IPSN'05 The Fourth International Symposium on Information Processing in Sensor Networks, Pages: 203-209, UCLA, Los Angeles CA, April 25-27, 2005.

Estimation in Sensor Networks: A Graph Approach

Haotian Zhang, José M. F. Moura, and Bruce Krogh
Carnegie Mellon University
Electrical and Computer Engineering
Pittsburgh, PA 15213, USA
{haotian,moura,krogh}@ece.cmu.edu

***Abstract*—In many sensor networks applications, sensors collect correlated measurements of a physical field, e.g., temperature field in a building or in a data center. However, the locations of the sensors are usually inconsistent with the application requirements. In this paper, we consider the problem of estimating the field at arbitrary positions of interest, where there are possibly no sensors, from the irregularly placed sensors. We map this sensor network on a graph, and, by introducing the concepts of interconnection matrices, system digraphs, and cut point sets, we can pose sensor network tradeoffs and derive real-time field estimation algorithms. The results of temperature field estimation, obtained from simulations and real world experiments, show that the methodology presented in this paper can successfully predict the field values at arbitrary locations, including others than the ones with sensors.**

I. Introduction

Sensor networks are often used in the applications where field values governed by continuous, distributed dynamics need to be monitored and controlled. e.g., temperature monitoring and controlling in buildings to provide comfort for occupants [1] or to minimize the cost of cooling in data centers with large numbers of computers [2]. Spatial and temporal irregularities in the physical data and the physical device array are usually inconsistent with the application requirements. For example, the sensors are placed at irregular positions, while the application usually requires tracking the field values at other locations.

In this paper, we consider the problem of estimating values at points other than the sensor locations in correlated dynamic fields, whose underlying physics can be modeled by lumped-parameter models. To be able to do this, we assume that the field is spatially and temporally correlated. A second important feature of our work is that we assume that the field of interest is well described by a partial difference equation (PDE) that captures the underlying physics. This is an appropriate model, for example, to describe heat transfers in a building or temperature distributions on a micro-electronic chip, since temperature dynamics are governed by partial differential equations (PDEs) that characterize the thermal effects of convection, conduction, and radiation [3], [4]. We do not directly consider the PDE model, but rather lumped parameter dynamic models derived from it, such as RC networks used in heat transfer studies [5], [6]. These lumped parameter dynamic models can be derived from discretizations of the PDE model under certain operating regimes.

II. Problem Statement

With reference to Figure 1, we consider the problem of estimating the values of the physical field in a region at specific locations of interest $R = \{r^1, \ldots, r^n\}$ based on measurements at sensor locations $S = \{s^1, \ldots, s^m\}$. The locations of interest R will be in general different from the sensor locations S. Furthermore, we want to estimate these field values based on a subset S' of the sensor measurements, where $S' \subset S$. For simplicity, we assume initially that the sensor measurements are synchronous and periodic in time, occurring at $t_0, t_1, \ldots$ with sampling period $t_{k+1} - t_k = T_s$. This means that we ignore the temporal irregularities and focus on dealing with the spatial irregularities.

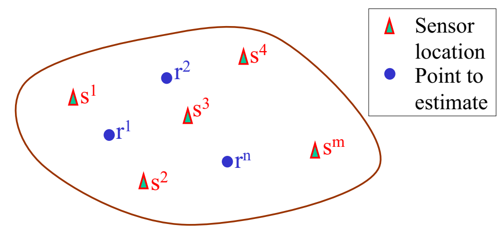

Fig. 1. Estimation in correlated distributed field.

We assume that the evolution of the field can be modeled at various levels of complexity. In these different models, we refer to the various unknown constants as *parameters*, denoted by a vector θ. The most detailed first-principles (fp) model of the distributed system dynamics is denoted by M_{fp}. We approximate M_{fp} by various finite-dimensional (lumped-parameter) models. For the class of problems considered in this paper, we use a detailed lumped-parameter model M_{de} that captures the relationship between the measured field values at locations S and the field values at the locations of interest, that is, at locations R. We then formulate appropriate estimation problems for the finite dimensional lumped-parameter model M_{de}.

We assume that the physical field is modeled by a detailed lumped-parameter model M_{de} described by[1]:

$$\dot{\mathbf{x}} = A(\theta)\mathbf{x} + B(\theta)\mathbf{u}, \qquad \mathbf{y} = C\mathbf{x} + \mathbf{w} \tag{1}$$

where: $\mathbf{x} \in R^p$ is the state; $\mathbf{u} \in R^q$ is the input; the components of the output $\mathbf{y} \in R^m$ correspond to available measurements in the sensor network application; $\mathbf{w}$ is the measurement noise; $\theta \in R^k$ is the vector of unknown parameters; and A, B, C are matrices of dimensions $p \times p$, $p \times q$, and $m \times p$, respectively. The state $\mathbf{x}$ collects the field values at all locations discretizing the space of interest; these include the sensors S and the locations of interest R. In general, however, the state collects many more locations than just these, as required by the discretization of the partial differential (or difference) equation describing the phenomena of interest; in other words, the dimensionality of the state is determined by a Nyquist

Research supported by NSF grant # ECS-0225449.

[1] We express the time evolution of the field by differential rather than difference equations. This is for notational convenience—it provides for more compact notation. In practice, we work with difference equations and discrete time.

sampling rate type argument. The input $\mathbf{u}$ represents, for example, in the temperature field application, the heat exchanges with the external world, usually unknown, unless sensors are at strategic locations.

The field described by (1) is correlated. It is well known from stochastic processes and linear systems that its correlation follows the Laypounov equation

$$\frac{d\Sigma_{\mathbf{x},t}}{dt} = A(\theta)\Sigma_{\mathbf{x},t} + \Sigma_{\mathbf{x},t}A(\theta)^T + B(\theta)QB(\theta)^T, \qquad (2)$$

where Q is the covariance of $\mathbf{u}$.

