Observability and Independence in Distributed Sensing and its Application

Hiroshi Saito
NTT Network Technology Laboratories
3-9-11, Midori-cho, Musashino-shi, Tokyo 180-8585, Japan
E-mail: saito.hiroshi@lab.ntt.co.jp, URL: http://www9.plala.or.jp/hslab/

***Abstract*—This paper addresses the observability and independence of a system monitored using networked distributed sensors under the assumption that some sensing results are indiscriminable. We theoretically analyze the independence of sensing templates, that is, whether sensing systems are not redundant, and the observability of such systems, that is, whether we can determine the state of a system from the sensing results. Application examples to which the results given in this paper can be applied are: (1) end-to-end network quality monitored by end users in which the route actually used is not known and (2) object detection/non-detection by binary sensors randomly deployed without positioning mechanisms.**

I. Introduction

As a system becomes larger, it becomes more difficult to monitor what is happening inside the system or observe the entire system. Therefore, it is beneficial for a system operator to use information from outside a system or information brought into it by using distributed probing mechanisms from the outside and to estimate the inside of a system.

For such a situation, a natural question is: Can we estimate (observe) the inside of a system or extract meaningful information for the global view of the system by collecting and processing small pieces of distributed information? This question regarding observability seems general because we can easily find such examples.

– Network quality: Each subscriber communicates with another through a network. Some subscribers may submit a report to a network operator about quality. The quality of each link is either one of two states, GOOD or BAD. The route occasionally changes due to load balancing or other reasons. Subscriber reports include information about the source, destination, and quality for the source and destination, but it does not include the route. Can a network operator know the quality of each link based on subscriber reports?

– Sensors detecting targets: Sensors are randomly deployed (or many people deploy each sensor as a part of participatory sensing) to detect whether there are target objects. If there is a target object within the sensing area of each sensor, the sensor sends a sensing report of DETECT to a sensor data center; otherwise, it sends a sensing report of NON-DETECT. Each sensor has sensing and communication capabilities, but it does not have a positioning function (or it does not inform its location due to location privacy). Therefore, we do not know the location of sensors. Can we determine the number, size, shape, and location of targets?

These examples have the same structure: binary sensing reports from distributed sensors under partially unknown conditions are collected to estimate the state of the system. The analysis of this paper covers these examples.

Observability for binary information systems is being discussed in multiple research areas such as networking, testing, and sensing.

In networking research, fault localization [1], [2] and network tomography [3]–[10] are concrete examples. The former is a technology for identifying the location of faults/failures mainly used in the wavelength-division multiplexing network. For example, Talpolcai et al. [1] proposed a localization method by using m-trail, which is a supervisory lightpath dedicated for failure localization. The ON-OFF status of this lightpath indicates whether or not a link failure event occurs along the lightpath. Each node identifies the failure link by a set of the detected statuses of m-trails. The latter is a technology that estimates the quality of each link of a network by using probes for monitoring the quality of each route (path), although it sometimes means the inference of routing topology or traffic matrix. The probe is normally an end-to-end unicast packet, though some researchers assume the use of multicast [11]–[14]. In general, network tomography does not model the network states or the observation results as binary variables, but the following studies model them as binary; thus, they are relevant to this paper. Zeng et al. discussed the generation of automatic test packets covering the points that may have problems [15], which also belongs to the testing research area. By using the test packets, identifying the problematic points is also discussed. Nguyen et al. [16] investigated a method of identifying a link as BAD by observing the binary state of the path. They introduced the concept of "identifiability", meaning that the link state probability vector can be identified by the observation of the path. Duffield [17] proposed an algorithm called the Smallest Consistent Failure Set (SCFC) inference algorithm for a tree network. The SCFC algorithm estimates BAD links based on the quality observations for the paths between the root and leaves. Nguyen et al. [18] investigated the probe selection problem to efficiently perform network tomography. The results shown in [18] are useful in discussing the results discussed in this paper. It is normally assumed with network tomography that the route is fixed, although we assume that the route can change in the example of network quality in this paper.

In testing research, the testability theory for digital circuits is the most relevant. Test patterns randomly chosen are offered to a combinatorial circuit with a fault. Research objectives are, for example, to generate test patterns to efficiently cover a given fault [19] and to compute the probability of detecting the fault [20]. It is known that generating a test for a single stuck-at fault in a circuit requires solving an NP-complete problem, and identifying the redundancy of a single stuck-at fault requires solving a co-NP-complete problem [19]. A model commonly used in testability theory for digital circuits is specific case of the model proposed in Section III-A.

There are studies we need to mention in sensing research. By obtaining a sensor report that provides binary information on whether a sensor detects a target object through a network such as that proposed in [21], [22], some parameters of target objects, such as size, perimeter length, and angles, can be estimated but other parameters cannot be [23]–[27]. These methods can be regarded as software sensors (soft sensors) with which sensors are randomly placed. In particular, [26] introduced the concept of a composite sensor node, which consists of multiple sensors arranged in a predetermined layout, such as a line, and defined observability by the existence of composite sensor node parameter values that distinguish two sets of parameter values of the target object shape. That is, if observable, we can expect that obtained binary sensor reports can uniquely determine the target object shape parameters under an ideal situation by using sensing results and a certain set of composite sensor node parameter values.

Another research area in sensing research is observability under sensor failure or sensing schedule [28]–[31]. These studies discussed methods for (i) determining if a given system remains observable in case of sensor failure, (ii) finding the minimum set of sensors to maintain observability, (iii) optimally placing sensors, and (iv) finding the policy for achieving optimum observability. However, these studies do not cover the topics discussed in this paper. In addition, although Kar et al. [32] studied parameter estimation with sensing results where a sensor may be under failure and sensing results may include errors, it does not cover the concept that we cannot distinguish sensing results for the sensing templates in the same class. This concept is one of the main features differentiating other studies from ours.

This paper introduces classes of sensing templates and models the fact that we cannot determine which sensor (sensing template) corresponds to a sensing result. In each class of sensing templates, a sensor is chosen with a given probability. This model covers a large class of problems, including those mentioned above, and bridges existing results in different research areas. This paper argues that observability is more applicable in a wider variety of applications than in conventional literature if it is unifiedly defined, and can contribute to determining whether meaningful results can be extracted from collected and processed small pieces of distributed binary information. We should note that there are unobservable cases. For such cases, no appropriate solution is a correct solution even though computer programs may derive some solutions.

II. Notations

In the remainder of this paper, a (row) vector is denoted in boldface if explicitly indicated otherwise. The following is a list of notations used in this paper.

- $\mathbf{I}(x) \equiv \begin{cases} 1, & \text{if } x \text{ is true} \\ 0, & \text{otherwise} \end{cases}$: an indicator function.
- $\mathbf{0}_j \equiv (0, \cdots, 0)$: the j-element zero vector.
- $\mathbf{e}_i \equiv (e_{i,1}, \cdots, e_{i,n_s})$: the i-th elementary n_s-element vector where $e_{i,j} = \mathbf{I}(i = j)$.
- $\mathbf{1}_j \equiv (1, \cdots, 1)$: the j-element vector of which elements are 1.
- $(\mathbf{x})_i$: the i-th element of any vector $\mathbf{x} = (x_1, \cdots, x_n)$.
- $x_i \cup y_i \equiv \max(x_i, y_i)$ for $x_i, y_i \in \{0, 1\}$.
- $\mathbf{x} \cup \mathbf{y} \equiv (x_1 \cup y_1, \cdots, x_n \cup y_n)$ for any binary vector $\mathbf{x} = (x_1, \cdots, x_n)$ and $\mathbf{y} = (y_1, \cdots, y_n)$. That is, $\mathbf{x} \cup \mathbf{y}$ is the element-wise OR operation for $\mathbf{x}$ and $\mathbf{y}$.
- $\mathbf{x} \cap \mathbf{y}$: the element-wise AND operation for $\mathbf{x}$ and $\mathbf{y}$.
- $\mathbf{x} \star \mathbf{y} \equiv \mathbf{I}(\sum_{i=1}^{n} x_i y_i > 0)$. The operator $\star$ means "sensing", as discussed in a later section.
- $\mathbb{B}(\mathbf{x}) \equiv \sum_{i=1}^{n} 2^{n-i} (\mathbf{x})_i$ for a binary vector $\mathbf{x}$. That is, $\mathbb{B}(\mathbf{x})$ is an operator that regards $\mathbf{x}$ as a binary number and expresses it as a decimal integer. For a given integer i, we can define an operator of the inverse of $\mathbb{B}(\mathbf{x}) = i$, that is, $\mathbf{x} = \mathbb{B}^{-1}(i)$.
- $|S|$: the number of elements in S.
- $S - x_i$: the set $\{x_1, x_2, \cdots, x_{i-1}, x_{i+1} \cdots, x_m\}$ for an element $x_i \in S = \{x_1, x_2, \cdots, x_m\}$.

III. Model

A. Model description

Let $\mathbf{s} = (s_1, \cdots, s_{n_s}) \in S_s$ be a binary state vector, where $S_s \subsetneqq 2^{n_s}$ is a given set of possible states. When there is (is not) a target at i, $s_i = 1$ (0).

