Data Fusion with Minimal Communication

Zhi-Quan Luo, *Member, IEEE,* and John N. Tsitsiklis, *Member, IEEE*

***Abstract*—Two sensors obtain data vectors x and y, respectively, and transmit real vectors $\vec{m}_1(x)$ and $\vec{m}_2(y)$, respectively, to a fusion center. We obtain tight lower bounds on the number of messages (the sum of the dimensions of $\vec{m}_1$ and $\vec{m}_2$) that have to be transmitted for the fusion center to be able to evaluate a given function $\vec{f}(x,y)$. When the function $\vec{f}$ is linear, we show that these bounds are effectively computable. Certain decentralized estimation problems can be cast in our framework and are discussed in some detail. In particular, we consider the case where x and y are random variables representing noisy measurements and $\vec{f}(x,y) = E[z|x,y]$, where z is a random variable to be estimated. Furthermore, we establish that a standard method for combining decentralized estimates of Gaussian random variables has nearly optimal communication requirements.**

***Index Terms*—Decentralized estimation, distributed computation, data fusion, communication complexity, lower bounds.**

I. Introduction and Problem Formulation

Let there be two sensors, S_1 and S_2, respectively. Sensor S_1 (respectively, S_2) obtains a data vector $x \in \Re^m$ (respectively, $y \in \Re^n$). Sensor, S_1 (respectively, S_2) transmits to a fusion center a message $\vec{m}_1(x)$ [respectively, $\vec{m}_2(y)$]. Here, $\vec{m}_1$: $\Re^m \mapsto \Re^{r_1}$ and $\vec{m}_2$: $\Re^n \mapsto \Re^{r_2}$ are vector-valued functions that we call *message functions*. Finally, the fusion center uses the values of the received messages to evaluate a given function $\vec{f}$: $\Re^{m+n} \mapsto \Re^s$. For this to be possible, the received messages must contain enough information; in particular, the function $\vec{f}$ must admit a representation of the form

$$\vec{f}(x,y) = \vec{h}(\vec{m}_1(x), \vec{m}_2(y)), \qquad \forall (x,y) \in \mathscr{S}, \quad (1.1)$$

for some function $\vec{h}$: $\Re^{r_1+r_2} \mapsto \Re^s$. Here $\mathscr{S}$ is some subset of $\Re^{m+n}$ representing the set of all pairs (x,y) that are of interest. For example, we might have some prior knowledge that guarantees that all possible observation pairs (x,y) lie in $\mathscr{S}$. For reasons to be explained later, we also require the functions $\vec{m}_1$, $\vec{m}_2$, and $\vec{h}$ to be continuously differentiable. In the sequel, we will occasionally refer to the functions $\vec{m}_1$, $\vec{m}_2$, and $\vec{h}$ as a *communication protocol.*

Manuscript received October 10, 1991; revised November 5, 1993. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant OPG0090391 and by the Army Research Office under Contract DAAL03-86-K-0171.

Z.-Q. Luo is with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ont. L8S 4L7, Canada.

J. N. Tsitsiklis is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139.

IEEE Log Number 9405087.

The above described framework is a generic description of the process of data fusion. Data are collected at geographically distant sites and are transmitted, possibly after being compressed, to a fusion center. The fusion center needs these data for a specific purpose. No matter what this purpose is, it can be always modeled as the task of evaluating a particular function of the data. For example, suppose that x and y are random variables, representing noisy observations. Let z be a vector random variable to be estimated, and suppose that we wish the fusion center to compute the mean square estimate $E[z|x,y]$. Assuming that the joint probability distribution of (x,y,z) is known, $E[z|x,y]$ can be expressed as a function $\vec{f}(x,y)$, and we are back to the model introduced in the preceding paragraph.

From now on, we adopt the above framework. We assume that the function $\vec{f}$ and the data domain $\mathscr{S}$ are given. Our objective is to choose the message functions $\vec{m}_1$ and $\vec{m}_2$, in some desirable manner. An obvious solution to our problem is to let $\vec{m}_1(x) = x$ and $\vec{m}_2(y) = y$. This corresponds to a centralized solution whereby all available data are transmitted to the fusion center. However, if communication is costly, as it sometimes is, there could be an advantage if less information were transmitted. We may thus pose the problem of choosing the message functions $\vec{m}_1$ and $\vec{m}_2$ so as to minimize the number $r = r_1 + r_2$ of real-valued messages that are transmitted by the two sensors (recall that r_i is the dimension of the range of $\vec{m}_i$), subject to the constraint that $\vec{f}$ can be represented in the form (1.1). The minimal possible value of r will be called the *communication complexity* corresponding to $\vec{f}$ and $\mathscr{S}$ and will be denoted by $C_1(\vec{f}; \mathscr{S})$, where the subscript "1" denotes the fact that the functions $\vec{m}_i$ and $\vec{h}$ are assumed to be in C^1 (i.e., continuously differentiable) functions. We will also consider the cases where the functions $\vec{m}_1$, $\vec{m}_2$, and $\vec{h}$ are restricted to be *linear* or *analytic*. For these cases, we use $C_{\text{lin}}(\vec{f}; \mathscr{S})$ and $C_\infty(\vec{f}; \mathscr{S})$ to denote the corresponding communication complexity. Clearly, one has $C_1(\vec{f}; \mathscr{S}) \le C_\infty(\vec{f}; \mathscr{S}) \le C_{\text{lin}}(\vec{f}; \mathscr{S})$.

A couple of remarks about out model of communication are in order.

1) The assumption of continuous differentiability is introduced in order to eliminate some uninteresting communication protocols. For example, if no smoothness condition is imposed, then each sensor S_i can simply interleave the bits in the binary expansions of each component of its data vector and send

the resulting real number to the fusion center, thus sending a single real-valued message. Upon receiving this message, the fusion center can easily decode it and determine the value of x and y. Thus, with a total of two messages, the fusion center can recover the values of x, y and thus evaluate $\vec{f}(x, y)$. Such a communication protocol is not interesting since it basically amounts to sending all the information collected by the sensors to the fusion center. We are interested instead in a protocol that can somehow intelligently compress the information contained in the values of x, y and send to the fusion center only that information that is relevant to the evaluation of $\vec{f}(x, y)$. As we shall see later, the smoothness condition on the message functions succeeds in eliminating uninteresting communication protocols such as the one described above. The differentiability requirement on the function $\vec{h}$ of (1.1) is quite mild and not unnatural given the assumption that $\vec{m}_1$ and $\vec{m}_2$ are differentiable.

2) We have assumed that messages are real valued, in contrast to the digital communication often used in practice. Although such a continuous model of communication cannot be implemented exactly using digital devices, it is nonetheless a useful idealization for certain types of problems. For example, most (if not all) of the parallel and distributed numerical optimization algorithms are usually described and analyzed as if real numbers can be computed and transmitted exactly [5]. In addition, there is a fair body of literature in which data are communicated and combined for the purpose of obtaining a centralized optimal estimate [6], [7], [18], [19], [21]. This literature invariably assumes that real-valued messages are transmitted. The schemes proposed in these papers are often evaluated on the basis of the number of transmitted messages. However, there has been no work that tries to derive the *minimal* number of required messages, and this is where our contribution lies. Another motivation for using a continuous communication model, as opposed to the discrete-model often used in the theoretical computer science community [22], is that it opens the possibility of applying tools from analysis, algebra, and topology to the systematic study of communication complexity problems. It is also worth noting that a similar continuous framework has been successfully applied to the study of computational complexity [3], [4].

Our formulation of the data fusion problem can be regarded as an extension of the *one-way* communication complexity model first introduced and studied by Abelson [1]. In particular, Abelson considered the situation where two processors P_1, P_2 wish to compute some real-valued function $f(x, y)$ under the assumption that the value of x (respectively, y) is given only to P_1 (respectively, P_2) and that the messages (real valued) can be sent only from P_1 to P_2. Our setting has a similar flavor, except that we are dealing with a different "organizational structure." It is also worth noting that Abelson's model of continuous communication protocols has an interesting parallel in the field of mathematical economics; in the latter field, the problem of designing a communication protocol is formulated as a problem of designing a decentralized process that performs a desired economic function [8], [9], [16]. Subsequent to Abelson's initial work, there have been several other studies [2], [14], [15] of the communication complexity of various specific problems under more general continuous models of communication (e.g., allowing messages to be sent in both directions). The discrete counterpart of Abelson's formulation was introduced in [22] and was followed by many studies of the communication complexity of specific graph and optimization problems (e.g., [10], [11], [17], [20]).

This paper is organized as follows. In Section II, we consider the case where $\vec{f}$ is linear and $\mathcal{S}$ is a subspace of $\Re^{m+n}$, and we restrict ourselves to linear protocols. We motivate this problem in the context of decentralized estimation of Gaussian random variables, under the assumption that the statistics of the underlying random variables are commonly known. We obtain a complete characterization of the corresponding communication complexity $C_{\text{lin}}(\vec{f}; \mathcal{S})$, together with an effective algorithm for determining it. In the process of deriving these results, we solve a problem in linear algebra that could be of independent interest. In Section III, we extend the results of Section II to the case of a general nonlinear function $\vec{f}$ and general communication protocols. In particular, we show that for the case of decentralized Gaussian estimation, the restriction to linear message functions does not increase the communication complexity. In Section IV, we consider a variation of the Gaussian case treated in Section II. The main difference from Section II is that the covariance matrix of the observation noise at any particular sensor is assumed to be known by that sensor but not by the other sensor or the fusion center. We apply a result from Section III and obtain a fairly tight bound on the communication complexity. In particular, we show that a standard method for combining decentralized estimates has nearly optimal communication requirements. To the best of our knowledge, this is the first time that a result of this type appears in the estimation literature.