To illustrate the field equations, we illustrate the measurement equation for the simple case where the first m entries in $\mathbf{x}$ are the sensor measurements and the last n entries in $\mathbf{x}$ are the field at the locations of interest. The vector $\mathbf{y}$ of sensor measurements corresponds to a subset of the state variables. Then, C is given by

$$C = \left[\begin{array}{cc} I_m & 0_{m\times(p-m)} \end{array} \right], \qquad (3)$$

where I_m is the $m \times m$ identity matrix, and $0_{m\times(p-m)}$ is a zero matrix of dimension $m \times (p-m)$. More complex sensor network patterns can be similarly represented.

Problem: Field estimation in sensor networks Our goal is to derive an algorithm to estimate in real-time the field variables $\mathbf{f} \in R^n$ at the locations of interest R from the sensor network measurements available:

$$\mathbf{f} = F\mathbf{x}, \qquad F = \left[\begin{array}{cc} 0_{n\times(p-n)} & I_n \end{array} \right] \qquad (4)$$

where F is an $n \times p$ matrix. The estimation of $\mathbf{f}$ is to be based on the available sensor measurements $\mathbf{y}$. In sensor networks, due to the constraints on communication and computation, we cannot broadcast *all* the sensor measurements $\mathbf{y}$ to implement the real-time estimation of the field at the locations of interest R. Rather, in estimating the field at a *given* location in R, we want to determine which ones are the relevant sensors and use their measurements and only theirs in the field estimation at that location of interest. In other words, our task is to partition the sensor network measurements and the system equations (1) in such a way that the field at each location of interest is estimated from a selected subset of sensor measurements $\mathbf{y_s} \in R^l$, where $l \ll m$. To achieve this, we first exploit the field correlation structure to map on graphical models this sensor network estimation problem. We then develop graph network algorithms to determine the set of relevant measurements and to partition the state equation (1) to achieve the field estimation at each location of interest R using only local measurements.

III. Approach Based on Graph Theory

In this section, we rework the sensor network field estimation problem as an estimation problem on graphs. We introduce three needed concepts: interconnection matrix, system digraph, and cut point set. We then derive a theorem and present an algorithm that partitions the system digraph and allows us to replace the estimation of the field from global measurements by estimation of the field using only local (i.e., neighboring) measurements. Finally, we illustrate the overall approach in detail with real sensor network measurements collected with Crossbow temperature sensors.

A. Concepts

The relationship among the input, the state, and the output of a linear dynamical system is well represented by *interconnection matrices* and *system digraphs*, [7]. When the output is a subset of the state variables, as it is usually the case in sensor networks, the relationship between the state and the output is trivial; we need only to consider the relationship between the input and the state. Because of this, we modify the concepts of *interconnection matrices* and *system digraphs* as follows.

*Definition 1 (**Interconnection Matrix**): For a linear system* $\mathbf{S}$ *with system equation:*

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \qquad (5)$$

where A and B are matrices of dimensions $p \times p$ and $p \times q$, respectively, the interconnection matrix is the binary $p \times (p+q)$ matrix $E = (e_{ij})$ defined as

$$E = \left[\begin{array}{cc} \bar{A} & \bar{B} \end{array} \right], \qquad (6)$$

where the matrices $\bar{A} = (\bar{a}_{ij})$ and $\bar{B} = (\bar{b}_{ij})$ are

$$\bar{a}_{ij} = \begin{cases} 1, & a_{ij} \neq 0, \\ 0, & a_{ij} = 0, \end{cases} \qquad \bar{b}_{ij} = \begin{cases} 1, & b_{ij} \neq 0, \\ 0, & b_{ij} = 0. \end{cases} \qquad (7)$$

The matrices $\bar{A}$ and $\bar{B}$ are Boolean representations of the original system matrices A and B. By converting matrix elements to binary values, the interconnection matrix E captures the structural properties of the dynamical system, without much concern for the specific numerical values of the system parameters, [7].

To interpret the system $\mathbf{S}$ structural properties, we introduce *digraphs* (directed graphs). A digraph is, [8], an ordered pair $\mathbf{D} = (V, L)$, where V is a nonempty finite set of *vertices* (points, nodes), and L is a relation in V, that is, L is a set of ordered pairs (v_j, v_i), which are the directed *edges* (lines, arcs) connecting the vertices of $\mathbf{D}$. Note that, in digraphs, the direction of an edge is important.

We now consider the *system digraph* for the system $\mathbf{S}$ with interconnection matrix $E = (e_{ij})$. We modify the concept of system digraphs in [7] as follows.

*Definition 2 (**System Digraph**): The system digraph $\mathbf{D} = (\mathrm{V}, \mathrm{L})$ of a system $\mathbf{S}$ with interconnection matrix $E = (e_{ij})$ has: (i) the vertex set $V = U \cup X$, where $U = \{u_1, u_2, \ldots, u_q\}$ and $X = \{x_1, x_2, \ldots, x_p\}$ are nonempty sets of input and state vertices of $\mathbf{D}$, respectively; and (ii) the edge set L, where the directed edge $(v_j, v_i) \in \mathrm{L}$ if and only if $e_{ij} = 1$. The vertex set V includes measurable vertices and unmeasurable vertices that are represented by two different marks, respectively, in the system digraph.*

As we commented below (1), the state and input of the system $\mathbf{S}$ describing the field may include *measurable inputs* and *unmeasurable inputs*, and *measurable states* and *unmeasurable states*: the measurable inputs and the measurable states correspond to inputs and states measured by the network sensors. Appropriately, we partition the input and state vertex sets U and X in the system digraph as follows: (i) the input vertex set $U = U_m \cup U_n$, where $U_m = \{u_{m_1}, \ldots, u_{m_i}\}$ is the vertex set of measurable inputs, and $U_n = \{u_{n_1}, \ldots, u_{n_j}\}$ is the vertex set of unmeasurable inputs; and (ii) the state vertex set $X = X_m \cup X_n$, where $X_m = \{x_{m_1}, \ldots, x_{m_i}\}$ is the vertex set of measurable states, and $X_n = \{x_{n_1}, \ldots, x_{n_j}\}$ is the vertex set of unmeasurable states. This grouping of the input vertex set and of the state vertex set partitions the digraph vertex set as $V = V_m \cup V_n$, where $V_m = U_m \cup X_m$ is the *measurable vertex set* and $V_n = U_n \cup X_n$ is the *unmeasurable vertex set*. In a system digraph, we represent measurable and unmeasurable vertices by different marks, e.g., by circles and squares, respectively.