Let $\mathbf{a}$ be a sensing template, where $(\mathbf{a})_i = 1$ means that we can observe s_i (Normally, a deployed sensor corresponds to a sensing template, which is similar to a sensing area. However, it is likely that the sensing area is a continuous area or a set of contiguous points on the discrete domain. On the other hand, the sensing template may not be composed of consecutive points.) The operator $\star$ means "sensing" because $\mathbf{a} \star \mathbf{s} = \mathbf{I}(\sum_{i=1}^{n_s} (\mathbf{a})_i s_i > 0)$ is the sensing result of $\mathbf{s}$ by $\mathbf{a}$. Here, $\mathbf{a} \star \mathbf{s} = 1$ (0) means that the sensing template $\mathbf{a}$ detects (does not detect) a target or targets in $\mathbf{s}$. That is, when there is a target occupying at least one of the observed points defined by the sensing template $\mathbf{a}$, $\mathbf{a} \star \mathbf{s} = 1$; otherwise, $\mathbf{a} \star \mathbf{s} = 0$. The sensing templates mentioned in the remainder of this paper are not identical to $\mathbf{0}_{n_s}$ unless explicitly indicated otherwise. Let $A_s \equiv \{\mathbf{a}_1, \cdots, \mathbf{a}_{n_a}\}$ be the set of all the sensing templates available, where $n_a \equiv |A_s|$ and $\mathbf{a}_i \neq \mathbf{a}_j$ for any $i \neq j$.

Let $\mathbf{b}(i)$ be an n_a-element column vector of which the j-th element $(\mathbf{b}(i))_j$ is given by $\mathbf{a}_j \star (\mathbb{B}^{-1}(i))$ for $j = 1, \cdots, n_a$ and define the matrices $B(2^{n_s}) \equiv (\, \mathbf{b}(0) \quad \cdots \quad \mathbf{b}(2^{n_s} - 1) \,)$ and $B(S_s) \equiv (\, \mathbf{b}(i_1) \quad \cdots \quad \mathbf{b}(i_j) \,)$, where $i_1 < \cdots < i_j$ and $S_s = \{\mathbb{B}^{-1}(i_1), \cdots, \mathbb{B}^{-1}(i_j)\}$.

We now introduce the concept of "classes of sensing templates." If multiple sensing templates are in the same class, we cannot determine which sensing template corresponds to a sensing result. Let n_c be the number of classes, c_i be the i-th class of sensing templates, and $|c_i|$ be the number of sensing templates in class c_i. When class-i is used, the sensing template $\mathbf{a}_j$ is independently used with probability $p_{i,j}, \sum_{j=1}^{n_a} p_{i,j} = 1$. If the sensing template $\mathbf{a}_j$ is not included in c_i, $p_{i,j} = 0$. Define $u_i(\mathbf{s}) \equiv \sum_{j=1}^{n_a} p_{i,j}(\mathbf{a}_j \star \mathbf{s})$, which is the probability that a sensing result is 1 when a class-i sensing template is used, and $\mathbf{u}(\mathbf{s}) \equiv (u_1(\mathbf{s}), \cdots, u_{n_c}(\mathbf{s}))$. Define $\mathbf{p}_i \equiv (p_{i,1}, \cdots, p_{i,n_a})$.

When $|c_i| = 1$, we call this class the reduced class. The conventional model discussed in [15]–[18] corresponds to $|c_i| = 1$ with $u_i = 1$ or 0 for all i and $p_{i,j} = 1$ or 0 for all i, j. In [15], the greedy algorithm provides a good approximated solution for the minimum set covering all the state, although the minimum set coverage problem is NP-complete.

The following is the summary of variables.

- A_s: the set of all available sensing templates.
- $\mathbf{a}$: a sensing template.
- $\mathbf{a}_i$: the i-th sensing template.
- $\mathbf{b}(i) \equiv {}^t(\mathbf{a}_1 \star (\mathbb{B}^{-1}(i)), \cdots, \mathbf{a}_{n_a} \star (\mathbb{B}^{-1}(i)))$.
- $B(2^{n_s}) \equiv (\, \mathbf{b}(0) \quad \cdots \quad \mathbf{b}(2^{n_s}-1) \,)$.
- $B(S_s) \equiv (\, \mathbf{b}(i_1) \quad \cdots \quad \mathbf{b}(i_j) \,)$, where $i_1 < \cdots < i_j$ and $S_s = \{\mathbb{B}^{-1}(i_1), \cdots, \mathbb{B}^{-1}(i_j)\}$.
- c_i: the i-th class of sensing templates.
- $|c_i|$: the number of sensing templates in class c_i.
- $n_a \equiv |A_s|$.
- n_c: the number of classes.
- $p_{i,j}$: the prob. that the sensing template $\mathbf{a}_j$ is used in c_i.
- $\mathbf{p}_i \equiv (p_{i,1}, \cdots, p_{i,n_a})$. $P \equiv \begin{pmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_{n_c} \end{pmatrix}$.
- $\mathbf{s} \equiv (s_1, \cdots, s_{n_s})$: a binary vector describing the state.
- S_s: a set of possible states.
- $u_i(\mathbf{s}) \equiv \sum_j p_{i,j}(\mathbf{a}_j \star \mathbf{s})$. $\mathbf{u}(\mathbf{s}) \equiv (u_1(\mathbf{s}), \cdots, u_{n_c}(\mathbf{s}))$.

B. Example

Consider a ring network with $2n_d + 1$ nodes as a concrete example of "network quality" as mentioned in the Introduction. The state can be described as $(s_1, \cdots, s_{n_s})$, where $s_i = 0$ (1) means the GOOD (BAD) quality of the link between the i-th node and $i+1$-th node and $n_s = 2n_d + 1$. If and only if all the links included in the route between a source and a destination are GOOD quality, the quality of the source-destination pair is GOOD (this property is called separable [17]). Let $\mathbf{a}_{j+(i-1)n_s}$ be the sensing template corresponding to the i-hop route $j - (j+1) - \cdots - (j+i)$ (mod n_s). Here, $(\mathbf{a}_{j+(i-1)n_s})_k = 1$ (0) means that the link between the k-th node and $k+1$-th node is included (not included) in this route.

Between each pair of nodes, a shorter hop route is chosen with probability p, and the other route is chosen with probability $1-p$. For simplicity, the quality of both directions of a link is assumed to be the same. Because we observe the quality between a source-destination pair without knowing the route used for the observed quality, each pair of nodes corresponds to a class of sensing templates with two members: one template for the shorter route around the ring, and another for the longer. Therefore, $n_c = (n_s - 1)n_s/2 = (2n_d + 1)n_d$ and $n_a = 2(2n_d + 1)n_d$.

For simplicity, consider $n_d = 1$. Then, for example, $\mathbf{a}_1 = (1, 0, 0)$ corresponds to the quality report for the direct link between nodes 1 and 2. Similarly, $\mathbf{a}_4 = (1, 1, 0)$ corresponds to the quality report for the route 1-2-3. When the link between nodes 1 and 2 is BAD and other links are GOOD, $\mathbf{s} = (1, 0, 0) = \mathbb{B}^{-1}(4)$. Hence, $\mathbf{b}(4) = {}^t(1, 0, 0, 1, 0, 1)$. Similarly, $B(2^{n_s}) = \begin{pmatrix} 0&0&0&0&1&1&1&1 \\ 0&0&1&1&0&0&1&1 \\ 0&1&0&1&0&1&0&1 \\ 0&0&1&1&1&1&1&1 \\ 0&1&1&1&0&1&1&1 \\ 0&1&0&1&1&1&1&1 \end{pmatrix}$. Between nodes 1 and 2, the one hop route corresponds to $\mathbf{a}_1$ and the two hop route corresponds to $\mathbf{a}_5$. Therefore, if the first, second, and third classes correspond to the node pairs between nodes 1 and 2, 2 and 3, and 3 and 1, respectively, $P = \begin{pmatrix} p&0&0&0&1-p&0 \\ 0&p&0&0&0&1-p \\ 0&0&p&1-p&0&0 \end{pmatrix}$. For example, $u_1(\mathbf{s}) = p(\mathbf{a}_1 \star \mathbf{s}) + (1-p)(\mathbf{a}_5 \star \mathbf{s})$ is p if the link between nodes 1 and 2 is BAD and other links are GOOD, $1-p$ if the link between nodes 1 and 2 is GOOD and one of the other links is BAD, 0 if all the links are GOOD, and 1 otherwise.

IV. Analysis

A. Definition

Definition 1: "$\{c_1, \cdots, c_{n_c}\}$ can distinguish $\mathbf{s}_1$ from $\mathbf{s}_2 (\neq \mathbf{s}_1)$" means that $\mathbf{u}(\mathbf{s}_1) \neq \mathbf{u}(\mathbf{s}_2)$ for given $\mathbf{s}_1$ and $\mathbf{s}_2$. "$\mathbf{s}_1$ is observable by $\{c_1, \cdots, c_{n_c}\}$" means that we can distinguish $\mathbf{s}_1$ from any other state by $\{c_1, \cdots, c_{n_c}\}$.

For the conventional model, the definition of "distinguish" given above is identical to that defined in [18].