We shall adopt the following notational conventions throughout this paper. For any matrix M and N of size $l \times m$ and $l \times n$, respectively, we use $[M, N]$ to denote the matrix of size $l \times (m + n)$ whose columns are the columns of M followed by the columns of N. We let $r(M)$ be the rank of M, and M^T its transpose. For any differentiable function $f: \Re^{m+n} \mapsto \Re$ of two vector variables $x \in \Re^m$ and $y \in \Re^n$, we use the notation $\nabla_x f(x, y)$ [respectively, $\nabla_y f(x, y)$], to denote the m-dimensional (respectively, n-dimensional) vector whose components are the partial derivatives of f with respect to the components of x (respectively, y). If $\vec{f}: \Re^{m+n} \mapsto \Re^s$ is a vector function with component mappings $f_1, f_2, \cdots, f_s$, then $\nabla \vec{f}$ will denote its Jacobian matrix whose i-column is given by the gradient vector ∇f_i. Similarly, $\nabla_x \vec{f}$ (respectively, $\nabla_y \vec{f}$) will denote the matrix whose ith column is $\nabla_x f_i$ (respectively, $\nabla_y f_i$).

II. Decentralized Gaussian Estimation with Linear Messages

In this section, we consider a simple decentralized estimation problem in which all of the random variables involved are Gaussian and all the message functions are *linear*. We will give a complete characterization of the communication complexity for this problem, together with an effective method for computing it. The results in this section and the techniques developed for proving them will provide insight and motivation for the results in the next section, where the general nonlinear case will be considered.

Let $z \in \Re^l$ be a zero-mean Gaussian random variable with known covariance matrix P_{zz}. Suppose that the sensors S_1 and S_2 collect data about z according to the formulas

$$x = H_1 z + v_1, \tag{2.1}$$

$$y = H_2 z + v_2, \tag{2.2}$$

where H_1, H_2 are some $m \times l$ and $n \times l$ matrices, respectively, and $v_1 \in \Re^m$, $v_2 \in \Re^n$ are zero-mean Gaussian noise variables, not necessarily independent. Let R be the covariance matrix of (v_1, v_2). Suppose that the fusion center is interested in computing $\vec{f}(x, y) = \boldsymbol{E}[z|x, y]$, the conditional expectation of z given the observed values x and y. Assuming that v_1 and v_2 are independent of z, we have

$$\vec{f}(x, y) = \boldsymbol{E}[z|x, y] = F\begin{bmatrix} x \\ y \end{bmatrix}, \tag{2.3}$$

where F is a matrix satisfying

$$P_{zz} H^T = F\left[H P_{zz} H^T + R\right] \tag{2.4}$$

with

$$H = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix}. \tag{2.5}$$

If the inverse of $[H P_{zz} H^T + R]$ exists, then

$$\vec{f}(x, y) = \boldsymbol{E}[z|x, y] = P_{zz} H^T \left[H P_{zz} H^T + R\right]^{-1} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.6}$$

Suppose that the matrices P_{zz}, H_1, H_2, and R are known to both sensors S_1, S_2 as well as the fusion center. Then, (2.3)–(2.6) imply that

$$\vec{f}(x, y) = Ax + By, \tag{2.7}$$

for some matrices A, B (of size $l \times m$ and $l \times n$, respectively) depending on P_{zz}, R, H_1, and H_2. Notice that the matrices A and B can be regarded as constant since they can be precomputed and can be assumed to be available at the sensors and the fusion center.

The set $\mathscr{S}$ of possible data pairs is the support of the probability distribution of (x, y) and our Gaussian assumptions imply that it is a subspace of $\Re^{m+n}$. If the covariance matrix $H P_{zz} H^T + R$ of (x, y) is positive definite, then it is clear that $\mathscr{S} = \Re^{m+n}$. On the other hand, if this matrix is singular, then (x, y) takes values in a proper subspace of $\Re^{m+n}$, with probability 1. In other words, we have either

$$\mathscr{S} = \Re^{m+n} \tag{2.8}$$

or

$$\mathscr{S} = \{(x, y) | Cx + Dy = 0\}, \tag{2.9}$$

where C and D are some matrices of size $k \times m$ and $k \times n$, respectively, and k is some positive integer. The entries of C and D can be determined from P_{zz}, H_1, H_2, and R, so both C and D can be viewed as commonly known by the sensors and the fusion center.

Under the restriction to linear message functions, we have the following characterization of the communication complexity.

Theorem 2.1: Let $\vec{f}(x, y)$ and $\mathscr{S}$ be given by (2.7) and (2.8)–(2.9). Suppose that the matrices P_{zz}, H_1, H_2, and R are known to both sensors S_1, S_2 as well as the fusion center. We then have

$$C_{\text{lin}}\left(\vec{f}; \mathscr{S}\right) = \begin{cases} r(A) + r(B), & \text{if } \mathscr{S} = \Re^{m+n}, \\ \min_X \{r(A - XC) + r(B - XD)\}, & \\ \quad \text{if } \mathscr{S} = \{(x, y) | Cx + Dy = 0\}, \end{cases} \tag{2.10}$$

where the minimum is taken over all possible real matrices X of size $l \times k$, and where k is the number of rows of C and D.

Proof: Consider any communication protocol for computing $\vec{f}$ with linear message functions. Let $\vec{m}_1(x) = M_1 x$ and $\vec{m}_2(y) = M_2 y$ be the message functions used by sensor S_1 and S_2, respectively, where M_1 is a matrix of size $r_1 \times m$ and M_2 is a matrix of size $r_2 \times n$. (So, r_1 and r_2 are the number of messages sent to the fusion center from sensor S_1 and S_2, respectively.) By (1.1), there exists some final evaluation function $\vec{h}$ such that

$$\begin{aligned} Ax + By = \vec{f}(x, y) &= \vec{h}(\vec{m}_1(x), \vec{m}(y)) \\ &= \vec{h}(M_1 x, M_2 y), \qquad \forall (x, y) \in \mathscr{S}. \end{aligned} \tag{2.11}$$

We consider two cases.

Case 1: $\mathscr{S} = \Re^{m+n}$. By (2.11), $Ax + By$ is a function of $M_1 x$ and $M_2 y$. So, there holds

$$Ax + By = \vec{h}(0, 0) = \text{constant},$$

for all (x, y) satisfying

$$M_1 x = 0, \qquad M_2 y = 0, \qquad (x, y) \in \Re^{m+n}. \tag{2.12}$$

Thus, (x, y) must be orthogonal to the rows of the matrix $[A, B]$ whenever (x, y) satisfies (2.12). In other words, the null space of

$$\begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix}$$

is contained in the null space of $[A, B]$. It follows that there exist matrices N_1 and N_2 of appropriate dimensions such that

$$[A, B] = [N_1, N_2] \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix}.$$

Therefore, we have

$$A = N_1 M_1, \qquad B = N_2 M_2,$$

which further implies $r(A) \le r(M_1) \le r_1$ and $r(B) \le r(M_2) \le r_2$. Thus, $r = r_1 + r_2 \ge r(A) + r(B)$, which proves $C_{\text{lin}}(\vec{f}; \Re^{m+n}) \ge r(A) + r(B)$.

We now show that $C_{\text{lin}}(\vec{f}; \Re^{m+n}) \le r(A) + r(B)$ by constructing a communication protocol for computing $\vec{f}$ with $r(A) + r(B)$ linear message functions. This is accomplished as follows. By the singular value decomposition, A can be written as $A = EF$ for some matrices E and F of size $l \times r(A)$ and $r(A) \times m$, respectively. Furthermore, A is known to both sensors S_1, S_2 and to the fusion center. Thus, the decomposition $A = EF$ can be precomputed so that both the sensors and the fusion center know the value of E and F. Now let sensor S_1 use the message function $\vec{m}_1(x) = Fx$, which clearly takes $r(A)$ messages. Upon receiving the value of Fx, the fusion center can compute Ax by using the formula $Ax = E(Fx)$. By an identical argument, the value of By can also be computed with $r(B)$ linear messages from the sensor S_2. As a result, the fusion center can compute $\vec{f}(x, y) = Ax + By$ with a total of $r(A) + r(B)$ linear messages, which proves $C_{\text{lin}}(\vec{f}; \Re^{m+n}) \le r(A) + r(B)$, as desired.

Case 2: We now assume that $\mathscr{S} = \{(x, y) | Cx + Dy = 0\}$. Again, by (2.11), we have

$$Ax + By = \vec{h}(0, 0) = \text{constant},$$

for all $(x, y) \in \Re^{m+n}$ satisfying

$$M_1 x = 0, \qquad M_2 y = 0, \qquad Cx + Dy = 0.$$

Therefore, the null space of the matrix

$$\begin{bmatrix} C & D \\ M_1 & 0 \\ 0 & M_2 \end{bmatrix}$$

is contained in the null space of the matrix $[A, B]$. As a result there holds

$$[A, B] = [X, P_1, P_2] \begin{bmatrix} C & D \\ M_1 & 0 \\ 0 & M_2 \end{bmatrix},$$

for some matrices X, P_1, and P_2 of appropriate dimensions. Thus, we have

$$A = XC + P_1 M_1, \qquad B = XD + P_2 M_2,$$

which implies

$$r(A - XC) = r(P_1 M_1) \le r(M_1) \le r_1,$$

and

$$r(B - XD) = r(P_2 M_2) \le r(M_2) \le r_2.$$

Therefore, $r(A - XC) + r(B - XD) \le r_1 + r_2$, which implies $C_{\text{lin}}(\vec{f}; \mathscr{S}) \ge \min_X \{r(A - XC) + r(B - XD)\}$.

It remains to prove $C_{\text{lin}}(\vec{f}; \mathscr{S}) \le \min_X \{r(A - XC) + r(B - XD)\}$. To do this, fix any matrix X that attains the minimum in the expression $\min_X \{r(A - XC) + r(B - XD)\}$. Notice from (2.9) that $\vec{f}(x, y) = Ax + By = (A - XC)x + (B - XD)y$, for all $(x, y) \in \mathscr{S}$. By the singular value decomposition of $A - XC$ and $B - XD$ and using an argument similar to the one used in Case 1, we see that $(A - XC)x$ can be computed with $r(A - XC)$ linear messages from sensor S_1, and $(B - XD)y$ can be computed with $r(B - XD)$ linear messages from S_2; thus, $\vec{f}(x, y) = (A - XC)x + (B - XD)y$ is computable with a total of $r(A - XC) + r(B - XD)$ linear messages. By the choice of X, we have $C_{\text{lin}}(\vec{f}; \mathscr{S}) \le \min_X \{r(A - XC) + r(B - XD)\}$, as desired. Q.E.D.