We illustrate the concept of interconnection matrices and structure digraphs when the system matrices in (1) are:

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

$$
B(\theta)^T = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & b_{81} & 0 \\
0 & 0 & b_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & b_{43} & 0 & 0 & 0 & 0 & 0
\end{bmatrix}, \tag{8}
$$

where the nonzero elements in $A(\theta)$ and $B(\theta)$ may be constants or functions of unknown parameters θ. The state vector $\mathbf{x} = [x_{m1}, \ldots, x_{m6}, x_{n1}, x_{n2}, x_{n3}]^T$, and the input vector $\mathbf{u} = [u_{m1}, u_{n1}, u_{n2}]^T$. Then the interconnection matrix E of the system $\mathbf{S}$ can be derived as:

$$
E = \begin{bmatrix} \bar{A} & \bar{B} \\ 0 & 0 \end{bmatrix}
$$

$$
= \begin{array}{c} \\ x_{m1} \\ x_{m2} \\ x_{m3} \\ x_{m4} \\ x_{m5} \\ x_{m6} \\ x_{n1} \\ x_{n2} \\ x_{n3} \\ u_{m1} \\ u_{n1} \\ u_{n2} \end{array}
\begin{array}{c}
\begin{array}{cccccccccccc} x_{m1} & x_{m2} & x_{m3} & x_{m4} & x_{m5} & x_{m6} & x_{n1} & x_{n2} & x_{n3} & u_{m1} & u_{n1} & u_{n2} \end{array} \\
\left[\begin{array}{ccccccccc|ccc}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
\end{array} \tag{9}
$$

According to Definition 2, the corresponding structure digraph of the system $\mathbf{S}$ is shown in Figure 2 where circle nodes represent measurable vertices, and square nodes represent unmeasurable vertices. A directed edge (v_i, v_j) shows the vertex v_j's dependence on v_i. For example, the edge (u_{n_1}, x_{m_3}) shows that the update of the state x_{m_3} (partially) depends on the value of the input u_{n_1}.

Now we introduce the concept of *cut point set* in the context of system digraphs. Assume $\mathbf{D} = (V, L)$ is a system digraph, where $V = V_m \cup V_n$, $V_m = U_m \cup X_m$, and $V_n = U_n \cup X_n$, as defined previously. Let X_s be a subset of state vertices (measurable and/or unmeasurable), i.e., $X_s \subseteq X$. Usually, the unmeasurable state vertices in X_s correspond to the field values at locations of interest. The cut point set P_c for the state vertex set X_s is defined as follows.

*Definition 3 (**Cut Point Set**): In the system digraph $\mathbf{D} = (V, L)$, the cut point set P_c for the state vertex set X_s is a set of measurable vertices, i.e., $P_c \subseteq V_m$, for which we can find an extended state vertex set $X'_s \supseteq X_s$ such that:*
i) For an arbitrary vertex $v_i \in P_c$, there exist at least one edge (v_i, v_j), where $v_j \in X'_s$;
ii) There does not exist any edge (v_i, v_j), where $v_i \in V \cap \bar{P}_c \cap \bar{X}'_s$ and $v_j \in X'_s$.

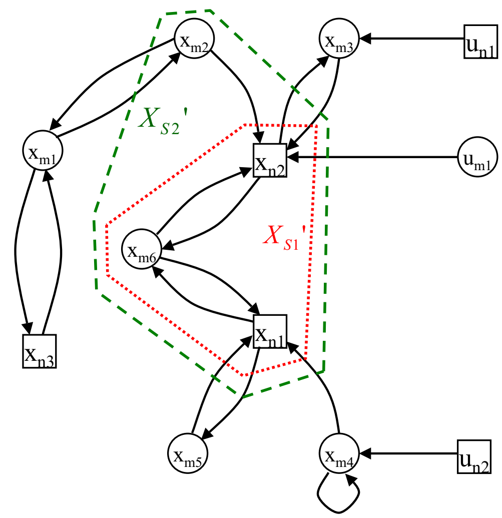

Fig. 2. Example of system digraphs.

In the above definition, condition (i) shows that each vertex in the cut point set P_c has a direct impact on at least one vertex in the extended vertex set X'_s; condition (ii) shows that the vertices in X'_s only depend on the vertices in P_c and X'_s, which implies that the vertices in X'_s have no relationship with the vertices not in P_c and X'_s. Note that the vertices in the cut point set P_c are measurable, i.e., their values are known, and P_c can contain both state vertices and input vertices. In other words, $P_c \subseteq V_m$. Then all the vertices in X'_s depend only on themselves and on the known vertices in P_c.

The following issues need to be addressed:

1) The cut point set for X_s may not be unique since we can add different vertices into X_s to form X'_s.
2) As a result of item 1, when there is more than one cut point set for X_s, we can define a *maximal/minimal cut point set* whose corresponding extended state vertex set X'_s has the maximal/minimal cardinality. Note that the concept of the maximal/minimal cut point set is defined on the size of X'_s, rather than the size of the cut point set itself.

Cut point sets are key in our approach. For a given vertex set X_c, which usually includes all the states to estimate, its cut point set P_c is the set of measurable vertices that impact the states in X_s, i.e., the states in X_s only depend on the vertices in P_c; other vertices (sensors) not in P_c can be ignored. We introduce this concept so that for a given task we use only the measurements from the subset of sensors that are strictly necessary, thus reducing the communication burden and the computational load required for estimating the field in the vertices in X_s.