Definition 2: If and only if there exists a (linear) function f such that $u_1(\mathbf{s}) = f(u_2(\mathbf{s}), \cdots, u_{n_c}(\mathbf{s}))$ for any binary vector $\mathbf{s}$, c_1 is (linearly) constructible from the set $\{c_2, \cdots, c_{n_c}\}$. If and only if c_1 is not (linearly) constructible from the set $\{c_2, \cdots, c_{n_c}\}$, c_1 is strongly independent (linearly independent) from the set $\{c_2, \cdots, c_{n_c}\}$.

The independence defined in [18] is equivalent to that defined above with f in which $\mathbf{a}_1 \star \mathbf{s} = \cup_{2 \leq i \leq n_c} t_i \mathbf{a}_i \star \mathbf{s}$ ($t_i \in \{0, 1\}$) for the conventional model.

B. Main theorem

In this subsection, we discuss how we determine that sensing template classes are mutually redundant and that observability is achieved through sensing template classes.

Theorem 1: For any S_s, (i) any given state is observable if and only if no two columns in $PB(S_s)$ are identical, and (ii) any given class is linearly independent if and only if all the row vectors in $PB(S_s)$ are linearly independent.

Proof: Note that ${}^t\mathbf{u}(\mathbf{s})$ is given by a column vector in $PB(S_s)$, where $\mathbf{s} \in S_s$. According to the definition of observability, any state is observable if and only if no two columns in $PB(S_s)$ are identical.

Note that the i-th row vector of $PB(S_s)$ is $(u_i(\mathbf{s}_{i_1}), \cdots, u_i(\mathbf{s}_{i_j}))$, where $i_1 < \cdots < i_j$ and

$S_s = \{\mathbb{B}^{-1}(i_1), \cdots, \mathbb{B}^{-1}(i_j)\}$. According to the definition of linear independence, any class is linearly independent if and only if all the row vectors of $PB(S_s)$ are linearly independent. ■

Theorem 2: Consider two sets of possible states S'_s and S_s such that $S'_s \subset S_s$.

(1) Any state observable in S_s is observable in S'_s.

(2) A class c_1 is linearly constructible from $\{c_2, \cdots, c_{n_c}\}$ for S'_s if c_1 is linearly constructible from $\{c_2, \cdots, c_{n_c}\}$ for S_s.

Proof: The set of column vectors in $PB(S'_s)$ is a subset of column vectors in $PB(S_s)$. Therefore, if no two column vectors in $PB(S_s)$ are identical, no two column vectors in $PB(S'_s)$ are either identical.

Assume that there exists a linear function f such that $u_1(\mathbf{s}) = f(u_2(\mathbf{s}), \cdots, u_{n_c}(\mathbf{s}))$ for any binary vector $\mathbf{s} \in S_s$. Because $\mathbf{s} \in S'_s$, f satisfies $u_1(\mathbf{s}) = f(u_2(\mathbf{s}), \cdots, u_{n_c}(\mathbf{s}))$ for any $\mathbf{s} \in S'_s$. ■

The meaning of this theorem is intuitive. The former part of this theorem means that a state we can uniquely identify in a set of states S_s, can, of course, be uniquely identified in a smaller set of states than S_s. The latter part roughly means that if u_1 for any state $\mathbf{s}$ can be constructed using other classes of sensing templates for a large state space, it is possible to construct u_1 using these other classes for a small state space.

Theorem 2 provides us with an important suggestion: If we can reasonably limit the possible states, it is likely that we can identify each state, even when the system is not observable for a large possible state set. On the other hand, if we can limit the possible states, sensing results may become redundant even when the original classes of sensing templates are linearly independent for a large possible state set.

Consider the example in Section III-B. Its possible state set S_s typically consists of all the combinations of BAD/GOOD links. However, if it is unlikely that all the links are BAD, we can limit possible states to a subset S'_s. For S_s, we may not be able to determine which links are BAD in quality. However, Theorem 2 shows that we may be able to determine these for S'_s.

Lemma 1: $n_a \leq 2^{n_s} - 1$.

Proof: Because $\mathbf{a}$ has n_s elements and $(\mathbf{a})_i \in \{0, 1\}$, there are at most $2^{n_s} - 1$ variations except for $\mathbf{a} \neq \mathbf{0}_{n_s}$. ■

Lemma 2: The rank of matrix $B(2^{n_s})$ is n_a.

The proof is shown in Appendix.

Theorem 3: If the rank of P is equal to n_c, $u_1(\mathbf{s}), \cdots, u_{n_c}(\mathbf{s})$ are linearly independent of each other when $S_s = 2^{n_s}$. If the rank of P is less than n_c, at least one of $u_i(\mathbf{s}) \in \{u_k(\mathbf{s})\}_{1 \leq k \leq n_c}$ is linearly constructible by other $\{u_k(\mathbf{s})\}_{k \neq i}$ for any S_s.

Proof: Assume $S_s = 2^{n_s}$. For a vector $\mathbf{w} \equiv (w_1, \cdots, w_{n_c})$ and a given state $\mathbf{s}$, $\sum_{i=1}^{n_c} w_i u_i(\mathbf{s}) = \sum_{i=1}^{n_c} w_i \sum_{k=1}^{n_a} p_{i,k}(\mathbf{a}_k \star \mathbf{s}) = \mathbf{w}P\mathbf{b}(\mathbb{B}(\mathbf{s}))$. Therefore, $(\sum_{i=1}^{n_c} w_i u_i(\mathbb{B}^{-1}(0)), \cdots, \sum_{i=1}^{n_c} w_i u_i(\mathbb{B}^{-1}(2^{n_s}-1))) = \mathbf{w}PB(2^{n_s})$. Note that columns of $B(2^{n_s})$ include $\begin{pmatrix} \mathbf{a}_1 \star \mathbf{s} \\ \vdots \\ \mathbf{a}_{n_a} \star \mathbf{s} \end{pmatrix}$ for any $\mathbf{s}$. Hence, "linearly constructible" means that $\mathbf{w}PB(2^{n_s}) = \mathbf{0}_{2^{n_s}}$ for $\mathbf{w} \neq \mathbf{0}_{n_c}$. If the rank of matrix $PB(2^{n_s})$ is equal to n_c, $\mathbf{w}PB(2^{n_s}) \neq \mathbf{0}_{2^{n_s}}$ for $\mathbf{w} \neq \mathbf{0}_{n_c}$. Because of Lemmas 1 and 2, the rank of matrix $PB(2^{n_s})$ is equal to the rank of P. Thus, if the rank of P is equal to n_c, $u_1(\mathbf{s}), \cdots, u_{n_c}(\mathbf{s})$ are linearly independent of each other.

If the rank of P is less than n_c, there exists a vector $\mathbf{w} \neq \mathbf{0}_{n_c}$ such that $\mathbf{w}P = \mathbf{0}_{n_a}$. Thus, $\mathbf{w}PB(2^{n_s}) = \mathbf{0}_{2^{n_s}}$. This means that at least one of $u_i(\mathbf{s}) \in \{u_k(\mathbf{s})\}_{1 \leq k \leq n_c}$ is linearly constructible when $S_s = 2^{n_s}$. By using Theorem 2, at least one of $u_i(\mathbf{s}) \in \{u_k(\mathbf{s})\}_{1 \leq k \leq n_c}$ is linearly constructible for any S_s. ■

In the example of network quality, consider the ring network discussed in Subsection III-B with $n_d = 1$. Because $PB(2^{n_s}) = \begin{pmatrix} 0 & 1-p & 1-p & 1-p & p & 1 & 1 & 1 \\ 0 & 1-p & p & 1 & 1-p & 1-p & 1 & 1 \\ 0 & p & 1-p & 1 & 1-p & 1 & 1-p & 1 \end{pmatrix}$, its rank is 3. Thus, according to Theorem 3, u_1, u_2, u_3 are linearly independent. On the other hand, when we limit the possible states to be $S_s = \{(0,0,1), (0,1,0), (1,0,0)\}$ (that is, the possible states are those in which one of the transit links is BAD), $B(S_s)$ is the matrix made from the second, third, and fifth columns of $B(2^{n_s})$. Then, the rank of $PB(S_s) = \begin{pmatrix} 1-p & 1-p & p \\ 1-p & p & 1-p \\ p & 1-p & 1-p \end{pmatrix}$ is 1 for $p = 1/2$ and 3 for $p \neq 1/2$. Therefore, according to Theorem 3, u_1, u_2, u_3 are linearly constructive from each other when $p = 1/2$. In practice, because $s_i \cup s_j = s_i + s_j$ $(s_i \neq s_j)$ for this S_s, $u_k(\mathbf{s}) = (s_1 + s_2 + s_3)/2$ for all k when $p = 1/2$.

Theorem 4: If $\mathbf{b}(\mathbb{B}(\mathbf{s})) \neq \mathbf{b}(\mathbb{B}(\mathbf{s}'))$ for any $\mathbf{s} \neq \mathbf{s}'$ and if the rank of P is n_a, observability is achieved. If there exist $\mathbf{s} \neq \mathbf{s}'$ such that $\mathbf{b}(\mathbb{B}(\mathbf{s})) = \mathbf{b}(\mathbb{B}(\mathbf{s}'))$, observability is not achieved.