Generically, the matrices A and B in the estimation equation (2.7) shall have full-row rank (that is, rank l) and the covariance matrix of (x, y) is nonsingular. Therefore, according to Theorem 2.1, a total of $2l$ (linear) messages are needed in order to enable the fusion center to make the desired estimation. (Recall that l is the dimension of the vector z being estimated.) The data fusion schemes that have been previously proposed in the literature [6], [7], [18], [19], [21] involve exactly $2l$ messages and are therefore generically optimal. In the event where either A or B is row-rank deficient, then a local data compression is possible and sending Ax and By is no longer optimal. We illustrate this point by the following simple example.

Example 2.1: Let

$$H_1 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \qquad H_2 = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

and suppose that z, v_1, v_2 are independent zero-mean Gaussian random vectors (in $\Re^2$) with autocovariance being equal to I. Consider the model

$$x = H_1 z + v_1$$
$$y = H_2 z + v_2.$$

Simple calculation shows that the covariance matrix of (x, y) is nonsingular and that

$$E[z | x, y] = Ax + By = \frac{1}{5}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} x + \frac{1}{5}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} y.$$

Therefore, according to Theorem 2.1, a total of two messages are needed to compute $E[z | x, y]$. Indeed, the sensors S_1, S_2 can simply send, respectively, the values $(1, 1)^T x$, $(1, 1)^T y$ to the fusion center. From the above formula, these two messages are clearly sufficient for the computation of $\boldsymbol{E}[z|x,y]$. Thus, sending Ax and By, which would take four messages, is not optimal.

In Example 2.1, the covariance matrix of (x, y) is nonsingular. Next, we consider a situation where the covariance matrix of (x, y) is singular.

Example 2.2: Let z_1, v_1, and v_2 be some independent zero-mean scalar Gaussian random variables with standard deviation equal to 1, and let $z_2 = -z_1$. Consider the data fusion model

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} v_1 \\ -v_2 \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} -v_1 \\ v_2 \end{bmatrix}.$$

It can be seen that the covariance matrix of (x_1, x_2, y_1, y_2) is singular; in fact, we have $\mathscr{S} = \{(x, y) \in \Re^4 | x_1 + x_2 + y_1 + y_2 = 0\}$ and $C = [1, 1]$ and $D = [1, 1]$. In this case, we use (2.3)–(2.5) to obtain

$$\boldsymbol{E}[z|x,y] = Ax + By = \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x + \frac{1}{2}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} y.$$

Thus, we have $r(A) + r(B) = 2 + 2 = 4$. Under normal circumstances (i.e., $\mathscr{S} = \Re^4$), four messages are required to fuse the data. However, due to the singularity of the covariance matrix of (x, y), a local data compression at both sensors is possible. Specifically, since $z_1 + z_2 = 0$ with probability 1, the knowledge of the value z_1 is sufficient for the fusion center to recover z_2; noting that $z_1 = (x_1 + y_1)/2$, we can simply let the sensors send x_1 and y_1 to the fusion center, which takes only two messages. On the other hand, one can verify directly that

$$\begin{aligned} &\min_X \{r(A - XC) + r(B - XD)\} \\ &= \min_X \left\{ r\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - X[1,1] \right) + r\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - X[1,1] \right) \right\} = 2, \end{aligned}$$

where the minimum is attained at $X = (1, 0)^T$. Thus, Theorem 2.1 implies that using two messages is optimal and no further data compression is possible.

Theorem 2.1 has provided a complete but nonconstructive characterization of the communication complexity of computing $\boldsymbol{E}[z|x,y]$ with linear message functions, for the Gaussian case. In order to turn Theorem 2.1 into a useful result, we show below that $C_{\text{lin}}(\vec{f}; \mathscr{S})$ and the minimizing matrix X in (2.10) are effectively computable (in polynomial time). The intuition behind this result can be drawn by considering the following two extreme cases. Suppose that $C = -D = I$. Then, $C_{\text{lin}}(\vec{f}; \mathscr{S}) = \min_X \{r(A - X) + r(B + X)\}$. Choosing $X = -B$, we see that $C_{\text{lin}}(\vec{f}; \mathscr{S}) \le r(A + B)$. On the other hand, using the inequality $r(A - X) + r(B + X) \ge r(A + B)$, for all X, we have $C_{\text{lin}}(\vec{f}; \mathscr{S}) \ge r(A + B)$, and, therefore, $C_{\text{lin}}(\vec{f}; \mathscr{S}) = r(A + B)$. For another extreme case, suppose that $D = 0$. Then, $C_{\text{lin}}(\vec{f}; \mathscr{S}) = r(B) + \min_X r(A + XC)$. Lemma 2.1 below shows that $\min_X r(A + XC)$ is computable in polynomial time. The proof of the polynomial computability of $C_{\text{lin}}(\vec{f}; \mathscr{S})$ in the general case is based on a combination of these techniques, as can be seen in the proof to follow.

Theorem 2.2: Suppose that the entries of the matrices A, B, C, and D are all rational numbers. Then, there is a polynomial time algorithm (in terms of the total sizes of the entries of A, B, C, and D) for computing $\min_X \{r(A - XC) + r(B - XD)\}$.

Remark: By transposing, we see that the minimization $\min_X \{r(A - CX) + r(B - DX)\}$ is also solvable in polynomial time.

Proof: The proof of Theorem 2.2 consists of a sequence of lemmas.

Lemma 2.1: For any rational matrices A and C, there exist square invertible matrices P and Q such that

a) P and Q depend only on C and are computable in polynomial time;
b) if $Y = XP$ and $\bar{A} = AQ^{-1}$, then $r(A + XC) = r([\bar{A}_1 + Y_1, \bar{A}_2])$, where Y_1 is a submatrix of Y given by a certain partition $Y = [Y_1, Y_2]$ of the columns of Y, and $[\bar{A}_1, \bar{A}_2]$ is a corresponding partition of $\bar{A}$ (partitioned in the same way as $Y = [Y_1, Y_2]$);
c) $\min_X r(A + XC) = r(\bar{A}_2)$. In particular, $\min_X r(A + XC)$ is computable in polynomial time.

Our next lemma, which is based on Lemma 2.1, reduces the original rank minimization problem to a simpler one.

Lemma 2.2: Let A, B, C, and D be some rational matrices. Then, there exist matrices $\tilde{A}_1$, $\tilde{A}_2$, $\tilde{B}_1$, and $\tilde{B}_2$ with

$$\min_X \{r(A - XC) + r(B - XD)\} = \min_W \left\{ r\left(\left[\tilde{A}_1 + W, \tilde{A}_2 \right] \right) + r\left(\left[\tilde{B}_1 + W, \tilde{B}_2 \right] \right) \right\}.$$

Moreover, the matrices $\tilde{A}_1$, $\tilde{A}_2$, $\tilde{B}_1$, and $\tilde{B}_2$ can be computed in polynomial time from A, B, C, and D.

The proofs of Lemmas 2.1 and 2.2 are lengthy and have been relegated to the Appendix.

Lemma 2.3: For any rational matrices A, B, C, and D, there holds.

$$\begin{aligned} &\min_X \{r([A + X, C]) + r([B + X, D])\} \\ &= \min_{Y,Z} r(A - B + CY + DZ) + r(C) + r(D), \quad (2.13) \end{aligned}$$

where the minimum is taken with respect to all matrices of the proper dimensions.

Proof: We first show that the left-hand side of (2.13) is no smaller than the right-hand side. To do this, we first notice

$$r([A + X, C]) = r([A + CY + X, C]), \qquad \forall Y, \quad (2.14)$$

$$r([B + X, D]) = r([B - DZ + X, D]), \qquad \forall Z. \quad (2.15)$$

Suppose that the minimum of $\min_X \{r([A+X,C]) + r([B+X,D])\}$ is attained at some X^*. Then, using the above relations, we have

$$\begin{aligned} &r([A+X^*,C]) + r([B+X^*,D]) \\ &\quad = r([A+CY+X^*,C]) + r([B-DZ+X^*,D]) \\ &\quad = r([A+CY+X^*,C]) \\ &\qquad + r([B-DZ^*+X^*,0]) + r(D) \\ &\quad \geq r([A-B+CY+DZ^*,C]) + r(D) \\ &\quad = r(A-B+CY^*+DZ^*) + r(C) + r(D) \\ &\quad \geq \min_{Y,Z} r(A-B+CY+DZ) + r(C) + r(D), \end{aligned}$$

where the second equality follows from choosing a Z^* so that the columns of D become perpendicular to the columns of $B - DZ^* + X^*$. (Such a Z^* can be found by solving for Z^* the system $D^T(B - DZ^* + X^*) = 0$. This system clearly has a solution when D has full rank. The case where D does not have full rank can be easily reduced to the full-rank case by throwing away some of the columns of D and letting the corresponding rows of Z^* be equal to zero.) The first inequality follows from the general matrix inequality $r(M) + r(N) \geq r(M+N)$, for all M and N; the third equality follows from choosing Y^* so that the columns of C are orthogonal to the columns of $A - B + CY^* + DZ^*$.

To show the other direction of the inequality, suppose that the minimum in the expression $\min_{Y,Z} r(A - B + CY + DZ)$ is attained at some matrices Y^* and Z^*. Then, using (2.14) and (2.15) we see that

$$\begin{aligned} &r([A+X,C]) + r([B+X,D]) \\ &\quad = r([A+CY^*+X,C]) + r([B-DZ^*+X,D]). \end{aligned}$$

Letting $X^* = -B + DZ^*$ and using the above relation, we obtain

$$\begin{aligned} &r([A+X^*,C]) + r([B+X^*,D]) \\ &\quad = r([A-B+CY^*+DZ^*,C]) + r(D) \\ &\quad \leq r(A-B+CY^*+DZ^*) + r(C) + r(D) \\ &\quad = \min_{Y,Z} r(A-B+CY+DZ) + r(C) + r(D), \end{aligned}$$

where the inequality follows from the general matrix inequality $r([M,N]) \leq r(M) + r(N)$, for all M and N; the last step is due to the definition of Y^* and Z^*. This complets the proof of (2.13). Q.E.D.