When there are multiple states to estimate, we have two choices: we either find a single "large" cut point set for all the states of interest; or, we find "small" cut point sets, one for each state of interest. The latter approach distributes the estimation, since each state is estimated based on the information from neighboring sensors, and the estimation can be implemented locally.

We explain cut point sets with reference to the system digraph in Figure 2. 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 vertex set $X'_{s1} = \{x_{n_1}, x_{n_2}, x_{m_6}\}$ encircled by the dotted line in Figure 2. 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 vertex set $X'_{s1} = \{x_{n_1}, x_{n_2}, x_{m_2}, x_{m_6}\}$ encircled by the dashed line in Figure 2. Since P_{c1} and P_{c2} are the only two cut point sets for X_s and $X'_{s1} \subset X'_{s2}$, P_{c1} and P_{c2} are the minimal and maximal cut point sets for X_s.

We now present an algorithm for finding the minimal cut point set for a given state vertex set. Assume that a system digraph $\mathbf{D} = (V, L)$ corresponds to an interconnection matrix $E = \{e_{ij}\}$ of dimension $p \times (p+q)$, i.e., the state $\mathbf{x} \in R^p$ and the input $\mathbf{u} \in R^q$. Let the state vertex set $X = \{x_1, \ldots, x_p\}$ and the input vertex set $U = \{u_1, \ldots, u_q\}$. The universal vertex set $V = X \cup U$. The task is to find the *minimal cut point set* P_{min} for the given state vertex set $X_s = \{x_{s_1}, \ldots, x_{s_m}\}$. Let $\emptyset$ be the empty set. This is accomplished by the following algorithm.

*Algorithm 1 (**Searching for the minimal cut point set**):*

1. **Initialization** *Let $P_{min} = \emptyset$, $X'_s = X_s$, $X_t = \emptyset$, and $X_{st} = X_s$.*
2. *Let P_t be the set of all the vertices in $\bar{X}'_s$ that are associated with at least one directed edge pointing to the vertices in X_{st}. If $P_t = \emptyset$, go to Step 6.*
3. *For each vertex in P_t: if it is a measurable (state or input) vertex, add it to P_{min}; if it is an unmeasurable state vertex, add it to X'_s and X_t; if it is an unmeasurable input vertex, go to Step 6.*
4. *If $X_t = \emptyset$, go to the Step 5; otherwise, let $X_t = \emptyset$, $X_{st} = X_t$, and go back to Step 2.*
5. *End. P_{min} is the minimal cut point set for the given state vertex set X_s, and X'_s is the extended state vertex set.*
6. *End. There is no cut point set for X_s.*

We next prove that the resulting cut point set P_{min} is the minimal cut point set for X_s.

Proof: Let X_s be the given state vertex set, P_{min} the resulting cut point set from Algorithm 1, and X'_s the corresponding extended state vertex set. According to Algorithm 1, only unmeasurable state vertices could be added into X'_s. Therefore, $\tilde{X}'_s = X'_s \cap \bar{X}_s$ is a set of unmeasurable state vertices, and the vertices in X_s depend on the vertices in $\tilde{X}'_s$. For each vertex in $\tilde{X}'_s$, there exists at least one path starting from this vertex to vertices in X_s, and all the vertices included in these paths (except the ending vertices) are unmeasurable state vertices.

We prove by contradiction that P_{min} is the minimal cut point set for X_s. Assume that there exists a cut point set smaller than P_{min}, denoted by P_{mm}, and its corresponding extended state vertex set is $X_{\text{mm}} \subset X'_s$. Since the vertices in X_s only depend on the vertices in P_{mm}, and P_{mm} is a set of measurable vertices, $\tilde{X}'_s$ must be a subset of X_{mm}, i.e., $\tilde{X}'_s \subset X_{\text{mm}}$. In other words, the vertices in $\tilde{X}'_s$ must be included in the paths starting from the vertices in P_{mm} to the vertices in X_s. Since $X_s \subset X_{\text{mm}}$ and $\tilde{X}'_s \subset X_{\text{mm}}$, we know $X_s \cup \tilde{X}'_s \subseteq X_{\text{mm}}$, i.e., $X'_s \subseteq X_{\text{mm}}$. This conflicts with the assumption of $X_{\text{mm}} \subset X'_s$. Therefore, there does not exist any cut point set smaller than P_{min}, i.e., P_{min} is the minimal cut point set for X_s. □

Based on Algorithm 1, we can derive another algorithm to find all the cut point sets for X_s. Let P_{min} be the minimal cut point set for X_s with corresponding extended state vertex set X'_s.

*Algorithm 2 (**Searching for all cut point sets**):*

1. **Initialization** *Let $P_c = P_{min}$.*
2. *Represent P_c by $\{v_1, \ldots, v_k\}$. For $1 \leqslant i \leqslant k$, let $X_{si} = X'_s \cup \{v_i\}$, and search for the minimal cut point set P_{ci} for X_{si} by Algorithm 1: if P_{ci} exists, save it as a new cut point set for the original state vertex set X_s.*
3. *For each new cut point set P_{ci}, let $P_c = P_{ci}$ and repeat Step 2. Do this recursively, until we cannot find any new cut point set for X_s.*
4. *End.*

If we cannot find a new cut point set for X_s by Algorithm 2, the P_{min} is the only cut point set for X_s; otherwise, among those new cut point sets, the one whose corresponding extended state vertex set X_{si} has the largest cardinality is called the *maximal cut point set* P_{max}.

B. Theorem

With the concepts of interconnection matrix E, system digraph $\mathbf{D} = (V, L)$, and cut point sets, we introduce the main result of our approach given in the following theorem.