Proof: Because of Theorem 1, observability is equivalent to that in which no two column vectors in $PB(S_s)$ are equal. If $\mathbf{b}(\mathbb{B}(\mathbf{s})) \neq \mathbf{b}(\mathbb{B}(\mathbf{s}'))$ for any $\mathbf{s} \neq \mathbf{s}'$ and if the rank of P is n_a, the two column vectors corresponding to $\mathbf{s}$ and $\mathbf{s}'$ in $PB(S_s)$ cannot be the same. Thus, observability is achieved. If there exist $\mathbf{s} \neq \mathbf{s}'$ such that $\mathbf{b}(\mathbb{B}(\mathbf{s})) = \mathbf{b}(\mathbb{B}(\mathbf{s}'))$, $P\mathbf{b}(\mathbb{B}(\mathbf{s})) = P\mathbf{b}(\mathbb{B}(\mathbf{s}'))$. Thus, observability is not achieved. ■

Theorems 3 and 4 show that the rank of P plays an important role. Because the condition "$\mathbf{b}(\mathbb{B}(\mathbf{s})) \neq \mathbf{b}(\mathbb{B}(\mathbf{s}'))$ for any $\mathbf{s} \neq \mathbf{s}'$" is the observability for the conventional model, we can say that the rank of P determines the independence and observability of the proposed model. Reduction in the rank of P normally occurs when elements of P satisfy a certain equation such as a linear equation. Therefore, independence is normally achieved almost everywhere in the parameter space in P. Observability is also normally achieved almost everywhere in the parameter space in P if the system without the probabilistic mixture (that is, the conventional model) is observable. In the numerical examples given in Section VI, however, the accuracy of parameter estimation deteriorates around the point at which observability is not achieved in the parameter space when the number of samples is finite. Simultaneously, we can say that, if we can design P, we should set $n_c = n_a$ and make P whose rank is $n_c (= n_a)$

to satisfy linear-independence and observability.

For the conventional model, Theorems 3 and 4 seem inconsistent with the results given in [18]. This is because the definition of independence in [18] differs from the definition of the linear independence in this paper to discuss the effect of introducing classes. In fact, this is not inconsistent.

When we focus on the example of network quality, we can derive the concrete result shown below. In this corollary, the case that the rank of P becomes less than n_c can typically occur when we observe some links with prob. 1 in a tree network and there is a source-destination pair for which these links form the route.

Corollary 1: Consider probing a network. Each class corresponds to a source-destination pair, and each sensing template corresponds to a route between them. Then, the rank of P is n_c, and the sensing template classes are linearly independent when $S_s = 2^{n_s}$.

Proof: We should note that each sensing template appears only in a single sensing template class and does not appear in other sensing template classes. This means that each column of P has a single non-zero element. Therefore, the rank of P is n_c, and the sensing template classes are linearly independent if we consider 2^{n_s} as a set of possible states. ■

V. Erroneous sensing

We extend the model discussed in Section III-A to the model in which there are sensing errors. Define an erroneous sensing result $\widetilde{\mathbf{a}\star\mathbf{s}}$ using $\mathbf{a}\star\mathbf{s}$ (the sensing result without errors).

$$\widetilde{\mathbf{a}\star\mathbf{s}} = \begin{cases} 0, & \text{with prob. } p_{i\to 0} \text{ when } \mathbf{a}\star\mathbf{s} = i \\ 1, & \text{with prob. } p_{i\to 1} \text{ when } \mathbf{a}\star\mathbf{s} = i. \end{cases} \quad (1)$$

For the erroneous sensing result, define $\tilde{u}_i(\mathbf{s}) \equiv \sum_{j\in c_i} p_{i,j}\widetilde{\mathbf{a}\star\mathbf{s}}$, which is the probability that an erroneous sensing result is 1 when a class-i sensing template is used. Then,

$$\begin{aligned} & \tilde{u}_i(\mathbf{s}) \\ = & \sum_{j\in c_i} p_{i,j}p_{1\to 1}\mathbf{I}(\mathbf{a}_j\star\mathbf{s}=1) + \sum_{j\in c_i} p_{i,j}p_{0\to 1}\mathbf{I}(\mathbf{a}_j\star\mathbf{s}=0) \\ = & (p_{1\to 1}-p_{0\to 1})\sum_{j\in c_i} p_{i,j}\mathbf{I}(\mathbf{a}_j\star\mathbf{s}=1) + p_{0\to 1}\sum_{j\in c_i} p_{i,j} \\ = & (p_{1\to 1}-p_{0\to 1})u_i(\mathbf{s}) + p_{0\to 1}\sum_{j\in c_i} p_{i,j}. \end{aligned} \quad (2)$$

Consequently, if $p_{1\to 1} \neq p_{0\to 1}$ and we know $p_{1\to 1} - p_{0\to 1}$ and $\sum_{j\in c_i} p_{i,j}$, Definitions 1 and 2 using u_i are equivalent to Definitions 1 and 2 using $\tilde{u}_i$. Therefore, we can use the results under the assumption that there are no sensing errors.

VI. Numerical examples

A. Network quality 1

Consider the ring network described in Subsection III-B. The possible states we assume are those in which at most n_B links are BAD.

Note that $\mathbf{b}(\mathbb{B}(\mathbf{s})) \neq \mathbf{b}(\mathbb{B}(\mathbf{s}'))$ for any $\mathbf{s} \neq \mathbf{s}'$. This is because if $(\mathbf{s})_i \neq (\mathbf{s}')_i$, then $\mathbf{a}\star\mathbf{s} \neq \mathbf{a}\star\mathbf{s}'$ for the sensing template $\mathbf{a} = \mathbf{e}_i \in A_s$. We should also note that $(P\mathbf{b}(\mathbb{B}(\mathbf{s})))_i = p\mathbf{a}_s\star\mathbf{s} + (1-p)\mathbf{a}_l\star\mathbf{s}$, where $\mathbf{a}_s$ is a sensing template for the shorter hop route between a certain pair of nodes and $\mathbf{a}_l$ is that for the longer hop route between these nodes. For two states $\mathbf{s}, \mathbf{s}'$ ($\mathbf{s} \neq \mathbf{s}'$), there exists $\mathbf{a}_s$ such that $\mathbf{a}_s\star\mathbf{s} \neq \mathbf{a}_s\star\mathbf{s}'$. This is because, if $(\mathbf{s})_i \neq (\mathbf{s}')_i$, $\mathbf{a}_s = \mathbf{e}_i$ makes $\mathbf{a}_s\star\mathbf{s} \neq \mathbf{a}_s\star\mathbf{s}'$. Thus, we can assume that $\mathbf{a}_s\star\mathbf{s} = 1$, $\mathbf{a}_s\star\mathbf{s}' = 0$ without loss of generality. Then, if $\mathbf{a}_l\star\mathbf{s} \geq \mathbf{a}_l\star\mathbf{s}'$, $P\mathbf{b}(\mathbb{B}(\mathbf{s})) \neq P\mathbf{b}(\mathbb{B}(\mathbf{s}'))$ for $p > 0$. If $\mathbf{a}_l\star\mathbf{s} = 0$, $\mathbf{a}_l\star\mathbf{s}' = 1$, $(P\mathbf{b}(\mathbb{B}(\mathbf{s})))_i = p \neq 1-p = (P\mathbf{b}(\mathbb{B}(\mathbf{s}')))_i$ for $p \neq 1/2$. As a result, if $p \neq 0, 1/2$, $P\mathbf{b}(\mathbb{B}(\mathbf{s})) \neq P\mathbf{b}(\mathbb{B}(\mathbf{s}'))$. According to Theorems 1 and 2, the system is observable, for any n_B for $p \neq 0, 1/2$. In addition, if $p = 1/2$, $P\mathbf{b}(\mathbb{B}(\mathbf{e}_i)) = P\mathbf{b}(\mathbb{B}(\mathbf{e}_j))$, that is, the system is not observable.

The relationship between the observability and the state estimation errors is now investigated. The theoretical results mentioned in this paper are based on the sensing results $\mathbf{u}(\mathbf{s})$. Unfortunately, however, we cannot obtain $\mathbf{u}(\mathbf{s})$ because of the finite number of sensing samples. Therefore, even when the system is observable, it may be difficult to estimate the system state with the finite number of sensing samples.

We assume that n_R layer-3 reports are available for each pair of nodes i, j but do not include the route information, and that $100r_B(i,j)\%$ of these n_R reports state BAD quality. (The assumption of no route information is validated when the ring network is a physical network and is a logically mesh network where traffic into an incoming layer-3 node is transferred to an outgoing layer-3 node through layer 1/2 functions at the nodes between them.) Define $\mathbf{r}_B \equiv (r_B(1,2), \cdots, r_B(n_s-1, n_s))$. Based on $r_B(i,j)$, we estimate the state of the network by finding $\mathbf{s}$ among the possible states to minimize $\min_{\mathbf{s}} |E[\mathbf{r}_B] - \mathbf{r}_B| = \min_{\mathbf{s}} |\mathbf{u}(\mathbf{s}) - \mathbf{r}_B|$. (Although this estimation method is brute force, the following result is almost insensitive to the estimation method used.)

