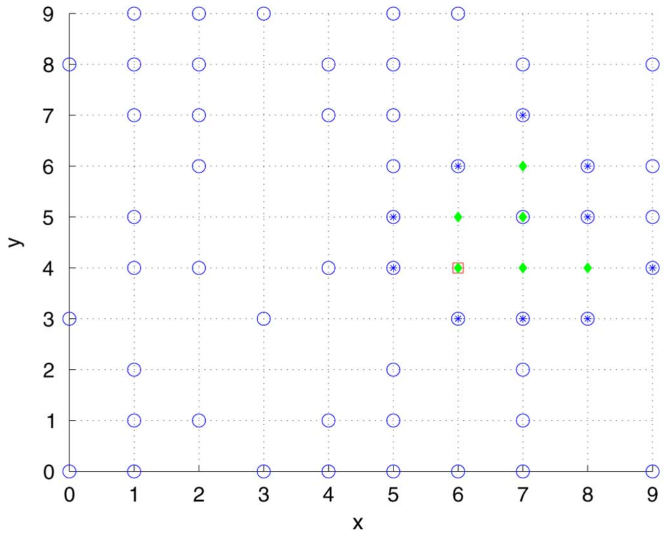

Fig. 3. Example of a cut-point set: the overlapped blue circle/stars are elements of the cut-point set, the green diamonds are elements of the extended state vertex set, the overlapped red-square/green-diamond is the desired location, and the overlapped blue-circle/green-diamond is the output of the corresponding KBF.

circles/stars are elements of the cut-point set, the diamonds are elements of X_s', the overlapped square/diamond is the desired location, and the overlapped circle/diamond is the output of the corresponding KBF. Using the approach described in Section III, we derive for each cut-point set a corresponding KBF. For the example in Fig. 2, we find, by exhaustive search, a total of 447 cut-point sets that correspond to 447 different KBFs. Each of these KBFs provides an estimate of the field at the desired location r_0.

Before comparing these 447 KBFs, we define the *size* of a KBF as follows.

Definition 3 Size of a KBF: The *size of a KBF* is the total number of its inputs and outputs, i.e., the size of a KBF equals $|P_c|+|X_s'\cap V_m|$—the total number of sensors used by the KBF. □

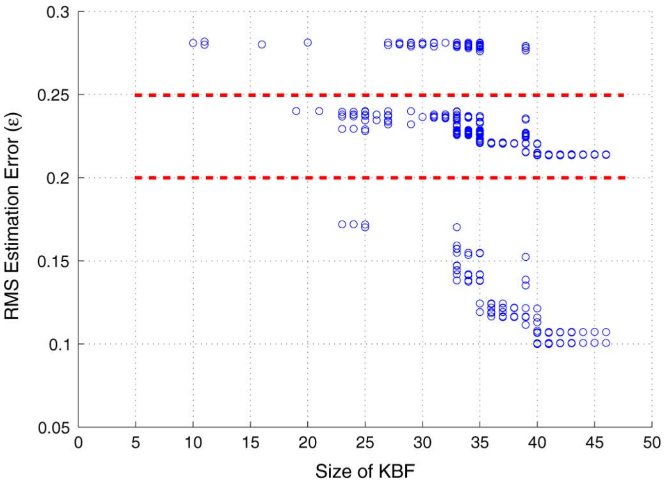

Fig. 4. Comparison of the estimation error performance of KBFs with different sizes: each circle represents a feasible KBF; the dashed lines are estimation error thresholds.

The size is the number of measurements used, either as inputs or outputs by the KBF. For example, the maximum possible size among all the 447 KBFs for the example in Fig. 2 is 53; this is the total number of sensors in region $\mathcal{A}$. Note that different KBFs may have the same size. One would expect that KBFs with larger sizes should lead to better RMS error performance. As we see next, this is not the case as proximity of the sensors to the locations of interest also influence the quality of the estimates because, as discussed in Section II, it is expected that the field at a pixel will exhibit higher correlation with the field at nearby locations than with the field at more distant locations.

To find a surrogate measure to the KBF RMS error that is simple to compute, we plot in Fig. 4 the RMS estimation error $\mathcal{E}_{\rm RMS}$ for each of the 447 KBFs versus its size. For each KBF, $\mathcal{E}_{\rm RMS}$ is calculated using (16) over 2000 time increments based on sensor measurements with Gaussian noise distribution $\mathcal{N}(0,1)$. Fig. 4 exhibits the following three interesting phenomena.

1) KBFs of the same size may have quite different RMS estimation error performance. For example, the $\mathcal{E}_{\rm RMS}$ of KBFs with size 35 ranges from 0.12 to 0.28.
2) KBFs can be explicitly grouped according to their RMS errors. For the example under study, we can group them into three sets. These three classes are defined by two thresholds $\mathcal{E}_{\rm RMS}=0.2$ and $\mathcal{E}_{\rm RMS}=0.25$, displayed by the two dashed lines in Fig. 4.
3) A key factor in determining the RMS error performance is the proximity of sensors to the location where we wish to estimate the field.