We are now ready to complete the proof of Theorem 2.2. By Lemma 2.2, it suffices to show the polynomial time computability of

$$\min_W \left\{ r\left(\left[\tilde{A}_1 + W, \tilde{A}_2\right]\right) + r\left(\left[\tilde{B}_1 + W, \tilde{B}_2\right]\right)\right\}$$

for some known (and polynomially computable) matrices $\tilde{A}_1$, $\tilde{A}_2$, $\tilde{B}_1$, and $\tilde{B}_2$. By Lemma 2.3, we only have to argue that $\min_{Y,Z} r(\tilde{A}_1 - \tilde{B}_1 + \tilde{A}_2 Y + \tilde{B}_2 Z)$ is computable in polynomial time [cf. (2.13)]. Letting

$$X = \begin{bmatrix} Y \\ Z \end{bmatrix},$$

we have

$$\begin{aligned} &\min_{Y,Z} r\left(\tilde{A}_1 - \tilde{B}_1 + \tilde{A}_2 Y + \tilde{B}_2 Z\right) \\ &\quad = \min_X r\left(\tilde{A}_1 - \tilde{B}_1 + \left[\tilde{A}_2, \tilde{B}_2\right] X\right) \\ &\quad = \min_X r\left(\tilde{A}_1^T - \tilde{B}_1^T + X^T\left[\tilde{A}_2, \tilde{B}_2\right]^T\right), \end{aligned}$$

and the result follows from Lemma 2.1(c). Q.E.D.

Theorem 2.2 provides a method for evaluating $C_{\text{lin}}(\vec{f}; \mathscr{S})$ [as given by (2.10)]. The proofs of Lemmas 2.1 and 2.2 (see the Appendix) show that the running time of this method is roughly equal to the running time of preforming several Gaussian eliminations and matrix inversions plus that of evaluating the rank of several matrices. It is still an open question whether there exist more efficient algorithms for computing $C_{\text{lin}}(\vec{f}; \mathscr{S})$. We also remark that the proof of Theorem 2.2 also provides us with a (polynomial time) algorithm for constructing a minimizing matrix X and a corresponding optimal communication protocol.

To close this section, we remark that Theorems 2.1 and 2.2 can be extended in a straightforward manner to the N sensor case. In particular, suppose $x^1 \in \Re^{n_1}, \cdots, x^N \in \Re^{n_N}$ are the observations obtained by the N sensors. Then, by the property of Gaussian random variables, there exist some matrices $A^1, \cdots, A^N$ such that

$$\vec{f}(x^1, \cdots, x^N) = E[z | x^1, \cdots, x^N] = A^1 x^1 + \cdots + A^N x^N.$$

Let $\mathscr{S}$ denote the support of the probability distribution of $(x^1, \cdots, x^N)$. Thus, $\mathscr{S} = \Re^{n_1 + \cdots + n_N}$, unless the covariance of $(x^1, \cdots, x^N)$ is singular in which case $x^1, \cdots, x^N$ are linearly related and

$$\mathscr{S} = \{(x^1, \cdots, x^N) | C^1 x^1 + \cdots + C^N x^n = 0\}$$

for some matrices $C^1, \cdots, C^N$. By an argument similar to that used in Theorems 2.1 and 2.2, we can show

$$C_{\text{lin}}\left(\vec{f}; \mathscr{S}\right) = \begin{cases} r(A^1) + \cdots + r(A^N), & \text{if } \mathscr{S} = \Re^{n_1 + \cdots + n_N}, \\ \min_X \{r(A^1 - XC^1) + \cdots + r(A^N - XC^N)\}, & \\ \qquad \text{if } \mathscr{S} = \{(x^1, \cdots, x^N) | C^1 x^1 + \cdots + C^N x^N = 0\}; & \end{cases}$$

Furthermore, the above minimum rank can be computed efficiently (i.e., in polynomial time).

III. The General Nonlinear Case

In this section, we consider the case where the function $\vec{f}$ is nonlinear and fairly arbitrary. Accordingly, we allow the message functions to be nonlinear as well. In terms of the decentralized estimation context, this is the situation that would arise if we were dealing with the optimal estimation of non-Gaussian random variables. We derive general lower bounds on the communication complexity for solving this problem. Our results imply that the lower bound of Theorem 2.1 remains valid even with general message functions. Thus, the restriction to linear message functions does not increase the communication complexity for the case of Gaussian random variables. We will also consider in this section the case of computing a *rational function* $\vec{f}(x, y)$ by using communication protocols whose message functions and final evaluation function are *analytic*. We will use some analytical tools to obtain an exact characterization of the communication complexity. This result will be used in Section IV, in our further analysis of decentralized Gaussian estimation.

In what follows, we assume that $\mathscr{G}$, the set of possible observation pairs for the two sensors S_1 and S_2, is described by

$$\mathscr{G} = \{(x, y) | \vec{g}_1(x, y) = 0, \vec{g}_2(x, y) \le 0; \quad x \in \Re^m, y \in \Re^n\}, \quad (3.1)$$

where $\vec{g}_1$: $\Re^{m+n} \mapsto \Re^{t_1}$ and $\vec{g}_2$: $\Re^{m+n} \mapsto \Re^{t_2}$ are some given differentiable functions, with t_1, t_2 some positive integers. When $\vec{g}_1 \equiv \vec{g}_2 \equiv 0$ we have $\mathscr{G} = \Re^{m+n}$, which corresponds to the case where the pair (x, y) is unrestricted.

Let $\vec{f}$: $\Re^{m+n} \mapsto \Re^s$ be a differentiable function of two vector variables x and y ($x \in \Re^m$, $y \in \Re^n$), and let $\vec{g} = (\vec{g}_1, \vec{g}_2)$. Our result is the following.

Theorem 3.1: Suppose that $\mathscr{G}$ [as defined by (3.1)] is nonempty. Suppose that either $\nabla\vec{g}(x)$ (the Jacobian of $\vec{g}$) has full rank for all $(x, y) \in \mathscr{G}$, or that $\vec{g}$ is a linear mapping. Then, for any $z = (x, y) \in \mathscr{G}$, we have

$$C_1(\vec{f}; \mathscr{G}) \ge \min_{Y \ge 0, X} \Big\{ r\Big(\nabla_x \vec{f}(z) - \nabla_x \vec{g}_1(z) X - \nabla_z \vec{g}_2(z) Y\Big) + r\Big(\nabla_y \vec{f}(z) - \nabla_y \vec{g}(z) X - \nabla_y \vec{g}_2(z) Y\Big)\Big\}. \quad (3.2)$$

Here, the minimum is taken over all matrices X and Y of appropriate dimensions, subject to the constraint that all entries of Y are nonnegative.

Proof: Consider any optimal communication protocol for computing $\vec{f}(x, y)$ over $\mathscr{G}$ (i.e., with a minimum number of messages). Let $\vec{m}_1$: $\Re^m \mapsto \Re^{r_1}$ and $\vec{m}_2$: $\Re^n \mapsto \Re^{r_2}$ be its message functions which are assumed to be continuously differentiable. Here, r_1 (respectively, r_2) is equal to the number of messages sent from sensor S_1 (respectively, S_2) to the fusion center. From equation (1.1), we have

$$\vec{f}(x, y) = \vec{h}(\vec{m}_1(x), \vec{m}_2(y)), \qquad \forall (x, y) \in \mathscr{G}, \quad (3.3)$$

where $\vec{h}$ is a continuously differentiable function. We need the following simple lemma.

Lemma 3.1: Let $\vec{p}$: $\Re^l \mapsto \Re^s$, $\vec{q}_1$: $\Re^l \mapsto \Re^{t_1}$, and $\vec{q}_2$: $\Re^l \mapsto \Re^{t_2}$ be three continuously differentiable functions. Suppose that the set $\mathscr{G} = \{z \in \Re^l | \vec{q}_1(z) = 0, \vec{q}_2(z) \le 0\}$ is nonempty and that $\vec{p}(z)$ is constant over $\mathscr{G}$. Let $\vec{q} = (\vec{q}_1, \vec{q}_2)$. If $\nabla\vec{q}(z)$ (the Jacobian of $\vec{q}$) has full rank for all $z \in \Re^l$, or if $\vec{q}$ is a linear mapping, then there exists a matrix function $A(z)$ of size $t_1 \times s$ and a matrix function $B(z) \ge 0$ (componentwise) of size $t_2 \times s$, so that $\nabla\vec{p}(z) = \nabla\vec{q}_1(z)A(z) + \nabla\vec{q}_2(z)B(z)$, for all $z \in \mathscr{G}$.

Proof: Let p_i be the ith component function of $\vec{p}$, $i = 1, \cdots, s$. Consider, for each i, the following constrained optimization problem:

$$\min_{z \in \mathscr{G}} p_i(z). \quad (3.4)$$

By assumption, each z satisfying $\vec{q}_1(z) = 0$ and $\vec{q}_2(z) \le 0$ is an optimal solution to (3.4). Since the regularity condition on the Jacobian of $\vec{q}$ or the linearity of $\vec{q}$ ensures the existence of a set of Lagrange multipliers, the necessary condition for optimality ([12, p. 300]) gives $\nabla p_i(z) = \nabla\vec{q}_1(z)a_i(z) + \nabla\vec{q}_2(z)b_i(z)$, for some vector function $a_i(z)$ of dimension t_1 and some vector function $b_i(z) \ge 0$ (componentwise) of dimension t_2, for all $i = 1, \cdots, s$. Writing these relations in matrix form yields the desired result. Q.E.D.