The results using a computer simulation are in Fig. 1, where the true state $s_{TRUE} \neq \mathbf{0}_{n_s}$ was randomly chosen and 100 cases of s_{TRUE} were used for each point in this figure. Figure 1 suggests the following. (1) At $p = 0.5$, the possibility of correct estimation seriously deteriorates. This is because some states among possible states are not distinguishable at $p = 0.5$. Around $p = 0.5$, the estimation error becomes large. Therefore, judgment concerning observability is critical. (2) Observability is not achieved only at $p = 0.5$. However, estimation deterioration occurs around $p = 0.5$. When the number n_R of reports is small, the range of p at which estimation errors can occur becomes larger. Practically, caution is needed for estimation around the point that observability is not achieved, particularly when the number of reports is small. (3) The possibility of correct estimation for $n_B = 2$ is higher than that for $n_B = 1$. In particular, the ratio of correct estimation at $p = 0.5$ is sensitive to n_B. This is because any state in which one link is BAD cannot be distinguished among other such states. Thus, when $n_B = 1, p = 0.5$, the estimation method is equivalent to random choice among the possible states ($\neq \mathbf{0}_{n_s}$) and the ratio of correct estimation is about $1/n_s$. On the other hand, when $n_B = 2$ even if $p = 0.5$, a state $\mathbf{s}_2$ in which two links are BAD can be distinguished from any state

$\mathbf{s}_1$ in which one link is BAD. This is because $u_i(\mathbf{s}_2) = 1$ can occur for a certain i, but $u_i(\mathbf{s}_1) = 1$ never occurs for any i if $p \neq 0, 1$. In addition, we can show that any two states $\mathbf{s}_2, \mathbf{s}_2'$ in which two links are BAD can be distinguished even if $p = 0.5$. Consequently, for $n_B = 2$, if the true states are any states in which two links are BAD, the true state can be identified even if $p = 0.5$. Therefore, the ratio of correct estimation is better for $n_B = 2$ than for $n_B = 1$ at $p = 0.5$, although any state in which one link is BAD cannot be observable for $n_B = 2$. (4) The deterioration of the ratio of correct estimation at $p = 0.5$ becomes sharper as the number of nodes increases. This may be due to the total number of samples used increases as the number of nodes increases.

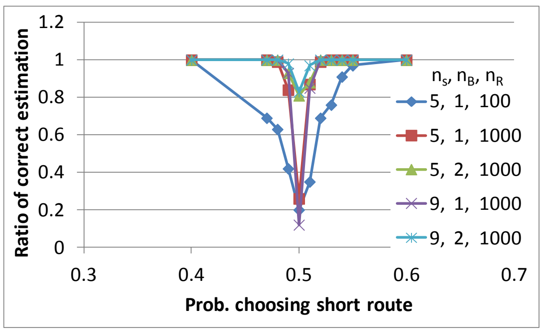

Fig. 1. Results for ring network

B. Network quality 2

Extend the example of network quality described in previous sections, and consider the network in Fig. 2. This network is a physical network and a part of a real commercial network in Japan. The total number of nodes is 45, and the total number of links is 54. Along the route from the source node to the destination node, there are one, two, or three rings. If the source and destination nodes are in different local rings and if these local rings have a common node, the route between the source and destination nodes passes through this common node without using the part of the main ring not belonging to these local rings. If the source and destination nodes are in different local rings and if these local rings do not have a common node, the route passes through the main ring.

There are n_r routes between each pair of source and destination nodes where n_r is $2, 4$, or 8 because there are two routes (clockwise and counter-clockwise) in each ring (note that $n_r = 2$ when two nodes are in the same ring). Assume that the shortest hop route among the n_r routes is chosen with probability $q_r(n_r, 1)$, the second shortest hop route among the n_r routes is chosen with probability $q_r(n_r, 2)$, $\cdots$, and the longest hop route among the n_r routes is chosen with probability $q_r(n_r, n_r)$. When the numbers of hops of the i-th and $(i+1)$-th shortest routes are equal, one of them is chosen as the i-th shortest route and the other is chosen as the $(i+1)$-th shortest route. In this example, $n_s = 54$, $n_c = 990$, and $n_a = 4516$.

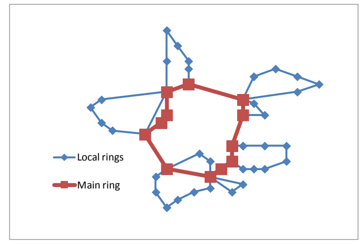

Fig. 2. Real network model

Similar to the previous subsection, we analyzed observability. The system is observable if $q_r(2, 1) \neq 0, 1/2$ and there is a pair of source and destination nodes with $n_r = 2$. The reason is similar to "Network quality 1." When $q_r(2, 1) = 1/2$, the system may not be observable. However, even when $q_r(2, 1) = 1/2$, observability depends on $q_r(n_r, i)$ $(n_r > 2)$. This is because the states that cannot be distinguished by two routes between any source and destination pair may be distinguished by n_r routes between another source and destination pair $(n_r > 2)$. For example, consider the distinguishability between "link-i is BAD" and "link-j is BAD" where links-i and j are in the same ring A. These two states cannot be distinguished by the two routes between a source and destination pair in ring A with $q_r(2, 1) = 1/2$. Consider a source in ring A and a destination in ring B and assume that there are four routes between them. The four routes are combinations of two routes in each ring, and if the probability of choosing a route among the four routes is the product of those in each ring, we cannot distinguish these states. This is because these four routes select these two links with the same probability. However, if we choose two routes using link-i with probability 0.7 and those using link-j with probability 0.3, we can distinguish "link-i is BAD" from "link-j is BAD".

To investigate the relationship between observability and estimation errors, we calculate $\mathbf{u}(\mathbf{s})$ and find $\mathbf{s}$ among the possible states to minimize $\min_{\mathbf{s}} |\mathbf{u}(\mathbf{s}) - \mathbf{r}_B|$ for the state estimation, similar in the previous subsection. Here, (1) $q_r(n_r, k)$ is given by $p^{n_r - k}(1-p)^{k-1}$ $(1 \leq k \leq n_r)$ for $n_r > 2$, (2) the set of possible states is 2^{n_s}, (3) the actual number of BAD links is 1 or 2, and (4) $n_R = 400$. When there was 1 BAD link, we tried every case in which one of the links was BAD. When there were 2 BAD links, we randomly generated 100 cases and used them for various $q_r(n_r, k)$.

Figure 3 shows that the ratio of successful estimation, which means that the estimated $\mathbf{s}$ is identical to the true $\mathbf{s}$, depends both on p and $q(2, 1)$. Observability is not achieved at the point $p = q_r(2, 1) = 1/2$. Although there is a huge number of parameters in real networks, it is likely that observability is not achieved at a few points determined by a limited number of parameters. Practically, however, estimation accuracy seriously deteriorated around that point. Of course, estimation accuracy depends on an estimation method. However, it is likely that

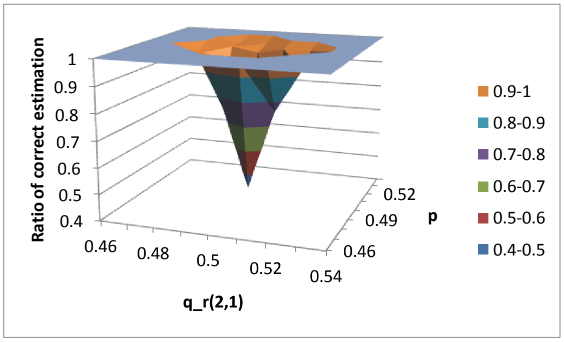

Fig. 3. Results for real network

there are serious deterioration areas of estimation accuracy in the parameter space.

C. Sensors detecting targets

Consider a grid plane $[1,\cdots,n_x]\times[1,\cdots,n_y]$. The sensing area of the class-k sensor located at (i,j) is assumed to be square $[i,i+L_k]\times[j,j+L_k]$. When a part of a sensing area is out of this grid area, we concentrate on the part within the grid area. Let $N_s(k)$ be the number of class-k sensors and $p_{k,(i,j)}$ be the probability that a class-k sensor is located at (i,j). The location of each sensor is assumed to be independent. The class-k sensing template set A_k is $\{\mathbf{a}_{k,(i,j)}\}_{i,j}$, where $1\le i\le n_x, 1\le j\le n_y$ and $\mathbf{a}_{k,(i,j)}$ correspond to the sensing area of the class-k sensor located at (i,j).

Assume that somewhere in $[1,\cdots,n_x]\times[1,\cdots,n_y]$ there is a single target object, which is a rectangle with unknown side lengths R_x, R_y. Also assume that the object moves within the grid area and each sensor senses the object at each time t. That is, the object located at $(i_0(t),j_0(t))$ occupies positions $[i_0(t),i_0(t)+R_x]\times[j_0(t),j_0(t)+R_y]$, where $(i_0(t),j_0(t))$ is also unknown and $1\le i_0(t)\le i_0(t)+R_x\le n_x$, $1\le j_0(t)\le j_0(t)+R_y\le n_y$.