The first phenomenon shows that the size of the KBF is not the key factor to predict its estimation performance. For example, the best KBF of size 25 has a much smaller RMS error than the worst KBF of size 35. With respect to the second issue, the classes grouping the KBFs can be refined. In fact, each of the three major classes in Fig. 4 can be further divided into subgroups, though these groups are not as the classification of the major classes. For example, the KBFs in the class for which

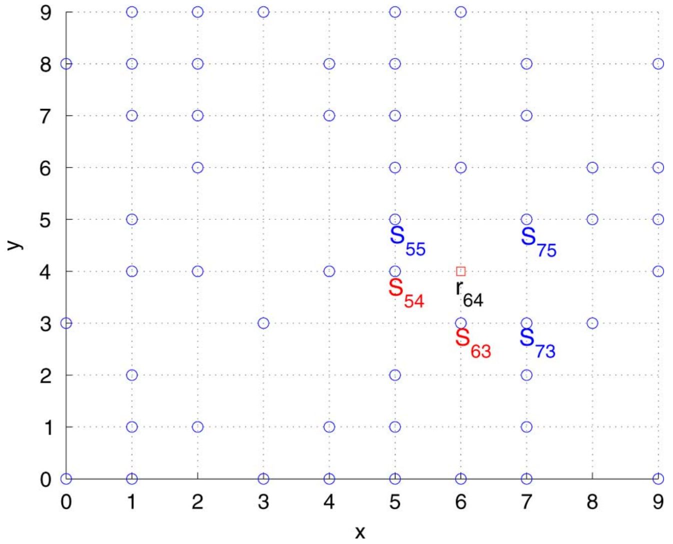

Fig. 5. Locations of the outputs of KBFs: the circles are sensors; the square is the location to estimate; s_{ij}'s are discussed outputs of KBFs.

$\mathcal{E}_{\rm RMS} < 0.2$ can clearly be grouped into several additional subclasses. Finally, the third point above shows that the estimation error performance of a KBF does not depend only on its size, but, more importantly, depends on how close its outputs are to the desired location.

The following two examples support these conclusions.

Example 2: We start by grouping the KBFs in Fig. 4 into three classes, but according to the following criteria. First, we label the sensor locations by using lexicographic order, starting from the bottom left location $(1,1)$ and proceeding from left to right and bottom to top. The sensor at location (i,j) is labeled as s_{ij}. We then consider the distance $d(s_{ij}, s_{i'j'})$ between nodes s_{ij} and $s_{i'j'}$ in the graph as the graph distance defined by the shortest path length between connecting these nodes. For example, $d(s_{63}, s_{64}) = 1$. From Fig. 5, we see that the two sensors s_{63} and s_{54} are closest to the desired location $(6,4)$, denoted by r_{64}, with distance of 1. We can group all the KBFs into three classes as follows:

1) Class C1: KBFs whose outputs include both s_{63} and s_{54};
2) Class C2: KBFs whose outputs include only one of s_{63} or s_{54};
3) Class C3: KBFs whose outputs include neither s_{63} nor s_{54}.

Fig. 6 shows the grouping of the KBFs in Fig. 4 according to the new criteria. We see that the groups are exactly the same as the groups resulting from the thresholds of the RMS error in Fig. 4. □

Example 3: From Fig. 5, the sensors closest to the desired location r_{64}, after s_{63} and s_{54}, are sensors s_{73}, s_{55}, and s_{75}; their distance to the desired location r_{64} is $\sqrt{2}$. We now refine Class C1, i.e., the group of KBFs where both s_{63} and s_{54} are simultaneously used as outputs, into four subclasses as follows:

1) Subclass SC1: KBFs whose outputs include all of s_{73}, s_{55}, and s_{75};
2) Subclass SC2: KBFs whose outputs include two of s_{73}, s_{55}, and s_{75};
3) Subclass SC3: KBFs whose outputs include one of s_{73}, s_{55}, and s_{75};

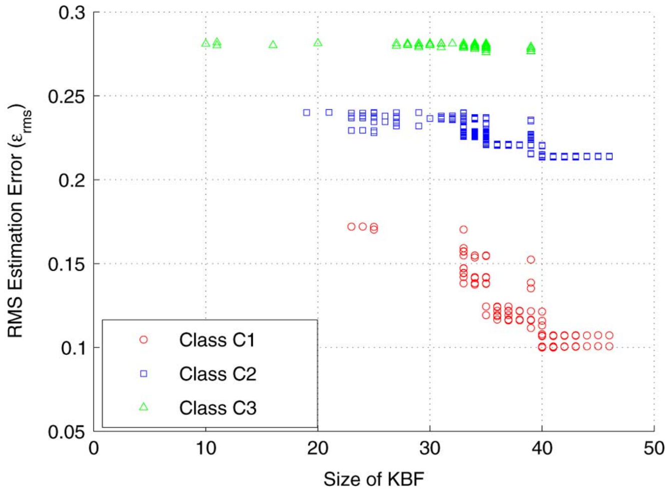

Fig. 6. Classification of all KBFs: Classes C1, C2, and C3 are represented by circles, squares, and triangles, respectively.

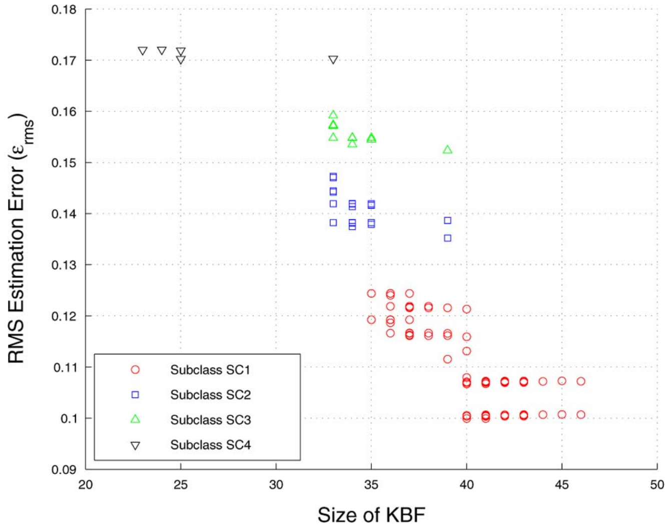

Fig. 7. Classification of the KBFs in Class C1: Subclasses SC1, SC2, SC3, and SC4 are represented by circles, squares, triangles, and inversed triangles, respectively.