By (3.3), $\vec{f}(x, y) - \vec{h}(\vec{m}_1(x), \vec{m}_2(y)) = 0$ for all (x, y) satisfying $\vec{g}_1(x, y) = 0$ and $\vec{g}_2(x, y) \le 0$. Let $\vec{p}(x, y) = \vec{f}(x, y) - \vec{h}(\vec{m}_1(x), \vec{m}_2(y))$ and let $\vec{q}_1(x, y) = \vec{g}_1(x, y)$, $\vec{q}_2(x, y) = \vec{g}_2(x, y)$. Then, $\vec{p}$, $\vec{q}_1$ and $\vec{q}_2$ satisfy the assumptions of Lemma 3.1. Thus, there exist some matrix functions $Q_1(x, y)$ and $Q_2(x, y) \ge 0$ such that $\nabla\vec{p}(x, y) = \nabla\vec{q}_1(x, y)Q_1(x, y) + \nabla\vec{q}_2(x, y)Q_2(x, y)$, for all $(x, y) \in \mathscr{G}$. Equivalently, for all $(x, y) \in \mathscr{G}$, we have

$$\nabla\vec{f} - \begin{bmatrix} \nabla\vec{m}_1 & 0 \\ 0 & \nabla\vec{m}_2 \end{bmatrix} \begin{bmatrix} \nabla_{\vec{m}_1}\vec{h} \\ \nabla_{\vec{m}_2}\vec{h} \end{bmatrix} = \nabla\vec{g}_1 Q_1 + \nabla\vec{g}_2 Q_2.$$

Fix any $(x, y) \in \mathscr{G}$ and let $X = Q_1(x, y)$ and $Y = Q_2(x, y)$. The above relation implies that

$$\nabla\vec{m}_1 \nabla_{\vec{m}_1}\vec{h} = \nabla_x \vec{f} - \nabla_x \vec{g}_1 X - \nabla_x \vec{g}_2 Y, \quad (3.5)$$

$$\nabla\vec{m}_2 \nabla_{\vec{m}_2}\vec{h} = \nabla_y \vec{f} - \nabla_y \vec{g}_1 X - \nabla_y \vec{g}_2 Y. \quad (3.6)$$

Since $\nabla\vec{m}_1$ is a matrix of size $m \times r_1$, we obtain

$$\begin{aligned} r_1 &\ge r(\nabla\vec{m}_1) \\ &\ge r\Big(\nabla\vec{m}_1 \nabla_{\vec{m}_1}\vec{h}\Big) \\ &= r\Big(\nabla_x \vec{f} - \nabla_x \vec{g}_1 X - \nabla_x \vec{g}_2 Y\Big), \end{aligned} \quad (3.7)$$

where the last step is due to (3.5). Similarly, (3.6) yields

$$r_2 \ge r\Big(\nabla_y \vec{f} - \nabla_y \vec{g}_1 X - \nabla_y \vec{g}_2 Y\Big).$$

Therefore,

$$\begin{aligned} C_1(\vec{f}; \mathscr{G}) &= r_1 + r_2 \\ &\ge r\Big(\nabla_x \vec{f} - \nabla_x \vec{g}_1 X - \nabla_x \vec{g}_2 Y\Big) + r\Big(\nabla_y \vec{f} - \nabla_y \vec{g}_1 X - \nabla_y \vec{g}_2 Y\Big) \\ &\ge \min_{Y \ge 0, X} \Big\{ r\Big(\nabla_x \vec{f} - \nabla_x \vec{g}_1 X - \nabla_x \vec{g}_2 Y\Big) + r\Big(\nabla_y \vec{f} - \nabla_y \vec{g}_1 X - \nabla_y \vec{g}_2 Y\Big)\Big\}, \end{aligned}$$

for all $(x, y) \in \mathscr{S}$. This completes the proof of Theorem 3.1. Q.E.D.

We remark that when $\vec{f}$ and $\mathscr{S}$ are given by (2.7) and (2.8)–(2.9) then the right-hand side of (3.2) reduces to the right-hand side of (2.10). This implies that $C_{\text{lin}}(\vec{f}; \mathscr{S}) = C_1(\vec{f}; \mathscr{S})$. In other words, for the problem of estimating a Gaussian random variable, the restriction to the linear message functions *does not* increase the communication complexity. It is not clear how such a restriction on the message functions will affect the communication complexity for estimating general random variables.

A disadvantage of Theorem 3.1 is that it only provides a lower bound for the communication complexity $C_1(\vec{f}; \mathscr{S})$. It is not known in general how far away this lower bound can be from $C_1(\vec{f}; \mathscr{S})$. However, we show next that if $\vec{f}$ is a rational vector function, then we can obtain tight lower bounds in a local sense, for the class of analytic communication protocols (Theorems 3.2 and 3.3). We need to fix some notation.

Notation: Let $\vec{f} = (f_1, \cdots, f_s)$ be a (vector) rational function and let $\mathscr{D}$ (the domain of $\vec{f}$) denote the open subset of $\Re^{m+n}$ over which $\vec{f}$ is well defined (finite). For $i = 1, \cdots, s$, and for any n-tuple $\alpha = (\alpha_1, \cdots, \alpha_n)$ and m-tuple $\beta = (\beta_1, \cdots, \beta_m)$ of nonnegative integer indexes, we define the functions $f_i^\alpha: \mathscr{D} \mapsto \Re$ and $f_i^\beta: \mathscr{D} \mapsto \Re$ by letting

$$f_i^\alpha(x, y) = \frac{\partial^\alpha f_i}{\partial y_1^{\alpha_1} \partial y_2^{\alpha_2} \cdots \partial y_n^{\alpha_n}}(x, y),$$

$$f_i^\beta(x, y) = \frac{\partial^\beta f_i}{\partial x_1^{\beta_1} \partial x_2^{\beta_2} \cdots \partial x_m^{\beta_m}}(x, y). \quad (3.8)$$

(We use the convention $f_i^0 = f_i$.) Furthermore, a notation such as $\operatorname{span}\{\nabla_x f_i^\alpha(x, y): \forall y \in D_y, \forall i, \alpha\}$ will stand for the vector space spanned by all vectors of the form $\nabla_x f_i^\alpha(x, y)$ that are obtained as y varies in a set D_y and as i and α vary within their natural domains. Finally, for any finite index set I and collection $\{\alpha_i: i \in I\}$ of vectors, we use $[a_i: i \in I]$ to denote the matrix whose columns are given by the vectors $a_i \in I$.

Theorem 3.2: Let $\mathscr{D}$ denote the domain of $\vec{f}$. Let D_x and D_y be two nonempty subsets of $\Re^m$ and $\Re^n$, respectively, with $D_x \times D_y \subseteq \mathscr{D}$. Consider an analytic communication protocol, consisting a total of $r_1 + r_2$ messages, for computing $\vec{f}$ over $\mathscr{S} = D_x \times D_y$, where r_1 (respectively, r_2) denotes the number of messages sent to the fusion center from sensor S_1 (respectively, S_2). Then

$$r_1 \geq \max_{x \in D_x} \dim \operatorname{span}\{\nabla_x f_i^\alpha(x, y): \forall y \in D_y, \forall i, \alpha\}, \quad (3.9)$$

$$r_2 \geq \max_{y \in D_y} \dim \operatorname{span}\{\nabla_y f_i^\beta(x, y): \forall x \in D_z, \forall i, \beta\}. \quad (3.10)$$

Proof: Due to symmetry, we shall only prove (3.9). Let the message functions be denoted by $\vec{m}_1(x)$, $\vec{m}_2(y)$ and let the final evaluation function be denoted by $\vec{h}$. We then have from (1.1),

$$\vec{f}(x, y) = \vec{h}(\vec{m}_1(x), \vec{m}_2(y)), \qquad \forall (x, y) \in D_x \times D_y.$$

Fix any $x \in D_x$. Differentiating the above expression with respect to y yields

$$f_i^\alpha(x, y) = h_i^\alpha(\vec{m}_1(x), \vec{m}_2(y), y), \qquad \forall y \in D_y, \forall i, \quad (3.11)$$

where h_i^α is a suitable analytic function. We now differentiate both sides of (3.11) with respect to x to obtain

$$\nabla_x f_i^\alpha(x, y) = \nabla_x \vec{m}_1(x) \nabla_{\vec{m}1} h_i^\alpha(\vec{m}_1(x), \vec{m}_2(y), y), \qquad \forall y \in D_y, \forall i. \quad (3.12)$$

Thus, for all $y \in D_y$ and for all i, the vector $\nabla_x f_i^\alpha(x, y)$ is in the span of the columns of the matrix $\nabla_x \vec{m}_1(x)$. Since the number of columns of $\nabla_x \vec{m}_1(x)$ equals r_1, it follows that

$$r_1 \geq \dim \operatorname{span}\{\nabla_x f_i^\alpha(x, y): \forall y \in D_y, \forall i, \alpha\}.$$

Since the above relation holds for all $x \in D_x$, we see the validity of (3.9). Q.E.D.

It should be clear from the proof that Theorem 3.2 remains valid if the function $\vec{f}$ is merely analytic, rather than rational.

We continue with a corollary of Theorem 3.2 that will be used in the next section.

Corollary 3.1: Let $\vec{f}$ be a rational function with domain $\mathscr{D}$. For any analytic protocol that computes $\vec{f}$ over an open set $\mathscr{S} \subset \mathscr{D}$, the number of messages r_1 and r_2 transmitted by sensors S_1 and S_2, respectively, satisfy

$$r_1 \geq \operatorname{rank}[\nabla_x f_i^\alpha(x, y): i, \alpha], \qquad \forall (x, y) \in \mathscr{S}$$

$$r_2 \geq \operatorname{rank}[\nabla_y f_i^\alpha(x, y): i, \alpha]. \qquad \forall (x, y) \in \mathscr{S}.$$

Proof: Given $(x, y) \in \mathscr{S} \subset \mathscr{D}$, let D_x and D_y be some open sets containing (x, y) and such that $D_x \times D_y \subset \mathscr{S}$, and apply Theorem 3.2. Q.E.D.