Theorem 1: In the system digraph $\mathbf{D} = (V, L)$ of a system $\mathbf{S}$, given a state vertex set X_s, assume that we find a cut point set P_c for X_s and the corresponding extended state vertex set X'_s. We then can establish a new system $\mathbf{S}'$, based on the system $\mathbf{S}$, whose inputs are the vertices in P_c, whose states are the vertices in X'_s, and whose system matrices A' and B' can be derived directly from the system matrices A and B of the original system $\mathbf{S}$. The states of the new system $\mathbf{S}'$ correspond to a subset of the states in $\mathbf{S}$, and have the exactly same time series values as those of the corresponding states in $\mathbf{S}$ under the same initial conditions and inputs.

Proof: For convenience, the vectors corresponding to the vertex sets U, X, X_s, X'_s, and P_c are represented in bold lowercase font as $\mathbf{u}$, $\mathbf{x}$, $\mathbf{x_s}$, $\mathbf{x'_s}$, and $\mathbf{p_c}$, respectively.

In the original system $\mathbf{S}$, let the state vector be $\mathbf{x} = [x_1 \ x_2 \ \ldots \ x_n]^T$, the input vector be $\mathbf{u} = [u_1 \ u_2 \ \ldots \ u_m]^T$, and the system matrices be $A = \{a_{ij}\}$ and $B = \{b_{ij}\}$ with dimensions $n \times n$ and $n \times m$, respectively. The state equation in matrix form corresponds to the group of linear equations:

$$\dot{x}_i = \sum_{j=1}^{n} a_{ij} x_j + \sum_{j=1}^{m} b_{ij} u_j, \qquad i = 1, 2, \ldots, n. \tag{10}$$

Assume that we find a cut point set, denoted by $\mathbf{p_c} = [x_{c_1} \ x_{c_2} \ \ldots \ x_{c_p} \ u_{c_{(p+1)}} \ u_{c_{(p+2)}} \ \ldots \ u_{c_q}]^T$, for a given state vertex set $\mathbf{x_s} = [x_{s_1} \ x_{s_2} \ \ldots \ x_{s_k}]^T$ and the corresponding extended vertex point set $\mathbf{x'_s} = [x_{s_1} \ x_{s_2} \ \ldots \ x_{s_l}]^T$, where $l \geq k$, $q \geq p$, $x_{s_i} \in \mathbf{x}$, $x_{c_i} \in \mathbf{x}$, and $u_{c_i} \in \mathbf{u}$. Note that the indices of elements in $\mathbf{p_c}$, $\mathbf{x_s}$, and $\mathbf{x'_s}$ have the same values as the indices of the corresponding elements in $\mathbf{x}$ and $\mathbf{u}$. For example, if x_{c_2} in X_C corresponds to x_6 in $\mathbf{x}$, then $c_2 = 6$.

According to the definition of the cut point set $\mathbf{p_c}$ and the resulting extended state vertex set $\mathbf{x'_s}$, if we only select from equation (10) the linear equations corresponding to the states in $\mathbf{x'_s}$ and ignore the zero items, we get a new group of linear equations:

$$\dot{x}_{s_i} = \sum_{j=1}^{l} a_{s_i s_j} x_{s_j} + \sum_{j=1}^{p} b_{s_i c_j} x_{c_j} + \sum_{j=p+1}^{q} b_{s_i c_j} u_{c_j}, \tag{11}$$

$i = 1, 2, \ldots, l$, i.e., the update of the states in $\mathbf{x'_s}$ only depends on the elements in $\mathbf{x'_s}$ and $\mathbf{p_c}$. If we write equation (11) in matrix form, we get the system equation of a new system $\mathbf{S}'$:

$$\dot{\mathbf{x}}' = A'\mathbf{x}' + B'\mathbf{u}', \tag{12}$$

where the input $\mathbf{u}' = \mathbf{p_c}$, the state $\mathbf{x}' = \mathbf{x'_s}$, and the system matrices $A' = \{a'_{ij}\}$ and $B' = \{b'_{ij}\}$ are given as:

$$a'_{ij} = a_{s_i s_j}, \text{ where } 1 \leq i, j \leq l,$$

and $$b'_{ij} = b_{s_i c_j}, \text{ where } 1 \leq i \leq l \text{ and } 1 \leq j \leq q.$$

For the new system $\mathbf{S}'$, since the linear equations in equation (12) are exactly the same as the corresponding linear equations in equation (10), and the elements of the input $\mathbf{u}'$ have the same time series values as the corresponding state elements and/or input elements in the original system $\mathbf{S}$, the elements in the state $\mathbf{x}'$ have the same time series values as the corresponding state elements in the original system $\mathbf{S}$. □

Intuitively, the new system $\mathbf{S}'$ extracts the exact information about the states in X'_s and P_c from the original system $\mathbf{S}$, and can be regarded as an independent subsystem of $\mathbf{S}$. Note that, in the new system $\mathbf{S}'$, all the inputs are known.

We explain the theorem using again the previous example shown in equations (1) and (8). Recall the state vertex set $X_s = \{x_{n_1}, x_{m_6}\}$, the cut point set $P_c = \{x_{m_2}, x_{m_3}, x_{m_4}, x_{m_5}, u_{m_1}\}$, and the corresponding extended state vertex set $X'_s = \{x_{n_1}, x_{n_2}, x_{m_6}\}$. By equations (1) and (8), if we only select the linear equations corresponding to the states in X'_s, we have:

$$\begin{aligned} \dot{x_{m_6}} &= a_{67}x_{n_1} + a_{68}x_{n_2} \\ \dot{x_{n_1}} &= a_{74}x_{m_4} + a_{75}x_{m_5} + a_{76}x_{m_6} \\ \dot{x_{n_2}} &= a_{82}x_{m_2} + a_{83}x_{m_3} + a_{86}x_{m_6} + b_{81}u_{m_1}. \end{aligned} \tag{13}$$

According to Theorem 1, if we write the above equations in matrix form, we get the state equation of the new system $\mathbf{S}'$:

$$\dot{\mathbf{x}'} = A'\mathbf{x}' + B'\mathbf{u}', \tag{14}$$

where $\mathbf{x}' = [x_{m_6}\ x_{n_1}\ x_{n_2}]^T$, $\mathbf{u}' = [x_{m_2}\ x_{m_3}\ x_{m_4}\ x_{m_5}\ u_{m_1}]^T$, and the system matrices

$$\begin{aligned} A' &= \begin{bmatrix} 0 & a_{67} & a_{68} \\ a_{76} & 0 & 0 \\ a_{86} & 0 & 0 \end{bmatrix}, \\ B' &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{74} & a_{75} & 0 \\ a_{82} & a_{83} & 0 & 0 & b_{81} \end{bmatrix}. \end{aligned} \tag{15}$$

In the new system $\mathbf{S}'$, all the inputs are known, and there is one measurable state in the state vector. If we assign the measurable state in the state vector as output, then the output is also known. We can then use classical system identification techniques to estimate the unknown parameters θ.