4) Subclass SC4: KBFs whose outputs include none of s_{73}, s_{55}, and s_{75}.

The result of this classification scheme is shown in Fig. 7, which is equivalent to classifying the KBFs in Class C1 if we use three RMS error thresholds: $\mathcal{E}_{\rm RMS} = 0.13, 0.15$, and 0.16. Similarly, we could divide the classes C2 and C3 into subclasses, since a clustering similar to that in Fig. 7 appears if one zooms in on the C2 and C3 regions in Fig. 6. We can further refine each subclass according to whether or not the KBFs include the closest sensors. □

Alternative Cost: The above examples show that the estimation error performance of a KBF depends on the number and the locations of its outputs: the outputs that are the closest to the desired location play a more important role in determining the estimation error performance. This conclusion provides a simple and effective method to obtain an approximation to the RMS estimation error of the KBF based only on the topology information of the associated sensors. This is very important in practice, since the number of feasible KBFs is potentially very large. We call this error approximation the *RMS estimation error approximate cost*.

Definition 4 (RMS Estimation Error Approximate Cost): Assume that the KBF has m outputs corresponding to m sensors $\{s_1, s_2, \ldots, s_m\}$. Let r be the location where it is desired to estimate the field, and $d_i = \mathcal{D}(s_i, r)$ be the distance between sensor s_i and the desired location r. The RMS estimation error approximate cost $\widetilde{\mathcal{E}}_{\mathrm{RMS}}(K_f)$ is

$$\widetilde{\mathcal{E}}_{\mathrm{RMS}}(K_f) = c_1 - c_2 \left(\sum_{i=1}^{m} \frac{1}{d_i^{n_1}} \right)^{n_2} \tag{18}$$

where c_1 and c_2 are constants, and n_1 and n_2 are given exponents. □

The above definition can be easily extended to multiple desired locations. Assume there are q desired locations $\{r_1, r_2, \ldots, r_q\}$, and let $d_{ij} = \mathcal{D}(s_i, r_j)$ be the distance between sensor s_i and the desired location r_j. The RMS estimation error approximate cost $\widetilde{\mathcal{E}}_{\mathrm{RMS}}(K_f)$ can be taken to be

$$\widetilde{\mathcal{E}}_{\mathrm{RMS}}(K_f) = c_1 - \sum_{j=1}^{k} c_{2j} \left(\sum_{i=1}^{m} \frac{1}{d_{ij}^{n_{1j}}} \right)^{n_{2j}}. \tag{19}$$

In this paper, we focus on the single desired location problem, which illustrates the major tradeoff of interest.

To use the *RMS estimation error approximate cost* as a surrogate for the RMS error to evaluate the KBFs performance, we first fit the parameters in (18) to data collected with a small subset of the possible reduced-order KBFs. In other words, the values of c_1, c_2, n_1, and n_2 are determined empirically, using experimental data or simulation results based on the state-space models for a subset of the feasible KBFs. Once the parameters c and n have been determined, the RMS estimation error approximate cost will only depend on the knowledge of the distance between the desired location and the sensors associated with the output of the given KBF. This distance is found directly from the topology of the network and is trivial to compute.

We illustrate this methodology for the sensor distribution in Fig. 2. Fig. 8 shows a plot of the RMS estimation error approximate cost versus the actual RMS estimation error for all 447 feasible KBFs. Each feasible KBF is represented by a blue circle. The parameters c_1, c_2, n_1, and n_2 are determined by minimizing the mean square difference between the RMS estimation errors for a subset of the feasible KBFs and their corresponding RMS estimation error approximate costs. This subset is obtained by choosing at random 50 out of the 447 feasible KBFs. The resulting estimated parameters are $c_1 = 0.28, c_2 = 0.06, n_1 = 6$, and $n_2 = 1.2$. The slope of the linear curve fitting plot in Fig. 8 is around 1, which means that the RMS estimation error approximate cost is approximately equivalent to the actual RMS estimation error, though a polynomial curve may provide a slightly better fit. We repeated the estimation of the parameters of the approximate cost with several different sets of 50 KBFs chosen randomly with similar results. Therefore, we conclude that the RMS estimation error approximate cost given by (18) can be used as an alternative to the actual RMS estimation error.

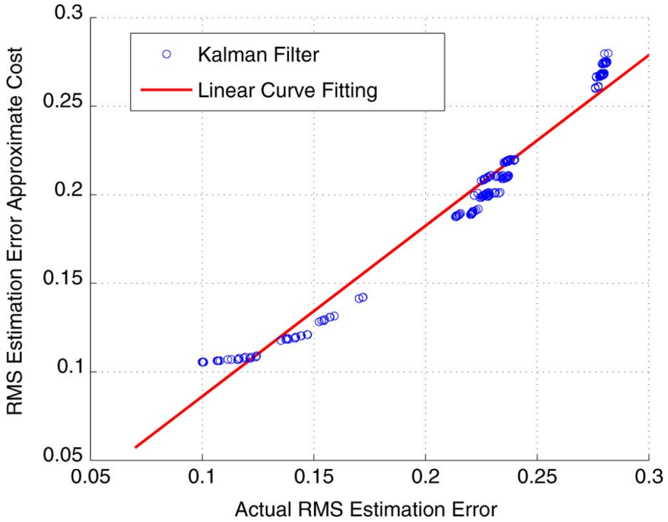

Fig. 8. RMS Estimation Error Approximate Cost versus Actual RMS Error: each circle represents a feasible KBF; the solid line is the linear curve fitting.

Instead of the actual RMS error, we will use in Section VI the RMS estimation error approximate cost $\widetilde{\mathcal{E}}_{\mathrm{RMS}}(K_f)$ given in (18) to study the tradeoffs among all feasible KBFs. This approach is much simpler since in general it is expensive to first evaluate the RMS error performance for all feasible KBFs.