If the object occupies the position (i,j), $(\mathbf{s})_{(i-1)n_y+j} = 1$; otherwise, $(\mathbf{s})_{(i-1)n_y+j} = 0$. In addition, $\mathbf{a}_{k,(i,j)}$ must be expressed as a row vector: $\mathbf{a}_{k,(i,j)} \equiv (\mathbf{0}_{(i-1)n_y+j-1}, \mathbf{1}_{L_k}, \mathbf{0}_{n_y-L_k}, \mathbf{1}_{L_k}, \mathbf{0}_{n_y-L_k}, \cdots, \mathbf{1}_{L_k}, \mathbf{0}_{n_y-L_k-j+1}, \mathbf{0}_{n_y(n_x-i-L_k+1)})$. The sensing template $\mathbf{a}_{k,(i,j)}$ is used with probability $p_{k,(i,j)}$. Define $\tilde{\mathbf{p}}_k = (p_{k,(1,1)}, p_{k,(1,2)},\cdots,p_{k,(1,n_y)},p_{k,(2,1)},\cdots,p_{k,(n_x,n_y)})$. Note that $n_s = n_x n_y, |c_k| = n_s$.

For simplicity, assume $L_1=\cdots=L_{n_c-1}=0$, $L_{n_c}=L$, and that the location of the target object is not near the rim of the grid area (that is, $i_0(t), j_0(t) > L$). Then, $\mathbf{a}_{k,(i,j)} = \mathbf{a}_{1,(i,j)}$ for $k=2,\cdots,n_c-1$ and for all i,j. Because $L_1=0$, $\mathbf{a}_{1,(i,j)} = \mathbf{e}_{(i-1)n_y+j}$. In addition, $\mathbf{a}_{n_c,(n_x,n_y)} = \mathbf{a}_{1,(n_x,n_y)}$ if $L>0$. (If $L=0$, $\mathbf{a}_{n_c,(i,j)} = \mathbf{a}_{1,(i,j)}$ for all i,j. In the remainder of this example, assume $L>0$ if explicitly indicated otherwise.) Therefore, $\{\mathbf{a}_1,\cdots,\mathbf{a}_{n_a}\}$ is given by $\{\mathbf{a}_{1,(1,1)}, \mathbf{a}_{1,(1,2)},\cdots,\mathbf{a}_{1,(1,n_y)},\mathbf{a}_{1,(2,1)},\cdots,\mathbf{a}_{1,(n_x,n_y)},\mathbf{a}_{n_c,(1,1)}, \mathbf{a}_{n_c,(1,2)},\cdots,\mathbf{a}_{n_c,(1,n_y)},\mathbf{a}_{n_c,(2,1)},\cdots,\mathbf{a}_{n_c,(n_x-1,n_y)},\mathbf{a}_{n_c,(n_x,1)}, \cdots,\mathbf{a}_{n_c,(n_x,n_y-1)}\}$, and $n_a = 2n_xn_y - 1$. As a result, $\mathbf{b}(\mathbb{B}(\mathbf{s}(t))) = \begin{pmatrix} {}^t\mathbf{s}(t) \\ \mathbf{a}_{n_c,(1,1)}\star\mathbf{s}(t) \\ \vdots \\ \mathbf{a}_{n_c,(n_x,n_y-1)}\star\mathbf{s}(t)\end{pmatrix}$ for any $\mathbf{s}(t)$, Note $\mathbf{p}_k = (\tilde{\mathbf{p}}_k, \mathbf{0}_{n_xn_y-1})$ for $k=1,\cdots,n_c-1$, and $\mathbf{p}_{n_c} = (\mathbf{0}_{n_y},\cdots,\mathbf{0}_{n_y},\mathbf{0}_{n_y-1},p_{n_c,(n_x,n_y)},p_{n_c,(1,1)},p_{n_c,(1,2)},\cdots, p_{n_c,(1,n_y)},p_{n_c,(2,1)},\cdots,p_{n_c,(n_x,n_y-1)})$. We should note that $\mathbf{b}(\mathbb{B}(\mathbf{s}(t))) \ne \mathbf{b}(\mathbb{B}(\mathbf{s}'(t)))$ for $\forall\mathbf{s}(t)\ne\mathbf{s}'(t)$ because $\mathbf{a}_{1,(i,j)} = \mathbf{e}_{(i-1)n_y+j}$.

On the other hand, for fixed $i_0(t), j_0(t), R_x$, and R_y, an obtained number $n_{k,1}(t)$ of class-k sensors detecting an object at t is given as follows. For $k<n_c$,

$$n_{k,1}(t) = N_s(k)\sum_{i=i_0(t)}^{i_0(t)+R_x}\sum_{j=j_0(t)}^{j_0(t)+R_y} p_{k,(i,j)} + \epsilon_{k,1}(t), \qquad (3)$$

and for $k=n_c$,

$$n_{k,1}(t) = N_s(k)\sum_{i=i_0(t)-L}^{i_0(t)+R_x}\sum_{j=j_0(t)-L}^{j_0(t)+R_y} p_{k,(i,j)} + \epsilon_{k,1}(t), \qquad (4)$$

where $\epsilon_{k,1}(t)$ is an error meaning the difference between the expectation of $n_{k,1}(t)$ and its observed value. Because there are four unknown parameters $i_0(t), j_0(t), R_x, R_y$, use four classes of sensors. That is, $n_c=4$.

To concretely consider this example, assume that $p_{1,(i,j)} = p_0(1)$, $p_{2,(i,j)} = p_0(2)+d_xi$, $p_{3,(i,j)} = p_0(3)+d_yj$, $p_{4,(i,j)} = p_0(4)$. The condition $\sum_{i=1}^{n_x}\sum_{j=1}^{n_y} p_{k,(i,j)} = 1$ must be satisfied. Thus, $p_0(1)=p_0(4)=1/n_s$, and $n_s(p_0(2)+d_x(n_x+1)/2) = n_s(p_0(3)+d_y(n_y+1)/2) = 1$.

It is clear that the rank of $P = \begin{pmatrix}\mathbf{p}_1\\ \vdots \\ \mathbf{p}_4\end{pmatrix}$ is four for $d_x\ne 0, d_y\ne 0, L>0$. If one of the parameters d_x, d_y, L is 0, one of the classes is linearly constructible: If $d_x=0$, class-2 is linearly constructible by other classes. Similarly, if $d_y=0$ ($L=0$), class-3 (class-4) is linearly constructible by other classes. This result is intuitive because, for example, class-2 sensors offer the same information as that from class-1 sensors when $d_x=0$. If we can control the random deployment of sensors, we should adopt $d_x>0, d_y>0, L>0$.

Furthermore, $P\mathbf{b}(\mathbb{B}(\mathbf{s}(t))) = \begin{pmatrix} \sum_{i=1}^{n_s}(\mathbf{s}(t))_i/n_s \\ p_0(2)\sum_{i=1}^{n_s}(\mathbf{s}(t))_i + d_x\sum_{i=1}^{n_x}\sum_{j=1}^{n_y} i(\mathbf{s}(t))_{(i-1)n_y+j} \\ p_0(3)\sum_{i=1}^{n_s}(\mathbf{s}(t))_i + d_y\sum_{i=1}^{n_x}\sum_{j=1}^{n_y} j(\mathbf{s}(t))_{(i-1)n_y+j} \\ \sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\mathbf{a}_{4,(i,j)}\star\mathbf{s}(t)/n_s\end{pmatrix}$. Therefore, if $\sum_{i=1}^{n_s}(\mathbf{s}(t))_i \ne \sum_{i=1}^{n_s}(\mathbf{s}'(t))_i$, $P\mathbf{b}(\mathbb{B}(\mathbf{s}(t))) \ne P\mathbf{b}(\mathbb{B}(\mathbf{s}'(t)))$. This means that we can distinguish $\mathbf{s}(t)$ from $\mathbf{s}'(t)$ if the target object size is different. On the other hand, even if $d_x, d_y\ne 0, L>0$, there exist $\mathbf{s}(t)$ and $\mathbf{s}'(t)$ that we cannot distinguish if $\sum_{i=1}^{n_s}(\mathbf{s}(t))_i = \sum_{i=1}^{n_s}(\mathbf{s}'(t))_i$. For example, we cannot distinguish the following two cases (if we do not know that the target object is a rectangle): the case in which the target object occupies two points $(L+1,L+1)$ and $(L+1,2L+3)$ but does not occupy any other states and the case in which it occupies $(L+1,L+2)$

and $(L+1, 2L+2)$ but does not occupy any other states. As a result, we cannot achieve observability when $S_s = \{\mathbb{B}^{-1}(0), \cdots, \mathbb{B}^{-1}(2^{n_s}-1)\}$.

If we limit the possible states to those satisfying the following two conditions, the conclusion for observability can be changed: (1) the target object is a single rectangle, (2) we know whether $R_x \geq R_y$ or $R_x \leq R_y$. By using the unknown parameters $i_0(t), j_0(t), R_x, R_y$, we can describe $P\mathbf{b}(\mathbb{B}(\mathbf{s}))$ for these possible states: $P\mathbf{b}(\mathbb{B}(\mathbf{s}(t))) = \begin{pmatrix} (1+R_x)(1+R_y)/n_s \\ (1+R_x)(1+R_y)\xi_x(i_0) \\ (1+R_x)(1+R_y)\xi_y(j_0) \\ (1+L+R_x)(1+L+R_y)/n_s \end{pmatrix}$, where $\xi_z(k) \equiv 1/n_s + d_z(2k(t)+R_z-n_z-1)/2$, $z \in \{x, y\}$, and $k = \{i_0, j_0\}$. (Note that, for $\mathbf{s}'(t) \neq \mathbf{s}(t)$, the vector $(i_0(t), j_0(t), R_x, R_y)$ is different.) Therefore, if and only if $d_x, d_y \neq 0, L > 0$, $P\mathbf{b}(\mathbb{B}(\mathbf{s}(t))) \neq P\mathbf{b}(\mathbb{B}(\mathbf{s}'(t)))$ because $(i_0(t), j_0(t), R_x, R_y)$ is uniquely determined by the given $P\mathbf{b}(\mathbb{B}(\mathbf{s}(t)))$. That is, observability is achieved.