In our approach to field estimation, we use this theorem twice: first, with the detailed model lumped-parameter $M_{de}(\theta)$ with unknown inputs, we use the theorem to obtain a new detailed model $M(\theta)$ with both known inputs and known outputs, so that we can estimate the unknown parameters θ by system identification; second, for model reduction, we use it to obtain a reduced model with measurable states as input. We discuss this next.

C. Approach Description

The flow chart in Figure 3 summarizes our approach. For ease of the discussion, each block of the flow chart in Figure 3 is assigned an index i and denoted by Block i, $i = 1, \ldots, 7$. We next describe the overall approach in 4 steps using the flow chart as a reference. In our discussion, for convenience, the parameter vectors θ_0, θ, and θ_r can also be regarded as the sets of parameters.

STEP 1. Modeling (Blocks $1 \rightarrow 2$**)**. Block 1 represents the real world. We first construct a detailed finite-dimensional, lumped-parameter model M_{de}, as shown in Block 2, in terms of the field variables at sensor locations S and specific locations of interest R, plus any inputs and additional state variables required to define a

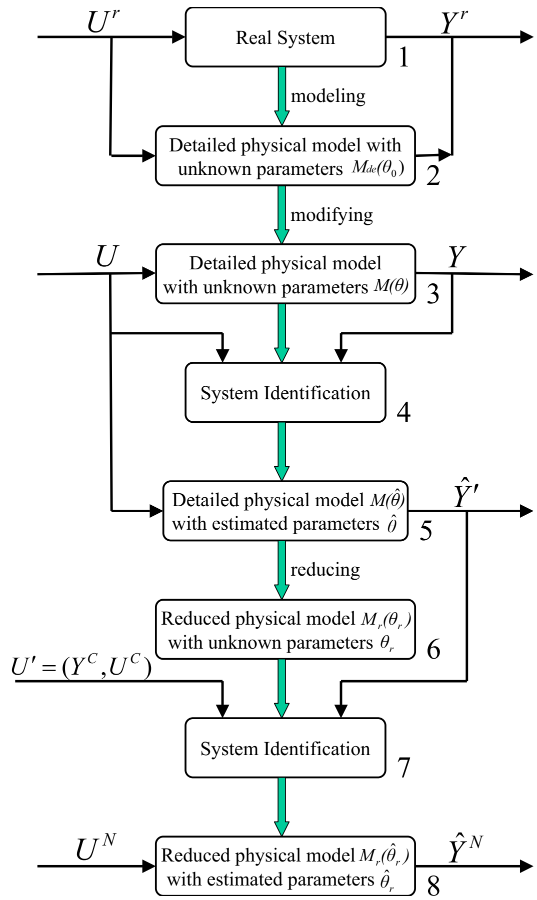

Fig. 3. Approach flow chart.

detailed model. We currently assume that we are given the detailed lumped-parameter model M_{de} and ignore the modeling process that starts from the first-principles model M_{fp} of the real system and results in the lumped-parameter model M_{de}. Usually the detailed lumped-parameter model M_{de} has unknown parameters θ_0; we make this explicit by referring to $M_{de}(\theta_0)$.

STEP 2. Parameter Estimation (Blocks $2 \rightarrow 3 \rightarrow 4 \rightarrow 5$**)**. Given experimental data, estimate the unknown parameters θ_0 in the detailed model $M_{de}(\theta_0)$ (Block 2). We use system identification techniques [9] that assume that both the input and the output of the system are known. We assign to the output of $M_{de}(\theta_0)$ available sensor measurements, i.e., the output Y^r is known. With respect to the input, we consider the two cases where the input U^r is known or the input U^r is unknown or partly known.

Case 1: input U^r is known.

In this case, the model modification process (Block $2 \rightarrow 3$) can be ignored. we estimate the parameters θ_0 directly by classical system identification techniques [9], which can be implemented by widely used tools found for example in MATLAB. We obtain the detailed model $M(\hat{\theta})$ with estimated parameters $\hat{\theta}$, as shown in Block 5, where $M = M_{de}$ and $\hat{\theta} = \hat{\theta_0}$.

Case 2: input U^r is unknown or partly known.

Some of the inputs may be difficult to measure directly, which makes it difficult to use classical system identification techniques to estimate the unknown parameter vector θ_0 in $M_{de}(\theta_0)$. We modify the original system $\mathbf{S}$ corresponding to $M_{de}(\theta_0)$ (Block 2), in particular, the assignment of system inputs and outputs, by Theorem 1 to obtain a new system $\mathbf{S}'$ that corresponds to a new detailed lumped-parameter model $M(\theta)$ (Block 3), where $\theta \subseteq \theta_0$. The new model $M(\theta)$ includes all the field locations of interest and has both known input and known output, so that we can use classical system identification techniques to estimate the unknown parameters θ, as shown in Blocks 4 and 5.