We now provide a partial converse of Theorem 3.2 by showing that the lower bounds (3.9) and (3.10) are tight in a local sense.

Let $\vec{f}(x, y) = (f_1(x, y), f_2(x, y), \cdots, f_s(x, y))$ be a collection of rational functions to be computed by the fusion center, where $x \in \Re^m$, $y \in \Re^n$. Suppose that $f_i(x, y) = p_i(x, y)/q_i(x, y)$, $i = 1, \cdots, s$, where p_i and q_i are relatively prime polynomials. We assume that all of the p_i's and q_i's are nonzero polynomials. Note that each p_i and q_i ($i = 1, \cdots, s$) can be written in the form

$$p_i(x, y) = \sum_{\beta = (\beta_1, \cdots, \beta_n) \in \mathscr{B}} p_{i\beta}(x) y_1^{\beta_1} y_2^{\beta_2} \cdots y_n^{\beta_n},$$

$$q_i(x, y) = \sum_{\beta = (\beta_1, \cdots, \beta_n) \in \mathscr{B}} q_{i\beta}(x) y_1^{\beta_1} y_2^{\beta_2} \cdots y_n^{\beta_n}, \quad (3.13)$$

where each $p_{i\beta}$, $q_{i\beta}$ is a suitable polynomial and $\mathscr{B}$ is a finite set of n-tuples of nonnegative integers. Symmetrically, we can write

$$p_i(x, y) = \sum_{\alpha = (\alpha_1, \cdots, \alpha_m) \in \mathscr{A}} p_{i\alpha}(y) x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_m^{\alpha_m},$$

$$q_i(x, y) = \sum_{\alpha = (\alpha_1, \cdots, \alpha_m) \in \mathscr{A}} q_{i\alpha}(y) x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_m^{\alpha_m}, \quad (3.14)$$

where each $p_{i\alpha}$, $q_{i\alpha}$ is a suitable polynomial and $\mathscr{A}$ is a finite set of m-tuples of nonnegative integers. We let

$$t_x = \max_{x \in \Re^m} r\left[\nabla p_{i\beta}(x), \nabla q_{i\beta}(x) : 1 \le i \le s,\ \beta \in \mathscr{B}\right], \tag{3.15}$$

$$t_y = \max_{y \in \Re^n} r[\nabla p_{i\alpha}(y), \nabla q_{i\alpha}(y) : 1 \le i \le s,\ \alpha \in \mathscr{A}]. \tag{3.16}$$

Finally, let $\overline{D}_x$ and $\overline{D}_y$ be the (open) sets of points at which the maxima in (3.15) and (3.16), respectively, are attained. Our result is the following.

Theorem 3.3: Let $\vec{f}(x, y) = (p_1(x, y)/q_1(x, y), \cdots, p_s(x, y)/q_s(x, y))$ be a rational (vector) function ($x \in \Re^m$, $y \in \Re^n$), where p_i, q_i ($i = 1, \cdots, s$) are relatively prime polynomials. Let $\mathscr{D} = \{(x, y) | q_i(x, y) \neq 0,\ \forall i\}$. Suppose that $(0, 0) \in \mathscr{D}$ and that $p_i(0, 0) \neq 0$ for all i. Then, for any $(x, y) \in \mathscr{D} \cap (\overline{D}_x \times \overline{D}_y)$, there exists an open set $\mathscr{G}$ of the form $\mathscr{G} = D_x \times D_y$ containing (x, y) such that $D_x \times D_y \subset \mathscr{D} \cap (\overline{D}_x \times \overline{D}_y)$ and

$$t_x = \max_{x \in D_x} \dim \operatorname{span}\left\{\nabla_x f_i^{\alpha}(x, y) : \forall y \in D_y, \forall i, \alpha\right\}, \tag{3.17}$$

$$t_y = \max_{y \in D_y} \dim \operatorname{span}\left\{\nabla_y f_i^{\beta}(x, y) : \forall x \in D_x, \forall i, \beta\right\} \tag{3.18}$$

and

$$C_\infty\left(\vec{f}; \mathscr{G}\right) = t_x + t_y, \tag{3.19}$$

where t_x and t_y are defined by (3.15) and (3.16), respectively.

Proof: For notational simplicity, we shall prove (3.17), (3.18), and (3.19) only for the case $s = 1$ (i.e., $\vec{f}$ is a scalar rational function). We will thus omit the subscript i from our notation. The general case of $s \ge 1$ can be handled by modifying slightly the proof given below.

Fix some $(x^*, y^*) \in \mathscr{D} \cap (\overline{D}_x \times \overline{D}_y)$. Let $D_x \times D_y \subset \mathscr{D} \cap (\overline{D}_x \times \overline{D}_y)$ be an arbitrary open set containing (x^*, y^*). The validity of (3.17) and (3.18) follows directly from Theorem 3.3 of [13]. Theorem 3.2 yields

$$C_\infty\left(\vec{f}; D_x \times D_y\right) \ge t_x + t_y. \tag{3.20}$$

It only remains to show that (3.20) holds with equality. To do this, we need to construct an analytic communication protocol for computing $\vec{f}(x, y)$ over some open subset $D_x \times D_y$ of $\mathscr{D} \cap (\overline{D}_x \times \overline{D}_y)$ containing (x^*, y^*). Let $\mathscr{B}_1 \subset \mathscr{B}$ and $\mathscr{B}_2 \subset \mathscr{B}$ be such that $|\mathscr{B}_1| + |\mathscr{B}_2| = t_x$ and

$$r\left[\nabla p_\beta(x^*), \nabla q_{\beta'}(x^*) : \beta \in \mathscr{B}_1,\ \beta' \in \mathscr{B}_2\right] = t_x. \tag{3.21}$$

Similarly, we let $\mathscr{A}_1 \subset \mathscr{A}$ and $\mathscr{A}_2 \subset \mathscr{A}$ be such that $|\mathscr{A}_1| + |\mathscr{A}_2| = t_y$ and

$$r[\nabla p_\alpha(y^*), \nabla q_{\alpha'}(y^*) : \alpha \in \mathscr{A}_1,\ \alpha' \in \mathscr{A}_2] = t_y. \tag{3.22}$$

Consider the communication protocol with $\vec{m}_1(x) = \{p_\beta(x), q_{\beta'}(x):\ \beta \in \mathscr{B}_1,\ \beta' \in \mathscr{B}_2\}$ and with $\vec{m}_2(y) = \{p_\alpha(y), q_{\alpha'}(y):\ \alpha \in \mathscr{A}_1,\ \alpha' \in \mathscr{A}_2\}$. Clearly, the total number of messages used in this protocol is equal to $t_x + t_y$. We claim that this protocol can be used to compute $\vec{f}(x, y)$ over some open set $D_x \times D_y$ containing (x^*, y^*). We need the following lemma whose proof can be found in [15, Theorem A.1]. (In fact, Lemma 3.2 was proved in [15] only for continuously differentiable functions. But its proof easily generalizes to the analytic functions.)

Lemma 3.2: Let $\mathscr{Q}$ be an open subset of $\Re^r$. Let $F: \mathscr{Q} \mapsto \Re^s$ be an analytic mapping such that

$$\max_{z \in \mathscr{Q}} r(\nabla F(z)) = s.$$

Suppose that $f: \mathscr{Q} \mapsto \Re$ is an analytic function with property

$$\nabla f(z) \in \operatorname{span}\{\nabla F(z)\}, \qquad \forall z \in \mathscr{Q}.$$

Then, there exists some analytic function h such that $f(z) = h(F(z))$ for all $z \in \mathscr{Q}'$, where $\mathscr{Q}'$ is some open subset of $\mathscr{Q}$.

Consider the polynomial mapping F: $\Re^{m+n} \mapsto \Re^{t_x + t_y}$ defined by $F(x, y) = (\vec{m}_1(x), \vec{m}_2(y))$. Clearly, $\max_{x,y} r(\nabla F(x, y)) = t_x + t_y$. Moreover, we have

$$\begin{aligned}\nabla F(x, y) &= \begin{bmatrix} \nabla \vec{m}_1(x) & 0 \\ 0 & \nabla \vec{m}_2(y) \end{bmatrix} \\ &= \left[\begin{bmatrix} \nabla p_\beta(x) \\ 0 \end{bmatrix}, \begin{bmatrix} \nabla q_{\beta'}(x) \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \nabla p_\alpha(y) \end{bmatrix}, \begin{bmatrix} 0 \\ \nabla q_{\alpha'}(y) \end{bmatrix},\right. \\ &\qquad \left. : \beta \in \mathscr{B}_1,\ \beta' \in \mathscr{B}_2,\ \alpha \in \mathscr{A}_1,\ \alpha' \in \mathscr{A}_2 \right],\end{aligned} \tag{3.23}$$

for all x and y, where the second step follows from the definition of $\vec{m}_1(x)$ and $\vec{m}_2(y)$. On the other hand, by differentiating $f(x, y) = p(x, y)/q(x, y)$ and using (3.13) and (3.14), we obtain, for all (x, y) sufficiently close to (x^*, y^*), that

$$\begin{aligned}\nabla f(x, y) &\in \operatorname{span}\{\nabla p(x, y), \nabla q(x, y)\} \\ &= \operatorname{span}\left\{\begin{bmatrix} \nabla_x p(x, y) \\ \nabla_y p(x, y) \end{bmatrix}, \begin{bmatrix} \nabla_x q(x, y) \\ \nabla_y q(x, y) \end{bmatrix}\right\} \\ &\subset \operatorname{span}\left\{\begin{bmatrix} \nabla_x p(x, y) \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \nabla_y p(x, y) \end{bmatrix},\right. \\ &\qquad \left.\begin{bmatrix} \nabla_x q(x, y) \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \nabla_y q(x, y) \end{bmatrix}\right\} \\ &\subset \operatorname{span}\left\{\begin{bmatrix} \nabla p_\beta(x) \\ 0 \end{bmatrix}, \begin{bmatrix} \nabla q_\beta(x) \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \nabla p_\alpha(y) \end{bmatrix},\right. \\ &\qquad \left.\begin{bmatrix} 0 \\ \nabla q_\alpha(y) \end{bmatrix} : \beta \in \mathscr{B},\ \alpha \in \mathscr{A}\right\} \\ &= \operatorname{span}\left\{\begin{bmatrix} \nabla p_\beta(x) \\ 0 \end{bmatrix}, \begin{bmatrix} \nabla q_{\beta'}(x) \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \nabla p_\alpha(y) \end{bmatrix},\right. \\ &\qquad \left.\begin{bmatrix} 0 \\ \nabla q_{\alpha'}(y) \end{bmatrix}, : \beta \in \mathscr{B}_1, \beta' \in \mathscr{B}_2, \alpha \in \mathscr{A}_1, \alpha' \in \mathscr{A}_2\right\},\end{aligned}$$