V. Communication Costs for Alternative KBFs

Sensors in WSNs are normally untethered and autonomous, consuming energy from their own power sources, e.g., batteries, that are usually difficult or costly to replenish. Therefore, power consumption is a critical problem in most applications of WSNs. Power is consumed not only by sensing, but also by communication and data processing. For most passive sensors, the power consumed by sensing is very limited. Also, if the local processing is limited, the computational burden and related power consumption at each sensor node can be assumed to be negligible. Most of the power consumed by each sensor node is related to the transmission of the sensor data, which is related to the amount of data that will be transmitted. Therefore, we focus in this section on the communication cost. As in Section IV, this cost is expensive to compute for all feasible KBFs, because of their sheer number for reasonably large WSNs. Our main goal in this section is to derive an alternative cost that is simple to compute and can serve as a surrogate to the exact computational cost.

To evaluate the power consumption for each KBF, we introduce the following definition of *communication cost*.

Definition 5 (Communication Cost): The communication cost of a KBF in a WSN is the power consumed by its sensors to transmit their readings at each filtering updating step to the corresponding fusion center. □

We now explain how to evaluate the communication cost associated with a given KBF, K_f. We denote this communication cost by $\mathcal{C}(K_f)$. To evaluate this cost, we need to assume a model for the communication. For simplicity, we adopt the following assumptions.

1) The sensor measurements for the reduced-order KBF K_f are fused at a single fusion center associated with that K_f. This center may be a node in the WSN. This is very realistic in many applications when the computational capabilities at most nodes are limited. When the nodes are more powerful, distributed filtering algorithms [8], [16], [17] can be applied.
2) The communication between sensors and the fusion center is achieved by multi-hop transmission; the network deploys for each hop link the stop-and-wait ARQ protocol [18]. That is, a node will repeat the transmission of a (sensor reading) packet until an ACK indicating correct transmission is received. We discuss the details of the stop-and-wait ARQ protocol below.

The power consumption associated with a KBF, should consist of transmission power consumption for sensor reading packets, reception power consumption for sensor reading packets, transmission power consumption for ACKs, and reception power consumption for ACKs. A closed-form version for the communication cost is not possible given the complexity of this cost. We make the following assumptions that reflect the usual relative amounts of power consumption for various operations in WSNs:

1) reception power consumption is negligible compared to transmission power consumption;
2) ACK has a much shorter length than a sensor reading packet, so the power consumption for both transmission and reception of ACK is negligible;
3) the power consumption used for each transmission/retransmission is similar.

Based on the above assumptions, the communication cost $\mathcal{C}(K_f)$ can be represented by the total number of required transmissions, denoted by N_t, to transfer all the sensors readings used at each update step by K_f to the fusion center at a given location, i.e.,

$$\mathcal{C}(K_f) = N_t. \tag{20}$$

As in the previous section, computing the communication cost for every possible KBF is expensive, since there can be many potential alternative cut-point sets and corresponding KBFs. As done with the RMS estimation error, we look for an alternative equivalent cost function that is easier to compute. To motivate and support our definition below of a metric that is simple to compute and serves as a surrogate for the communication cost, we carry out a simulation study that shows that this new metric is a good indicator to the communication cost.

In our study, we regard the communication process in wireless sensor networks WSNs as a discrete event system, since the start and end of packet transmissions and the searching for the next hop in routing schemes are both discrete events that happen only at certain points in time. Accordingly, we use a discrete event simulator to simulate the communication process for all the available KBFs for the sensor distribution in Fig. 2. We assume that the communication between sensors and the fusion center is multi-hop transmission. For each hop link, an ARQ protocol is employed. We select the Greedy Routing Scheme (GRS) [19] as our routing scheme because of its simplicity. In the GRS scheme, each node selects from among its neighbors the node closest to the final destination node as the next hop destination.

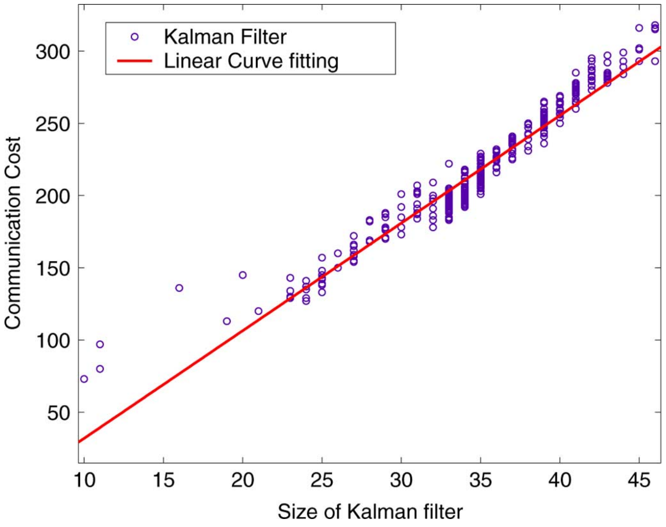

Fig. 9. Communication cost versus size of KBF: each circle represents a feasible KBF; the solid line is the linear curve fitting.

The results of the discrete event simulation are shown in Fig. 9. This figure plots by circle marks the different communication costs for all KBFs within the sensor communication range $R = 2$. We choose as transmission success probability $p = 0.95$, as transmission waiting time $T_w = 0.1$ (ms), and as transmission delay time T_d, where T_d is a random number drawn from a uniform density over the interval $(0, 1)$.

Fig. 9 shows that the communication cost, i.e., the total number of transmissions needed to send a packet from each of the sensors used by a given KBF to the fusion center, is approximately linearly proportional to the size of the KBF. Therefore, we define an approximate communication cost for a KBF that is based only on its size. We name it the *communication approximate cost*.