Referring to equation (4), the field variables $\mathbf{f}$ we want to estimate are actually a subset of unknown/unmeasurable states in the state vector $\mathbf{x}$. To apply Theorem 1, in the corresponding system digraph for the original system $\mathbf{S}$, we need to find the cut point set P_c for a state vertex set X_s, where X_s includes all the state vertices corresponding to the field variables in $\mathbf{f}$ and at least one measurable state vertex. Since $X_s \subseteq X_s'$, the corresponding extended state vertex set X_s' includes at least one measurable state vertex. By Theorem 1, we assign P_c as input, the corresponding extended state vertex set X_s' as state, the measurable state vertices in X_s' as output to obtain a new system $\mathbf{S}'$, denoted by model $M(\theta)$. Since the input and output of the new system are known, we use classical system identification techniques to estimate θ. Based on the resulting model $M(\hat{\theta})$ (Block 5), with estimated parameters $\hat{\theta}$, we estimate the unmeasurable state vertices in X_s, i.e., the field variables in $\mathbf{f}$.

It may appear that a maximal cut point set for X_s is preferable since more measurable vertices can be used to estimate the unknown model parameters. However, a larger cut point set may introduce more unknown parameters in the resulting new system. The tradeoff between the two factors will be the subject of further study.

STEP 3. Model Reduction (Blocks $5 \rightarrow 6 \rightarrow 7 \rightarrow 8$). The detailed model $M(\hat{\theta})$ (Block 5) that resulted in Step 2 may be computationally too intensive to be useful in practice to estimate the real-time field values at the locations of interest due to power constraints and limited communication bandwidth in sensor networks.

We consider now reduced-order models to the detailed model. We use this term with a meaning different from the usual one. In the literature, reduced-order models have the same number of inputs and outputs as the original (full-order) model. Our goal is different: it is to estimate the field values at the locations of interest using the smallest subset of sensors needed. We seek to establish a simpler lumped-parameter model than the detailed model $M(\hat{\theta})$. We refer to this simpler model as the *reduced model* M_r. With respect to the availability of the cut point set for the state variables we want to estimate, we consider the following two cases.

Case 1: we can find a cut point set.

We use again Theorem 1 to reduce the model. Let $X_s = \mathbf{f}$ and assume that we are able to find the minimal cut point set P_c for X_s. Choose P_c to be the input, the corresponding extended state vertex set X_s' to be the state, and X_s to be the output. According to Theorem 1, the system equation for the new reduced model $M_r(\hat{\theta_r})$ (Block 8) can be derived directly from the detailed model $M(\hat{\theta})$, where $\hat{\theta_r} \subset \hat{\theta}$. The important fact to note is that the reduced-order model M_r that we derive, although much simpler than the detailed model $M(\hat{\theta})$, is an exact model to compute the field values at the locations of interest. In this case, Blocks 6 and 7 should be ignored, and the reduced model $M_r(\theta_r)$ (Block 6) can be directly used for Step 4 (field estimation).

Case 2: we cannot find a cut point set.

In this case, it may not be possible to obtain the exact reduced model. We establish a reduced model $M_r(\theta_r)$ (Block 5) with unknown parameters θ_r whose inputs are some specified sensor measurements, and whose outputs are the field variables at the locations of interest, i.e., $\mathbf{f}$. Since we can estimate $\mathbf{f}$ from the detailed model $M(\hat{\theta})$, we are able to use classic system identification techniques to estimate the unknown parameters θ_r in the reduced model $M_r(\theta_r)$, as shown in Block 7. In this case, the resulting model $M_r(\hat{\theta_r})$ (Block 8) is an approximate reduced model for $M(\hat{\theta})$.

STEP 4. Field Estimation (Block 8). Given the measurements of a small subset of sensors, estimate the field variables at the locations of interest, i.e., $\mathbf{f}$. The reduced model $M_r(\hat{\theta_r})$ derived in Step 3 can be used to obtain the real-time estimation of the field values in $\mathbf{f}$. However, since, in most cases, the system dynamics change gradually, the parameter values of the reduced model should be updated regularly to reflect these changes. Therefore, Steps 2 and 3 are repeated regularly, with the update interval depending on system properties.

DISCUSSION In our approach described above, only the final step in the flow chart of Figure 3 refers to real-time computation. The "system identification" operations in Blocks 4 and 7 are performed off-line, applying standard algorithms to data collected from the sensor network. Both operations of model modification and model reduction, i.e., searching for cut point sets, are also performed off-line. After we obtain the resulting reduced model, we use it to implement on-line estimation.

A basic assumption in our approach is that a linear lumped-parameter model is appropriate to describe the spatial and/or temporal dynamics of the random field. This is common practice and extends easily to handle nonlinear fields by applying dynamic linearization, where we linearize around nominal operating points. This can be implemented in practice by a multi-model switch technique.

We have assumed that the sensor locations are known. However, in practice, precise location of the sensors is not required since the model estimation is achieved with the actual real data; in other words, the model that is estimated accounts for the actual physical interactions among the field measurements, regardless of the actual positions of the sensors being known or not to the modeler.

As we discussed under Definition 3 in page 3, we may find multiple cut point sets for a given state vertex set, i.e., we may obtain multiple reduced models with different scales. The tradeoff between estimation accuracy and model complexity is then an important issue in practice. Usually, a larger model leads to more accurate estimation, since it utilizes more information from the sensors. On the other hand, a larger model requires more computational power and more communication among sensors. How to select the optimal reduced model for a given task is an important issue in our future research.

IV. EXPERIMENTS

We present experimental results to validate the methodology for field estimation presented in this paper. We carried out an experiment using a subdivision (left side of the dashed line) of the Carnegie Mellon University Intelligent Workplace (IW). We placed multiple Crossbow temperature sensors at locations $S = \{s_1, ..., s_9\}$ and chose a single location r where the temperature is to be estimated, as shown in Figure 4. The wireless sensors communicate through the 802.11 wireless network protocol.