where the last step follows from (3.15) and (3.16) and the fact that (3.21) and (3.22) hold for all (x, y) close to (x^*, y^*). This, together with (3.23), implies $\nabla f(x, y) \in \operatorname{span}\{\nabla F(x, y)\}$ for all (x, y) near (x^*, y^*). Thus, we can invoke Lemma 3.2 (with the correspondence $s \leftrightarrow t_x + t_y$ and $z \leftrightarrow (x, y)$) to conclude that there exists some analytic function $h: \Re^{t_x+t_y} \mapsto \Re$ such that $f(x, y) = h(F(x, y))$ for all (x, y) near (x^*, y^*). Since $F(x, y) = (\vec{m}_1(x), \vec{m}_2(y))$, we see that $f(x, y)$ can be computed on the basis of the messages $\vec{m}_1(x)$ and $\vec{m}_2(y)$ over some open subset D containing (x^*, y^*). Now take an open subset of D with the product form $D_x \times D_y$ such that $(x^*, y^*) \in D_x \times D_y$. This completes the proof of Theorem 3.2. Q.E.D.

In essence, Theorem 3.3 states that the lower bounds of Theorem 3.2 are tight, in a local sense, for the class of analytic communication protocols. The adjective "local" stands for the fact that the protocol of Theorem 3.3 works only for (x, y) in a possibly small open set $D_x \times D_y$. Theorem 3.3 also provides an alternative way of computing t_x and t_y [cf. (3.17) and (3.18)]. This is particularly useful since in some applications the computation of t_x and t_y as defined by (3.15) and (3.16), respectively, can be quite involved, whereas the computation of t_x and t_y as given by (3.17) and (3.18), respectively, is relatively simple; see the proof of Theorem 4.1 for an example.

We note that, instead of quoting the results of [13], we could have proved the lower bound $C_\infty(\vec{f}; D) \geq t_x + t_y$ directly from Theorem 3.1. However, such an approach is more complicated. Finally, note that Theorem 3.2 asserts the existence of a local analytic protocol with $t_x + t_y$ messages. Ideally, we would like to have a rational protocol (in which both the message functions and the final evaluation functions are rational, instead of analytic), which uses only $t_x + t_y$ messages, and which is global (in the sense that the domain $\mathscr{G}$ of the protocol coincides with the domain $\mathscr{D}$ of $\vec{f}$).

IV. Decentralized Gaussian Estimation Revisited

In this section, we consider a variation of the decentralized Gaussian estimation problem of Section II. In contrast to Section II, we will now assume that some of the statistics of the random variables involved are only locally known. We shall apply the results from Section III to obtain some tight bounds on the number of messages that have to be transmitted from the sensors to the fusion center and establish near optimality of a natural communication protocol.

Let $z \in \Re^m$ be an unknown Gaussian random variable to be estimated by the fusion center. Let there be two sensors S_1 and S_2 that are making observations of z according to

$$u = H_1 z + v_1, \tag{4.1}$$

$$w = H_2 z + v_2, \tag{4.2}$$

where $u \in \Re^n$ (respectively, $w \in \Re^n$) denotes the data vector observed by S_1 (respectively, S_2). Here, v_1 and v_2 are n-dimensional Gaussian noise vectors, independent of z and independent of each other. Also, H_1 and H_2 are two coefficient matrices of size $n \times m$. We note that, in practice, the number n of observations obtained by each sensor is typically much larger than the dimension m of the random variable z to be estimated. For this reason, we will be focusing on the case $n \geq m$. Finally, we have assumed in our model that the number of observations n made by each sensor is the same: this is no loss of generality, however, since one can add zero rows to one of the H matrices.

Let R_1 and R_2 be the covariance matrices of v_1 and v_2, respectively. Let P_{zz} be the covariance matrix of z, which we assume for simplicity to be positive definite. We assume that the fusion center wishes to compute the conditional expectation $E[z|u, w]$. Assuming the existence of the inverse in the equation below, we have

$$E[z|u, w] = P_{zz} H^T \left[H P_{zz} H^T + R \right]^{-1} \begin{bmatrix} u \\ w \end{bmatrix}, \tag{4.3}$$

where

$$H = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix}, \qquad R = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}. \tag{4.4}$$

Note that the invertibility of $H P_{zz} H^T + R$ is equivalent to assuming that the support of the distribution of (u, w) is all of $\Re^{2n}$. For this case, the results of Section II show that if the matrices P_{zz}, H_i, and R_i $(i = 1, 2)$ are commonly known by the sensors and the fusion center, then the communication complexity is equal to $2m$. Let us now assume that P_{zz} is known by the two sensors and the fusion center, while the matrices H_1, R_1 are known only to sensor S_1 and the matrices H_2, R_2 are known only to sensor S_2. This case can be quite realistic. For example, the coefficients of H_1 might be determined locally and on-line by sensor S_1 (as would be the case if sensor S_1 were running an extended Kalman filter). Also, the entries of R_1 might be estimated locally and on-line by sensor S_1, by computing the empirical variance or autocorrelation of past observations. In both of the cases described above, the values of H_1 and R_1 would be known only by sensor S_1.

In relation to the notation used earlier in the present paper, we have $x = (H_1, R_1, u)$, $y = (H_2, R_2, w)$, and the function to be computed by the fusion center is

$$\begin{aligned} \vec{f}(x, y) &= \vec{f}(H_1, R_1, u; H_2, R_2, w) \\ &= P_{zz} H^T \left[H P_{zz} H^T + R \right]^{-1} \begin{bmatrix} u \\ w \end{bmatrix}, \end{aligned} \tag{4.5}$$

where H and R are given by (4.4). Note that P_{zz} does not appear as an argument in the left-hand side of (4.5), because it is considered as a commonly known constant.

Finally, we let $\mathscr{G}$ be the set of all $(H_1, R_1, u, H_2, R_2, w)$ such that R_1 and R_2 are symmetric positive definite matrices. (Note that on $\mathscr{G}$, the matrix $H P_{zz} H^T + R$ is guaranteed to be invertible.)

Theorem 4.1: Let $\vec{f}$ and $\mathscr{G}$ be as above and assume that $n \geq m$. Then,

$$m^2 + m \leq C_\infty(\vec{f}; \mathscr{G}) \leq m^2 + 3m.$$

Proof: The upper bound follows from well-known formulas for the combining of measurements. We repeat the argument here for the sake of completeness. As is well known (see e.g., [6], [7]), we have

$$E[z|u,w] = \left[P_{zz}^{-1} + H_1^T R_1^{-1} H_1 + H_2^T R_2^{-1} H_2\right]^{-1} \cdot \left[H_1^T R_1^{-1} u + H_2^T R_2^{-1} w\right]. \quad (4.6)$$

This suggests the following protocol. Sensor S_1 transmits $H_1^T R_1^{-1} x$ (m messages) and $H_1^T R_1^{-1} H_1$ to the fusion center. The latter is a symmetric matrix of dimension $m \times m$. Since the off-diagonal entries need only be transmitted once, $m(m+1)/2$ messages suffice. The situation for sensor S_2 is symmetrical, and we see that the total number of transmitted messages is equal to $m^2 + 3m$. It is clear from (4.6) that these messages enable the fusion center to compute $E[z|u,w]$.

We continue with the proof of the lower bound. To keep notation simple, we will only prove the lower bound for the case $m = n$. The argument for the general case ($n \geq m$) is very similar. In any case, it should be fairly obvious that increasing the value of n, while keeping the value of m constant, cannot decrease the communication complexity. (A formal proof is omitted.) Thus, any lower bound established for the case $n = m$ is valid for the case $n \geq m$ as well.

A further simplification of the proof is obtained by considering the special case where $P_{zz} = I$. As long as $m = n$, and P_{zz} is positive definite, any decentralized estimation problem can be brought into this form, by performing an invertible coordinate transformation to the vector z. Thus, this assumption results in no loss of generality.