Definition 6 (Communication Approximate Cost): Assume that the size of the KBF K_f is n_f, i.e., the total number of sensors (inputs and outputs) used by K_f is n_f. Then, the communication approximate cost of K_f is defined as

$$\widetilde{\mathcal{C}}(K_f) = d_1 \cdot n_f + d_2 \tag{21}$$

where d_1 and d_2 are given constants. □

The values of d_1 and d_2 are determined by fitting the linear relation (21) to the available simulation/experimental data for a small subset of KBFs. Once the parameters d_1 and d_2 have been determined, $\widetilde{\mathcal{C}}(K_f)$ only depends on the size of K_f, which is found by inspection of the KBF. In Fig. 9, the solid line is the communication approximate cost derived by linearly curve fitting the data points; the parameter values are $d_1 = 7.451$ and $d_2 = -42.6$. These values were obtained by using the costs associated with 50 randomly chosen KBFs out of the 447 feasible KBFs. As we did for the approximate error cost, we repeated the estimation of these parameters with several sets of 50 different KBFs chosen randomly among the set of 447 feasible KBFs, obtaining each time a similar linear approximation as represented in Fig. 9. This numerical study supports the strategy of replacing the expensive-to-compute communication cost by the much simpler communication approximate cost when studying the tradeoffs between RMS estimation error performance and communication cost. This is what we will do in the next section, where we study the sensor network tradeoffs between estimation error performance and communication cost among all feasible KBFs. We use the communication approximate costs instead of the actual communication costs whenever these are not available or are only partly available from simulations or experiments.

VI. Pareto Optimality for the Estimation-Communication Tradeoff

Since, in general, there are multiple choices of cut-point sets, i.e., multiple choices of KBFs, for the real-time estimation of the field value at the desired locations, an important question is: *which KBF is the "best" one?* We address this issue for field estimation applications in WSN using the two cost functions introduced in Sections IV and V, namely: 1) accuracy of the estimation of the field and 2) cost of communication among sensors. Therefore, the selection of the appropriate cut-point set and the corresponding KBF is actually a problem of balancing the tradeoffs between these two competing objectives.

In this section we adopt Pareto optimality techniques [20] to evaluate and optimize this trade off. Pareto optimality strikes a balance between competing objectives in multi-criteria optimization problems. Most real-world problems have several objective functions that are to be simultaneously minimized with respect to the decision variables. A solution is said to be *Pareto optimal* if there exist no feasible solution that would improve one of the objective functions without causing a simultaneous deterioration in at least one other objective function. In most Pareto optimality problems, there is more than one Pareto optimal solution; the set of all feasible Pareto optimal solutions is called the *Pareto-optimal set*. The Pareto optimal feasible solutions are referred to as *nondominated* solutions. The other possible feasible solutions not on the Pareto optimal set are referred to as *dominated* solutions. The *Pareto front* is the plot of the nondominated solutions in the multidimensional cost space [20].

To apply this framework to the field estimation sensor network problem, the decision variables correspond to variables chosen to define KBFs, each arising from a feasible cut-point set. Our two competing objectives are the estimation performance measured by the RMS estimation error $\mathcal{E}$ associated with the KBF, and the communication cost $\mathcal{C}$ associated with transmitting the data to a fusion center to implement the KBF.

To find a particular solution on the Pareto front, we propose the following scalar cost function $J(\mathcal{E},\mathcal{C})$ that is a linear combination of the RMS estimation error $\mathcal{E}$ and the communication cost $\mathcal{C}$.

Definition 7 (RMS Error Versus Communication Cost Tradeoff): For a given KBF K_f, let $\mathcal{E}(K_f)$ be its RMS estimation error, and let $\mathcal{C}(K_f)$ be its communication cost. The cost function J for RMS error versus communication cost is defined as

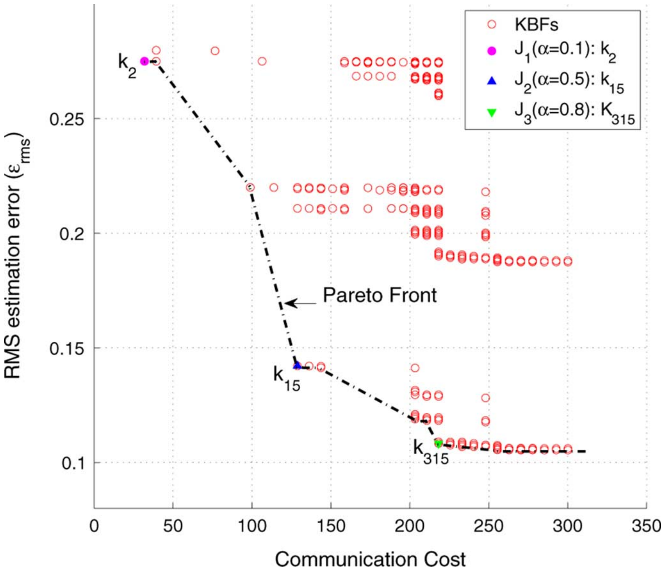

Fig. 10. Pareto optimality: Tradeoff between estimation and communication *using the approximate cost functions*. The circles are feasible KBFs; the dash-dot line is the Pareto front; the cost function $J(\mathcal{E},\mathcal{C};K_f)$ for different α values: J_1 for $\alpha = 0.1$, resulting in the KBF k_2; J_2 for $\alpha = 0.5$, resulting in the KBF k_{15}; J_3 for $\alpha = 0.8$, resulting in the KBF k_{315}.

$$J(\mathcal{E},\mathcal{C};K_f) = \alpha \cdot \left(\frac{\mathcal{E}(K_f)}{\bar{\mathcal{E}}}\right)^2 + (1-\alpha)\cdot\left(\frac{\mathcal{C}(K_f)}{\bar{\mathcal{C}}}\right)^2 \quad (22)$$

where $0 < \alpha < 1$ is a constant, and $\bar{\mathcal{E}}$ and $\bar{\mathcal{C}}$ are constants used for normalization purposes. □

The constant α, referred to as the regularization parameter, determines the relative emphasis placed on each term of the cost function: $\alpha = 0$ makes the communication cost the sole concern; $\alpha = 1$ places the entire design burden on the field estimation errors.