We construct a detailed lumped-parameter model. This model can be described by an RC thermal model $M_{de}(\theta_0)$ that includes all the sensor locations $S = \{s_1, ..., s_9\}$ and the desired location r and whose input corresponds to the ambient temperature and the temperature on the remaining of the IW building (right side of the dashed line in Figure 4). This model has parameters with known values (obtained from handbook data) and parameters with unknown values, i.e., θ_0 (due to the lack of information about the heat transfer characteristics of free air in this room). We assume that there is no sensor placed outside the building and in the right part of the building, therefore, the input of $M_{de}(\theta_0)$ is unknown. Since RC model can be represented by state-space equations, in the corresponding systen digraph, we can use Theorem 1 to establish a modified lumped-parameter model $M(\theta)$, where $\theta \subset \theta_0$, whose known input includes sensor measurements at the sensor locations $\{s_1, ..., s_7\}$, whose state includes temperatures at location $\{s_8, s_9, r\}$ and whose known output corresponds to sensor measurements at locations $\{s_8, s_9\}$.

We collected temperature measurements at the sensor locations S. We use then measurements to estimate the unknown parameters θ in the model M using system identification techniques. We now find a small subset of sensors $S' = \{s_2, s_3, s_6, s_8, s_9\}$ to estimate the temperature at the desired location r and establish a reduced lumped-parameter model M_r. Based on Theorem 1, the reduced model M_r can be derived by finding a minimal cut point set for {r} in the system digraph corresponding to the model M. Finally, given measurements of the sensors in S', we estimate the real-time temperature at the location r based on the model M_r. Reference [10] contains the details of this experiment. Figure 5 shows the results on estimating the temperature at the location r based on the reduced model M_r. The agreement displayed in the figure between the actual readings (solid line) and their prediction (dotted line) across the 4000 samples (except for occasional wild variations in the real measurement time series that are attributed to malfunctioning of the sensors between readings 2600 and 2800) shows that our methodology can successfully predict the temperature field at locations other than the ones where we have physically placed our sensors.

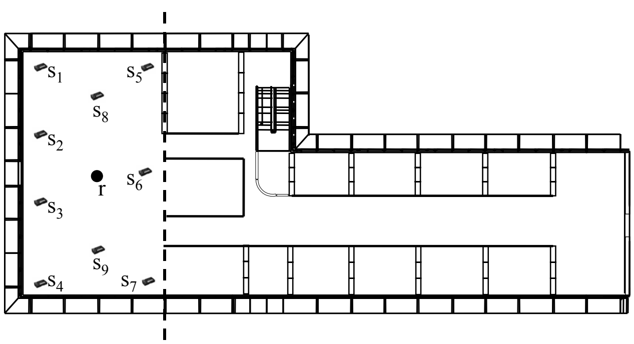

Fig. 4. Floor layout of the IW building.

V. CONCLUSION

In this paper, we introduce the problem of field estimation in sensor networks. We define three concepts: interconnection matrices, system diagraphs and cut point sets. These help us map the problem of correlated field estimation onto a graph. We present a theorem that shows that the field at arbitrary locations can be estimated directly from local sensors. We develop algorithms that can determine for each location of interest which local sensors to use to estimate the field. Finally, we present an experiment with real data using Crossbow temperature wireless sensors to show that the proposed graph-based approach can successfully estimate the real-time field values at locations where there are no sensors. We are currently performing a much larger experiment and studying the applicability of the method described in the paper to study the temperature distribution in a room housing a large farm of servers.

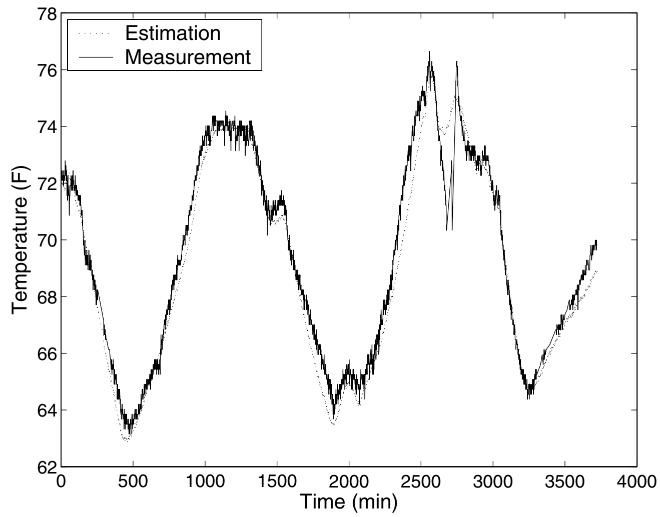

Fig. 5. Actual temperature and estimated temperature at location r for the IW.

REFERENCES

[1] A. T. P. So, "Building control and automation systems," in *Perspectives in Control Engineering*, T. Samad, Ed. IEEE Press, 2001, pp. 393–416.

[2] C. D. Patel, "A vision of energy aware computing from chips to data centers," in *Proc. of the International Symposium on Micro-Mechanical Eng. (ISMME)*, Tsuchiura and Tsukuba, Japan, Dec. 1-3, 2003.

[3] R. A. Parsons, Ed., *ASHRAE Handbook – Fundamentals (I-P Edition)*. American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., 1997, ch. "Heat Transfer".

[4] G. B. Folland, *Introduction to partial differential equations*. Princeton University Press, 1995.

[5] G. Hudson and C. P. Underwood, "A simple building modelling procedure for matlab/sumulink," in *Proceedings of Building Simulation '99*, vol. 2, Kyoto, Japan, Sept. 1999, pp. 777–783.

[6] M. Barcella, W. Huang, M. Stan, and K. Skadron, "Architecture-level compact thermal R-C modeling," in *Tech Report CS-2002-20, Univ. of Virginia Dept. of Computer Science*, USA, July 2002.

[7] D. D. Šiljak, *Decentralized Control of Complex Systems*. Academic Press, Inc., 1991, pp. 2–6.

[8] N. Deo, *Graph Theory with Applications to Engineers and Computer Science*. Prentice Hall, 1974.

[9] L. Ljung, *System Identification: Theory for the User*. Prentice Hall, 1999.

[10] H. Zhang, B. Krogh, J. M. F. Moura, and W. Zhang, "Estimation in virtual sensor-actuator arrays using reduced-order physical models," in *43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, Dec. 14-17 2004.