The method for estimating object parameters is directly related to $P\mathbf{b}(\mathbb{B}(\mathbf{s}(t)))$ in this example. Equations (3) and (4) become ${}^t(n_{1,1}(t), \cdots, n_{4,1}(t)) = (N_s(1), \cdots, N_s(4))P\mathbf{b}(\mathbb{B}(\mathbf{s}(t))) + {}^t(\epsilon_{1,1}(t), \cdots, \epsilon_{4,1}(t))$. As a result, we can obtain the estimated parameters by the following equation.

$$\operatorname{argmin}_{\{i_0(t), j_0(t)\}_{1\leq t\leq T}, R_x, R_y} \sum_{k=1}^{n_c} \sum_{t=1}^{T} \epsilon_{k,1}^2(t) \qquad (5)$$

We conducted a simulation to investigate the relationship between parameter estimation and observability. The results are shown in Fig. 4, where $R_x = R_y = 10$, $n_x = n_y = 30$, $T = 10$, $N_s(i) = 100$ for $i = 1, 2, 3, 4$ and $d_x = d_y$. For each run of the simulation, sensors were randomly deployed according to the given $p_{k,(i,j)}$, which is a function of d_x and d_y. Here, $i_0(t)$ and $j_0(t)$ were also randomly generated according to the uniform probability distribution over $[1, n_x]$ and $[1, n_y]$, respectively. At the end of the simulation, parameters $R_x, R_y, \{i_0(t), j_0(t)\}_{1\leq t\leq T}$ were estimated according to Eq. (5), which uses the fact that there is a single rectangular area. One hundred runs were conducted to plot $(R_x - \hat{R}_x)^2 + (R_y - \hat{R}_y)^2$ and $\sum_{t=1}^{T}(i_0(t) - \hat{i}_0(t))^2 + (j_0(t) - \hat{j}_0(t))^2$ in Fig. 4, where $\hat{x}$ denotes the estimate of x.

As shown in Fig. 4, the square errors in the estimation of parameters R_x, R_y is insensitive to the values of d_x, d_y, and R_x, R_y can be estimated within reasonable estimation errors. On the hand, the square errors in the estimation of parameters $i_0(t), j_0(t)$ drastically increases as d_x, d_y approaches 0. This is because these parameters are not observable at $d_x, d_y = 0$. In other words, we can estimate the location of an object for $d_x, d_y \neq 0$ as well as the object parameters R_x, R_y.

VII. Conclusion

We theoretically analyzed the observability of a system monitored by distributed sensors and the independence of their sensing results. Because independence means the efficiency of the sensing system, the theoretical results on independence are useful when we design a monitoring system. In addition,

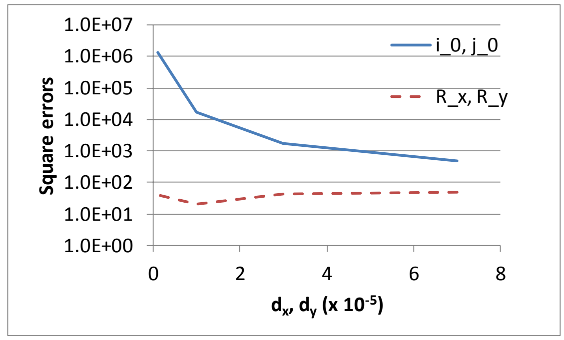

Fig. 4. Estimating R_x, R_y and locations of sensors

observability should be satisfied when we estimate the state of the system by using the sensing results. The definitions of observability and independence given in this paper cover a wider range of systems with monitoring mechanisms than in conventional literature, e.g., networks with end-to-end probing and systems where randomly deployed sensors are monitoring an object in a monitored area.

The theoretical results imply that observability is normally maintained almost everywhere in the parameter space if the system without a probabilistic mixture is observable. Independence is also normally achieved almost everywhere in the parameter space. Even under erroneous sensing, the results on observability and independence are valid. In addition, by limiting the set of possible states, we can make the system observable (see Theorem 2).

Practically, however, the estimation of the system may be difficult around areas in which observability is not achieved in the parameter space. This is because of the finite number of sensing samples.

References

[1] Janos Tapolcai, et al., Network-Wide Local Unambiguous Failure Localization (NWL-UFL) via Monitoring Trails, IEEE/ACM Trans. Networking, 20, 6, pp. 1762- 1773, 2012.

[2] Carmen Mas and Patrick Thiran, An Efficient Algorithm for Locating Soft and Hard Failures in WDM Networks, IEEE Journal on Selected Areas in Communications, 18, 10, pp. 1900-1911, 2000.

[3] R. Castro, et al., Network tomography: Recent developments, Statistical Science, 19, 3, pp. 499-517, 2004.

[4] F. Lo Presti, et al., Multicastbased inference of network-internal delay distributions, IEEE/ACM Trans. Networking, 10, 6, pp. 761- 775, 2002.

[5] A. Dhamdhere, et al., Netdiagnoser: troubleshooting network unreachabilities using end-to-end probes and routing data, ACM CoNEXT, pp. 1-12, 2007.

[6] A. Batsakis, et al., Practical passive lossy link inference, PAM, 2005.

[7] V. N. Padmanabhan, et al., Server-based Inference of Internet Link Lossiness, INFOCOM2003, pp. 145-155, 2003.

[8] M. Coates, et al., Internet tomography, IEEE Signal Process. Magazine, 19, 3, pp. 47–65, 2002.

[9] Jian Ni and Sekhar Tatikonda, Network Tomography Based on Additive Metrics, IEEE Trans. Inf. Theory, 57, 12, pp. 7798–7809, 2011.

[10] Aiyou Chen,et al., Network Tomography: Identifiability and Fourier Domain Estimation, IEEE Trans. Signal Processing, 58, 12, pp.6029–6039, 2010.

[11] R. Caceres, et al., Multicastbased inference of network-internal loss characteristics, IEEE Trans. Inf. Theory, 45, 7, pp. 2462–2480, 1999.

[12] N. G. Duffield, et al., Multicast topology inference from measured end-to-end loss, IEEE Trans. Inf. Theory, 48, 1, pp. 26–45, 2002.
[13] F. L. Presti, et al., Multicastbased inference of network-internal delay distributions, IEEE/ACM Trans. Networking, 10, 6, pp. 761–775, 2002.
[14] S. Ratnasamy et al., Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements, INFOCOM, 1999.
[15] Hongyi Zeng, et al., Automatic Test Packet Generation, CoNEXT, 2012.
[16] Hung X. Nguyen and Patrick Thiran, The Boolean Solution to the Congested IP Link Location Problem: Theory and Practice, INFOCOM 2007, pp. 2117 - 2125, 2007.
[17] Nick Duffield, Network Tomography of Binary Network Performance Characteristics, IEEE Trans. Inf. Theory, 52, 12, pp. 5373-5388, 2006.
[18] Hung X. Nguyen and Patrick Thiran, Active Measurement for Multiple Link Failures Diagnosis in IP Networks, C. Barakat and I. Pratt (Eds.): PAM 2004, LNCS 3015, pp. 185-194, 2004.
[19] Srinivas Devadas and Kurt Keutzer, Synthesis for Testability: A Brief Survey, IEEE International Symposium on Circuits and Systems, pp. 3097 – 3100, 1990.
[20] Sreejit Chakravarty and Harry B. Hunt III, On Computing Signal Probability and Detection Probability of Stuck-at Faults, IEEE Trans. Computers, 39, 11, pp. 1369–1377, 1990.
[21] H. Saito et al., Wide Area Ubiquitous Network: The Network Operator's View of a Sensor Network, IEEE Comm. Magazine, 46, 12, pp. 112–120, 2008.
[22] Masahiro Umehira, et al., Concept and Feasibility Study of Wide Area Ubiquitous Network for Sensors and Actuators, IEEE VTC 2007 Spring, pp.165-169, 2007.
[23] H. Saito, et al., Shape Estimation Using Networked Binary Sensors, INFOCOM 2009, pp. 2901 - 2905, 2009.
[24] H. Saito, et al., Estimating Size and Shape of Non-convex Target Object Using Networked Binary Sensors, IEEE SUTC 2010.
[25] H. Saito, et al., Parameter Estimation for Non-Convex Target Object Using Networked Binary Sensors, IEICE Trans. Inf. & Syst., E94-D, 4, pp.772-785, 2011.
[26] H. Saito, et al., Estimating Parameters of Multiple Heterogeneous Target Objects Using Composite Sensor Nodes, IEEE Trans. Mobile Computing, 11, 1, pp. 125-138, 2012.
[27] H. Saito, et al., Stochastic Geometric Filter and Its Application to Shape Estimation for Target Objects, IEEE Trans. Signal Processing, 59, 10, pp. 4971-4984, 2011.
[28] C. Commault, et al., Observability Preservation Under Sensor Failure, IEEE Trans. Automatic Control, 53 , 6, pp. 1554 - 1559, 2008.
[29] G. Hoblos, et al., Optimal design of fault tolerant sensor networks, Control Applications 2000, pp. 467 - 472, 2000.
[30] Yu Ru and C. N. Hadjicostis, Sensor Selection for Structural Observability in Discrete Event Systems Modeled by Petri Nets, IEEE Trans. Automatic Control, 55 , 8, pp. 1751 - 1764 , 2010.
[31] Mohammad Rezaeian, Sensor Scheduling for Optimal Observability Using Estimation Entropy, Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 307-312, 2007.
[32] Soummya Kar, et al., Distributed Parameter Estimation in Sensor Networks: Nonlinear Observation Models and Imperfect Communication, IEEE Trans. Inf. Theory, 58 , 6, pp. 3575 - 3605, 2012.