To calculate the cost function $J(\mathcal{E},\mathcal{C};K_f)$ for each KBF, we need to know both the RMS estimation error $\mathcal{E}$ and the communication cost $\mathcal{C}$ for the KBF. In Sections IV and V, we discussed how to obtain these values from experimental/simulation results. When the number of feasible KBFs is large, it is not feasible to obtain experimental/simulation KBF results for all possible cut-point sets. In these cases, we use the surrogate cost metrics we introduced in Definitions 4 and 6, namely, the RMS estimation error approximate cost $\widetilde{\mathcal{E}}(K_f)$ and the communication approximate cost $\widetilde{\mathcal{C}}(K_f)$. The resulting cost function $J(\widetilde{\mathcal{E}},\widetilde{\mathcal{C}};K_f)$ can be regarded as an approximate alternative to the original cost function $J(\mathcal{E},\mathcal{C};K_f)$.

Example 4: To illustrate the Pareto optimality approach to choose the best KBFs, we consider the set of KBFs from the previous examples. We denote the 447 KBFs by $\{k_1, k_2, \ldots, k_{447}\}$. To find Pareto optimal solutions, we start by populating the Pareto plane with many points representing feasible KBFs using the approximate cost functions and then draw the Pareto front. Fig. 10 shows the Pareto front and cost function $J(\mathcal{E},\mathcal{C};K_f)$

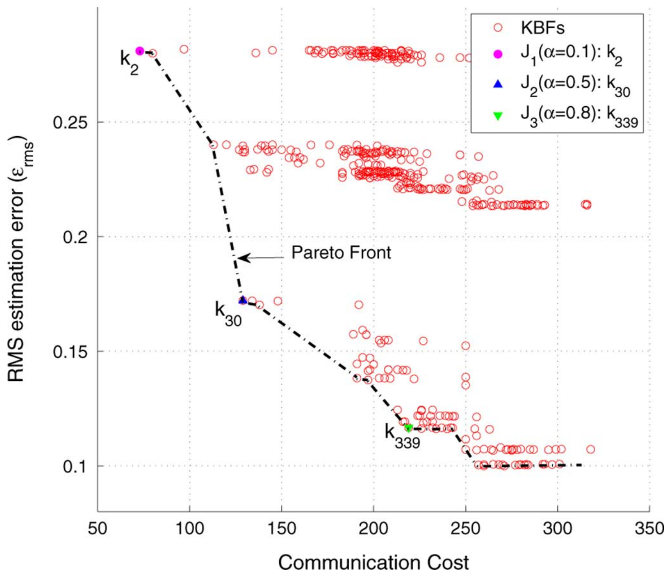

Fig. 11. Pareto optimality: Tradeoff between estimation and communication *using the exact cost computations*. The circles are feasible KBFs; the dash-dot line is the Pareto front; the cost function $J(\mathcal{E}, \mathcal{C}; K_f)$ for different α values: J_1 for $\alpha = 0.1$, resulting in the KBF k_2; J_2 for $\alpha = 0.5$, resulting in the KBF k_{30}; J_3 for $\alpha = 0.8$, resulting in the KBF k_{339}.

(22) for different α values. In Fig. 10, each circle represents a feasible KBF with its communication approximate cost and RMS estimation error approximate cost as its coordinates. The Pareto front, denoted by the dash-dot line in Fig. 10, is not smooth because the KBFs are discrete points in the space. The minimum values of the cost function $J(\mathcal{E}, \mathcal{C}; K_f)$ were computed for three values of α, 0.1, 0.5, and 0.8, by simply computing the weighted sum for each point in the figure for each value and sorting the results to find the minimum. The resulting costs are given by $J_i, i = 1, 2, 3$, in the figure caption, and the points with these cost values are labeled in Fig. 10 as k_2, k_{15}, and k_{315}, respectively.

For comparison, Fig. 11 shows the results using the exact computations of the RMS estimation errors and communication costs. We note that, although the numbers differ slightly, the general distributions of points in Figs. 10 and 11 are very similar. Moreover, the KBFs selected by each approach are the same, or very close, as shown in Fig. 12.

The difference in computation time is significant. Using the approximate cost functions required only 140 ms, where as the computations using the exact cost functions took 130 s, three orders of magnitude more time than using the approximate cost. The absolute times for this example, which involves only 53 sensors and limited data (2000 readings from each sensor), are not the issue here, since in real applications with a very large WSN the computation time for using the exact cost functions will be much larger, most likely prohibitive. □

The above example illustrates how to use Pareto optimality to trade off estimation error performance and communication cost. In summary, implementing the proposed Pareto optimal approach involves the following steps: 1) obtain the actual estimation errors and communication costs for a small subset of the feasible KBFs by experimentation or simulation; 2) compute the approximate alternatives to the estimation errors and communication for the remaining KBFs; 3) populate the Pareto plane with the data points representing all feasible KBFs; 4) choose a KBF that minimizes the cost function $J(\mathcal{E}, \mathcal{C}; K_f)$ (22) for a value of α that reflects the desired relative emphasis on estimation accuracy versus communication cost.

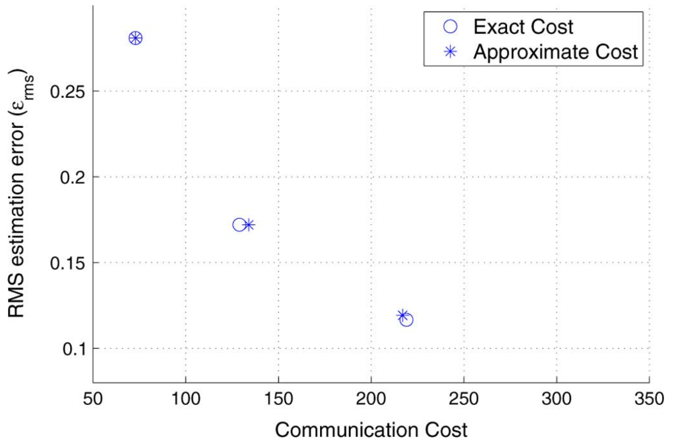

Fig. 12. Compare of Pareto optimality results between using the exact cost computations and using the approximate cost functions. The coordinates are the exact computation of the communication costs and RMS estimation errors. The circles are the Pareto optimality results by using the exact cost computations; the stars are the Pareto optimality results by using the approximate cost functions.

Remark: Although we motivated our approach with examples, our methodology is not limited to the examples presented. The approach is general and applies to the class of models that we consider (physical fields arising from partial differential equations, or Markov fields). In particular, the steps in the procedure, summarized in the previous paragraph, can be applied to other problems for which the assumed LTI model is appropriate. Also, as we point out in Sections IV and V, Monte Carlo simulations were used for our case study to evaluate RMS estimation error and the communication cost, and several cases were run to validate the approximation functions in Figs. 8 and 9.

This method can be easily extended to balance other goals in other WSN applications. The only difference is finding the appropriate approximate alternatives for the different objective functions used in this WSN application.

VII. Conclusion

Operation of a wireless sensor network (WSN) involves tradeoffs due to the limited resources of sensors and their potentially large number. This paper proposes an approach to evaluate the network tradeoffs between field estimation error and communication cost in a WSN. We present a graph-based approach to field estimation in WSNs proposed in [5]. We exploit the field correlation as captured by an underlying physical model. To estimate the field values at spatial locations where there are no sensors, we apply Kalman–Bucy filtering (KBF) to deal with sensor measurement noise. Rather than using a single global KBF, we derive a reduced-order model of much lower dimension to estimate the field at specified locations. Each reduced-order model is associated with a cut-point set P_c that encircles the desired location. There are many possible cut-point sets, so there are many feasible KBFs. The design issue is now which of these KBFs should be adopted to estimate the field at the desired location. Different KBFs estimate the field with different estimation errors and incur different communication costs. Computing these two metrics $\mathcal{E}$ and $\mathcal{C}$ is costly. We introduce approximate alternatives to compute the estimation error and communication cost in order to dramatically reduce the burden of their exact computation. Finally, we propose a Pareto optimality scheme to balance between the two contradictory goals: minimum estimation error $\mathcal{E}$ and minimum communication cost $\mathcal{C}$. We use simulation results to explain and verify our approach to selecting the best KBF by Pareto optimality.

In this paper, we give a generalized framework for real-time field estimation of correlated distributed fields in WSNs. This method is useful in many WSN applications such as building climate control and environmental monitoring. The Pareto optimality scheme we use to trade off between estimation errors and communication costs can also be used to balance other goals in WSN applications.

Directions to extend the work in this paper include the following.

1) *Distributed KBFs for field estimation*. In this paper, we assume that for each KBF there exists a single fusion center. With the improvement of the processing capability of sensor nodes, distributed KBFs may be appropriate for many WSN applications. The problem of distributed KBFs has been studied in several articles [8], [16], [17]. References [16] and [17] assume that each node has observations of the state of the physical process, which is not true in our case where each sensor has only observations of a subset of the state variables. Reference [8] derives a completely distributed version of the KBF by applying the cut-set approach presented in Section III. It will be important to study in the context of the distributed solution in [8] the tradeoffs between estimation errors and communication costs.

2) *Grouping method for field estimation of multiple locations*. If there are multiple distributed field locations to be estimated, we have two choices to implement real-time estimation: establish a KBF to estimate collectively the field at all desired locations; or divide the desired locations into several subgroups, and establish a KBF for each of the subgroups. When the desired locations are sparsely distributed, considering both estimation error performance and communication cost, it will be important to determine which approach is better. If the second choice is selected, the question is then to determine the optimal grouping method for the desired locations. In this case, the tradeoff between estimation error performance and communication cost becomes more complicated.

References

[1] A. T. P. So, "Building control and automation systems," in *Perspectives in Control Engineering Technologies, Applications, and New Directions*, T. Samad, Ed. New York: Wiley-IEEE Press, 2001, pp. 393–416.

[2] C. D. Patel, "A vision of energy aware computing from chips to data centers," presented at the Int. Symp. Micro-Mechanical Enigneering (ISMME), Tsuchiura and Tsukuba, Japan, Dec. 1–3, 2003.

[3] R. Cardell-Oliver, K. Smettem, M. Kranz, and K. Mayer, "A reactive soil moisture sensor network: Design and field evaluation," *Int. J. Distribut. Sensor Netw.*, vol. 1, no. 2, pp. 149–162, Apr.–June 2005.

[4] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," presented at the 1st ACM Int. Workshop Wireless Sensor Networks Applications (WSNA 2002), Atlanta, GA, Sep. 28, 2002.

[5] H. Zhang, J. M. F. Moura, and B. Krogh, "Estimation in sensor networks: A graph approach," in *4th Int. Symp. Information Processing Sensor Networks (IPSN 2005)*, Los Angeles, CA, Apr. 25–27, 2005, pp. 203–209.