Appendix

Lemma 3: If $(\mathbf{b}(k))_i = (\mathbf{b}(k))_j$ for all $k = 2^l$ $(l = 0, \cdots, n_s - 1)$, $i = j$.

Proof: $(\mathbf{b}(k))_i = (\mathbf{b}(k))_j$ for $k = 2^l$ $(l = 0, \cdots, n_s - 1)$ means $(\mathbf{a}_i \star \mathbf{e}_1, \cdots, \mathbf{a}_i \star \mathbf{e}_{n_s}) = (\mathbf{a}_j \star \mathbf{e}_1, \cdots, \mathbf{a}_j \star \mathbf{e}_{n_s})$. Therefore, $\mathbf{a}_i = \mathbf{a}_j$. Thus, $i = j$. ■

Lemma 4: $\mathbf{b}(2^{n_s} - 1) = \mathbf{1}_{n_a}$

Proof: Because $\mathbf{a}_j \neq \mathbf{0}_{n_s}$, $\mathbf{a}_j \star (\mathbb{B}^{-1}(2^{n_s} - 1)) = \mathbf{a}_j \star \mathbf{1}_{n_s} = 1$ for all j. Thus, $\mathbf{b}(2^{n_s} - 1) = \mathbf{1}_{n_a}$. ■

Lemma 5: For all i, j, there exists an integer k such that $\mathbf{b}(k) = \mathbf{b}(i) \cup \mathbf{b}(j)$ when $S_s = 2^{n_s}$.

Proof: Define an n_a-element column vector $\mathbf{h}_i$ for $i = 1, \cdots, n_s$ by $(\mathbf{h}_1 \quad \cdots \quad \mathbf{h}_{n_s}) = \begin{pmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_{n_a} \end{pmatrix}$. Then, there exists an integer $t_{i,k} \in \{0, 1\}$ such that $\mathbf{b}(i) = \cup_{k=1}^{n_s} t_{i,k} \mathbf{h}_k$ because $\mathbf{b}(i) = \begin{pmatrix} \mathbf{a}_1 \star \mathbb{B}^{-1}(i) \\ \vdots \\ \mathbf{a}_{n_a} \star \mathbb{B}^{-1}(i) \end{pmatrix} = \begin{pmatrix} \cup_k t_{i,k} (\mathbf{a}_1)_k \\ \vdots \\ \cup_k t_{i,k} (\mathbf{a}_{n_a})_k \end{pmatrix}$. Here, $t_{i,k}$ is the k-th element of $\mathbb{B}^{-1}(i)$. Therefore, $\mathbf{b}(i) \cup \mathbf{b}(j) = \cup_{k=1}^{n_s} (t_{i,k} \cup t_{j,k}) \mathbf{h}_k$. Note that $t_{i,k} \cup t_{j,k} \in \{0, 1\}$. Because the set $\{\mathbf{b}(0), \cdots, \mathbf{b}(2^{n_s} - 1)\}$ includes any combination of $\cup$ of $\{\mathbf{h}_k\}_{k=1,\cdots,n_s}$, there exists an integer l such that $\mathbf{b}(l) = \mathbf{b}(i) \cup \mathbf{b}(j)$. ■

Lemma 6: For an n_a-element column vector $\mathbf{x}_i \equiv \mathbf{b}(k_1) \cap \cdots \cap \mathbf{b}(k_i)$, there exists an integer vector $\mathbf{n}(\mathbf{x}_i)$ such that $\mathbf{x}_i = B(2^{n_s}) \mathbf{n}(\mathbf{x}_i)$ for all i and $\forall k_1, \cdots, k_i \in \{0, \cdots, 2^{n_s} - 1\}$.

Proof: Define $D_i \equiv \{\mathbf{b}(k_1) \cap \cdots \cap \mathbf{b}(k_i) | \forall k_1, \cdots, k_i \in \{0, \cdots, 2^{n_s} - 1\}\}$. Prove that there exists a vector $\mathbf{n}(\mathbf{x}_i)$ for any $\mathbf{x}_i \in D_i$ such that $\mathbf{x}_i = B(2^{n_s}) \mathbf{n}(\mathbf{x}_i)$ by the mathematical induction on i by using Lemma 5. ■

Proof of Lemma 2: Define a set of integers $\Gamma(i) \equiv \{j | (\mathbf{b}(j))_i = 1, j \in \{0, \cdots, 2^{n_s} - 1\}\}$. Set $\Theta = \{1, \cdots, n_a\}$. Define $n_1(i) \equiv \sum_{1 \leq j \leq 2^{n_s}} (B(2^{n_s}))_{i,j}$, where $(X)_{i,j}$ is the (i, j)-element of a matrix X. Here, $n_1(i)$ is the number of "1" in the i-th row of the matrix $B(2^{n_s})$. Set $k_r = 1$. Note that $\Gamma(i) \neq \emptyset$ for any i because of Lemma 4.

(1) If there exists an integer $i \in \Theta$ such that $n_1(i) > n_1(j)$ for any $j \in \Theta - i$, obtain an integer column vector $\mathbf{n}_{k_r}$ such that $B(2^{n_s}) \mathbf{n}_{k_r} = \cap_{j \in \Gamma(i)} \mathbf{b}(j)$. The existence of $\mathbf{n}_{k_r}$ is due to Lemma 6. Note that $(\cap_{j \in \Gamma(i)} \mathbf{b}(j))_k = \begin{cases} 1, & \text{for } k = i \\ 0, & \text{for } k \in \Theta - i. \end{cases}$ This is because if $(\cap_{j \in \Gamma(i)} \mathbf{b}(j))_k = 1$ for $k \in \Theta - i$, $n_1(i) \leq n_1(k)$. This contradicts the assumption.

(2) If there exists an integer $i_1, i_2 \in \Theta$ such that $n_1(i_1) = n_1(i_2) > n_1(j)$ for any $j \in (\Theta - i_1) - i_2$, obtain an integer column vector $\mathbf{n}_k$ such that $B(2^{n_s}) \mathbf{n}_k = \cap_{j \in \Gamma(i_1)} \mathbf{b}(j)$. The existence of $\mathbf{n}_k$ is due to Lemma 6. Note that $(\cap_{j \in \Gamma(i_1)} \mathbf{b}(j))_k = \begin{cases} 1, & \text{for } k = i_1 \\ 0, & \text{for } k \in \Theta - i_1. \end{cases}$ This is because (i) if $(\cap_{j \in \Gamma(i_1)} \mathbf{b}(j))_k = 1$ for $k \in (\Theta - i_1) - i_2$, $n_1(i_1) \leq n_1(k)$. This contradicts the assumption; (ii) if $(\cap_{j \in \Gamma(i_1)} \mathbf{b}(j))_{i_2} = 1$, $n_1(i_1) < n_1(i_2)$ or $(\mathbf{b}(k))_{i_1} = (\mathbf{b}(k))_{i_2}$ for all k. Due to Lemma 3, $(\mathbf{b}(k))_{i_1} = (\mathbf{b}(k))_{i_2}$ for all k means $i_1 = i_2$. This contradicts the assumption.

For both (1) and (2), we can obtain $\mathbf{n}_{k_r}$, which $(B(2^{n_s}) \mathbf{n}_{k_r})_k = \begin{cases} 1, & \text{for } k = i \\ 0, & \text{for } k \in \Theta - i \end{cases}$ for $i \in \Theta$. Set $\Theta = \Theta - i$ and $k_r = k_r + 1$ and repeat until $\Theta = \emptyset$.

By using a matrix M_r for interchanging the order of rows,

$$M_r B(2^{n_s}) (\mathbf{n}_1 \quad \cdots \quad \mathbf{n}_{n_s} \quad *) = \begin{pmatrix} 1 & * & \cdots & * & * \cdots \\ 0 & 1 & \cdots & * & * \cdots \\ & & \ddots & & * \cdots \\ 0 & 0 & \cdots & 1 & * \cdots \end{pmatrix}$$

where the rank of the matrix of the right-hand side is n_a and that of the left-hand side is equal to or less than min(rank(M_r), rank($B(2^{n_s})$), rank($(\mathbf{n}_1 \quad \cdots \quad \mathbf{n}_{n_s} \quad *)$)). Therefore, rank$(B(2^{n_s})) \geq n_a$. On the other hand, rank$(B(2^{n_s})) \leq \min(n_a, 2^{n_s} - 1) = n_a$ because of Lemma 1. Hence, rank$(B(2^{n_s})) = n_a$.