[6] A. Asif and J. M. F. Moura, "Data assimilation in large time varying multidimensional fields," *IEEE Trans. Image Process.*, vol. 8, no. 11, pp. 1593–1607, Nov. 1999.

[7] A. Asif and J. M. F. Moura, "Inversion of block matrices with L-block banded inverse," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 630–642, Feb. 2005.

[8] U. A. Khan and J. M. F. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4919–4935, Oct. 2008, Part 1.

[9] G. F. Franklin, J. D. Powell, and M. L. Workman*, Digital Control of Dynamic Systems*. Reading, MA: Addison-Wesley, 1990.

[10] D. D. Šiljak*, Decentralized Control of Complex Systems*. New York: Academic, 1991, pp. 2–6.

[11] H. Zhang, "Field estimation in wireless sensor networks," Ph.D. dissertation, Electr. Comput. Eng. Dept., Carnegie Mellon Univ., Pittsburgh, PA, Dec. 2005.

[12] E. S. Sousa, "The performance of a spread spectrum link in a Poisson field of interferers," in *Proc. IEEE Military Communication Conf. (MILCOM 1989)*, Boston, MA, Oct. 15–18, 1989, vol. 1, pp. 149–153.

[13] J. Ilow and D. Hatzinakos, "Analytic alpha-stable noise modeling in a Poisson field of interferers or scatterers," *IEEE Trans. Compon. Packag. Technol.*, vol. 46, no. 6, pp. 1601–1611, Jun. 1998.

[14] V. Matta, S. Marano, P. Willett, and L. Tong, "Dumb isotropic sensors can find DOAs," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP 2005)*, Philadelphia, PA, Mar. 19–23, 2005, pp. IV 761–IV 764.

[15] L. Kleinrock and J. Silvester, "Spatial reuse in multihop packet radio networks," in *Proc. IEEE*, Jan. 1987, vol. 75, pp. 156–167.

[16] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Distributed Kalman filtering in sensor networks with quantifiable performance," presented at the 4th Int. Symp. Information Processing Sensor Networks (IPSN 2005), Los Angeles, CA, Apr. 25–27, 2005.

[17] R. B. Quirino and C. P. Bottura, "An approach for distributed Kalman filtering," *SBA Control Automatica*, vol. 12, no. 1, pp. 19–28, Jan./Apr. 2001.

[18] E. E. Johnson, "ARQ protocol features for HF radio links," in *Proc. IEEE Military Communication Conf. (MILCOM 2002)*, Anaheim, CA, Oct. 7–10, 2002, vol. 1, pp. 125–130.

[19] G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," ISI, Marina del Ray, CA, ISI Res. Rep. ISU/RR-87-180, Mar. 1987.

[20] Y. Sawaragi, H. Nakayama, and T. Tanino*, Theory of Multi-Objective Optimization*. New York: Academic, 1985.

Haotian Zhang (S'01–M'05) received the B.S. and M.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1997 and 2000, respectively, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2005.

He has worked for Maxtor Corporation, Shrewsbury, MA, and Marvell Semiconductor, Inc., Longmont, CO, and currently works for LSI Corporation, Longmont, CO.

José M. F. Moura (S'71–M'75–SM'90–F'94) received the Engenheiro Electrotécnico degree from the Instituto Superior Técnico (IST), Lisbon, Portugal, and the M.Sc., E.E., and D.Sc. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge.

He is a Professor of electrical and computer engineering and, by courtesy, of biomedical engineering, at Carnegie Mellon University (CMU), Pittsburgh, PA. He was on the faculty at IST, has held visiting faculty appointments at MIT, and was a visiting research scholar at the University of Southern California, Los Angeles. He is a founding co-director of the Center for Sensed Critical Infrastructures Research (CenSCIR) and Director of the Information and Communication Technologies Institute (ICTI) at CMU that manages a large education and research program between CMU and Portugal (www.cmu.edu/portugal). His research interests include statistical and algebraic signal processing, image, bioimaging, and video processing, and digital communications. He has published over 330 technical journal and conference papers, is the co-editor of two books, holds seven patents, and has given numerous invited seminars at international conferences, U.S. and European universities, and industrial and government laboratories.

Dr. Moura is the President (2008–2009) of the IEEE Signal Processing Society (SPS). He was Editor-in-Chief for the IEEE TRANSACTIONS IN SIGNAL PROCESSING, interim Editor-in-Chief for the IEEE SIGNAL PROCESSING LETTERS, and was on the Editorial Board of several journals, including the *IEEE Proceedings*, the *IEEE Signal Processing Magazine*, and the ACM *Transactions on Sensor Networks*. He was on the steering and technical committees of numerous conferences and is or was a member of several IEEE Boards. He is a Fellow of the American Association for the Advancement of Science (AAAS), and a corresponding member of the Academy of Sciences of Portugal (Section of Sciences). He was awarded the 2003 IEEE Signal Processing Society Meritorious Service Award and in 2000 the IEEE Millennium Medal. In 2007, he received the CMU's College of Engineering Outstanding Research Award. He is affiliated with several IEEE societies, Sigma Xi, AMS, IMS, and SIAM.

Bruce Krogh (S'82–M'82–SM'92–F'98) is Professor of electrical and computer engineering at Carnegie Mellon University, Pittsburgh, PA. His current research interests include synthesis and verification of embedded control software, distributed sensing and control strategies using wireless networks, and discrete event and hybrid dynamic systems.

Prof. Krogh was a past Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL AND DISCRETE EVENT DYNAMIC SYSTEMS: THEORY AND APPLICATIONS and founding Editor-in-Chief of the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY. He is a Distinguished Member of the IEEE Control Systems Society.