Let

$$A = \begin{bmatrix} H_1 H_1^T + R_1 & H_1 H_2^T \\ H_2 H_1^T & H_2 H_2^T + R_2 \end{bmatrix}$$

and note that

$$\vec{f} = \left[H_1^T, H_2^T\right] A^{-1} \begin{bmatrix} u \\ w \end{bmatrix}.$$

We will now evaluate $\nabla_{R_2} \vec{f}$ at $(H_2, R_2) = (I, 0)$. Note that $\nabla_{R_2} \vec{f}$ is a matrix of size $m(m+1)/2 \times m$, because R_2 has $m(m+1)/2$ independent entries. A typical row of this matrix, denoted by $\partial \vec{f} / \partial R_2(i,j)$, contains the partial derivatives of $\vec{f}$ with respect to a simultaneous change of the (i,j)th and the (j,i)th entry of R_2. We now use the formula $\nabla A^{-1} = -A^{-1} \nabla A A^{-1}$ to see that the transpose of

$$\left.\frac{\partial \vec{f}}{\partial R_2(i,j)}\right|_{\substack{H_2 = I \\ R_2 = 0}} \quad (4.7)$$

(which is an m-dimensional column vector) is equal to

$$-\left[H_1^T, I\right] A^{-1} \begin{bmatrix} 0 & 0 \\ 0 & E_{ij} \end{bmatrix} A^{-1} \begin{bmatrix} u \\ w \end{bmatrix}. \quad (4.8)$$

Here, E_{ij}, for $i \neq j$ denotes the $m \times m$ matrix all of whose entries are zero except for its (i,j)th and (j,i)th entries, which are equal to 1. For $i = j$, E_{ii} has all zero entries, except for the (i,i)th entry, which is equal to 1. It is now easily verified that

$$A^{-1}\Big|_{\substack{H_2 = I \\ R_2 = 0}} = \begin{bmatrix} H_1 H_1^T + R_1 & H_1 \\ H_1^T & I \end{bmatrix}^{-1} = \begin{bmatrix} R_1^{-1} & -R_1^{-1} H_1 \\ -H_1^T R_1^{-1} & I + H_1^T R_1^{-1} H_1 \end{bmatrix}. \quad (4.9)$$

Taking (4.9) into account, (4.8) becomes, after some algebra,

$$E_{ij} H_1^T R_1^{-1} u - E_{ij}(I + H_1^T R_1^{-1} H_1) w.$$

We differentiate once more, this time with respect to w, and obtain

$$\nabla_w \left(\frac{\partial \vec{f}}{\partial R_2(i,j)} \right)\Bigg|_{\substack{H_2 = I \\ R_2 = 0}} = -E_{ij}(I + H_1^T R_1^{-1} H_1).$$

In terms of the notation used in Section III, each entry of the matrix $-E_{ij}(I + H_1^T R_1^{-1} H_1)$ corresponds to a function of the form f_i^α. Our objective being to apply Corollary 3.1, we will now compute the gradient of a typical entry of $E_{ij}(I + H_1^T R_1^{-1} H_1)$, with respect to the variables of sensor S_1. More precisely, we only take the gradient with respect to R_1. (By not taking the gradient with respect to some of the components of x, we are essentially deleting some of the rows of the matrix $[\nabla_x f_i^\alpha(x,y)\colon i, \alpha]$, and this cannot increase the rank of that matrix.) In fact, it is more convenient to represent this gradient as an $m \times m$ symmetric matrix, rather than a vector of dimension $m(m+1)/2$, with the entries in the upper triangular part corresponding to the components of the gradient. It is then understood that the rank in Corollary 3.1 will be computed in the vector space of $m \times m$ symmetric matrices.

The (p,q)th entry of $E_{ij}(I + H_1^T R_1^{-1} H_1)$ can be written as $e_p^T E_{ij}(I + H_1^T R_1^{-1} H_1) e_q$, where e_p and e_q are the pth and qth unit vectors, respectively. We then use the formula $\nabla_A x^T A y = xy^T + yx^T$ to evaluate the gradient of the above expression, with respect to R_1, at the point $H_1 = R_1 = I$. (The correct formula is only xy^T. We get the symmetric form because ∇_A stands for derivative in the direction of a_{ij} and a_{ji} simultaneously). The result is seen to be $-(e_q e_p^T E_{ij} + E_{ij} e_p e_q^T)$. Note that $e_q e_i^T E_{ij} + E_{ij} e_i e_q^T = E_{qj}$, if $i \neq j$, and $e_q e_i^T E_{ij} + E_{ij} e_i e_q^T = E_{qj} + E_{jq}$, if $i = j$. Therefore, the matrices $e_q e_p^T E_{ij} + E_{ij} e_p e_q^T$ span the vector space of symmetric matrices, which is of dimension $m(m+1)/2$. In the notation of Corollary 3.1, the rank of $[\nabla_x f_i^\alpha(x,y)\colon i, \alpha]$ evaluated at $x^* = (H_1, R_1, u) = (I, I, u)$

and $y^* = (H_2, R_2, w) = (I, 0, w)$ is at least $m(m+1)/2$.

The desired lower bound now follows from Corollary 3.1, except for the minor difficulty that $(x^*, y^*) \notin \mathscr{S}$. (This is because in $\mathscr{S}$ we have required R_2 to be positive definite.) However, an easy continuity argument shows that the rank of $[\nabla_x f_i^\alpha(x,y): i, \alpha]$ remains at least $m(m+1)/2$ in an open set around (x^*, y^*). Thus, we can apply Corollary 3.1 to a point in the vicinity of (x^*, y^*) that belongs to $\mathscr{S}$, and the proof is complete. Q.E.D.

We should emphasize here that the decentralized estimation problem considered by Theorem 4.1 is a *nonlinear* one, even though the random variables involved are all Gaussian. Recall that in the centralized setting the optimal estimate of a Gaussian random variable is always linear. However, in the decentralized setting this is no longer the case as the matrices H_i, R_i, $i = 1, 2$, are known only locally at each sensor, thus making the optimal estimate (4.5) to be computed by the fusion center a nonlinear one. For such a nonlinear decentralized estimation problem, what Theorem 4.1 asserts is far from obvious: at least $m^2 + m$ messages are needed to compute the optimal Gaussian estimate (4.5). This suggests that the well-known data fusion scheme (4.6), which uses $m^2 + 3m$ linear messages, has nearly optimal communication requirements.

As discussed in [18], the decentralized estimation formula (4.6) corresponds to the data fusion scheme, whereby each sensor computes a local optimal estimate using its own data and then the fusion center combines these two local estimates using a linear transformation. It is known [18] that this fusion scheme is valid only when the noise covariance matrix R is block diagonal. In particular, if the measurement noises are correlated, then the locally optimal estimates no longer carry enough information for the computation of the globally optimal estimate. In other words, $m^2 + 3m$ messages will not be enough in this case.

Finally, we mention some possible extensions of Theorem 4.1. First, we note that there is a gap of $2m$ messages between the lower and upper bounds provided by Theorem 4.1. Although for large m this gap is small compared to the quadratic lower bound, it is still of interest to close it. Also, we have, for the sake of simplicity, only stated and proved the result for the two sensor case. We expect the result (and the proof) of Theorem 4.1 to hold for the general N sensor case (just like Theorems 2.1 and 2.2).

V. Discussion

In this paper, we considered the problem of minimizing the amount of communication in decentralized estimation. When the random variables involved are Gaussian, we have obtained some tight bounds on the number of messages that have to be communicated in order for a fusion center to perform statistically optimal estimation. Our results may provide useful insight and guidelines to design communication protocols for the decentralized estimation problems when the communication resource is scarce. While the paper was focused on static estimation problems, it might be interesting to consider extensions to decentralized Kalman filtering problems.

Appendix

Proof of Lemma 2.1: Using Gaussian elimination, we can write

$$C = P\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} Q,$$

where P and Q are some invertible square matrices. Clearly, P and Q are computable in polynomial time. Let $\bar{A} = AQ^{-1}$ and $Y = XP$. We then have

$$\begin{aligned} r(A + XC) &= r\left(A + XP\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} Q\right) \\ &= r\left(AQ^{-1} + XP\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\right) \\ &= r\left(\bar{A} + Y\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\right) \\ &= r(\bar{A} + [Y_1, 0]) \\ &= r\left(\left[\bar{A}_1 + Y_1, \bar{A}_2\right]\right), \end{aligned} \tag{A.1}$$

where we have partitioned Y into $[Y_1, Y_2]$ so that

$$Y\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} = [Y_1, 0]$$

and have partitioned $\bar{A} = [\bar{A}_1, \bar{A}_2]$ accordingly. This proves part (b). Since P is invertible and $Y = [Y_1, Y_2] = XP$, we have from (A.1)

$$\begin{aligned} \min_X r(A + XC) &= \min_{Y_1} r\left(\left[\bar{A}_1 + Y_1, \bar{A}_2\right]\right) \\ &= r(\bar{A}_2), \end{aligned} \tag{A.2}$$

where the last step follows by taking $Y_1 = -\bar{A}_1$. Since Q can be computed in polynomial time, we see that $\bar{A}_2$ is also computable in polynomial time, which further implies that $r(\bar{A}_2)$ can be evaluated in polynomial time. Combining this with (A.2) yields part (c). Q.E.D.

Proof of Lemma 2.2: First, by Lemma 2.1, there exists some linear transformation $Y = XP$ under which

$$r(A + XC) = r([A'_1 + Y_1, A'_2]), \tag{A.3}$$

where P and A' are two matrices computable in polynomial time and $A' = [A'_1, A'_2]$ is partitioned according to the partition $Y = [Y_1, Y_2]$. Under the same linear transformation $Y = XP$, we have

$$\begin{aligned} r(B + XD) &= r(B + Y\hat{D}) \\ &= r\left(B + Y_1\hat{D}_1 + Y_2\hat{D}_2\right), \end{aligned} \tag{A.4}$$

where

$$\hat{D} = P^{-1}D \quad \text{and} \quad \hat{D} = \begin{bmatrix} \hat{D}_1 \\ \hat{D}_2 \end{bmatrix}$$

is partitioned according to $Y = [Y_1, Y_2]$. Using Lemma 2.1 again (with the correspondence $B + Y_1\hat{D}_1 \leftrightarrow A$, $\hat{D}_2 \leftrightarrow C$ and $Y_2 \leftrightarrow X$), we obtain

$$\min_{Y_2} r\left(B + Y_1\hat{D}_1 + Y_2\hat{D}_2\right) = r(B' + Y_1 D'), \tag{A.5}$$

where B' and D' are some matrices computable in polynomial time from B, $\hat{D}_1$, and $\hat{D}_2$.

Combining (A.3) and (A.4), we obtain

$$\min_X \{r(A + CX) + r(B + XD)\}$$
$$= \min_{Y_1, Y_2} \left\{ r([A'_1 + Y_1, A'_2]) + r\left(B + Y_1\hat{D}_1 + Y_2\hat{D}_2\right) \right\}$$
$$= \min_{Y_1} \left\{ r([A'_1 + Y_1, A'_2]) + \left(\min_{Y_2} r\left(B + Y_1\hat{D}_1 + Y_2\hat{D}_2\right)\right) \right\}$$
$$= \min_{Y_1} \{r([A'_1 + Y_1, A'_2]) + r(B' + Y_1 D')\}, \qquad \text{(A.6)}$$

where the last step follows from (A.5).

To complete the proof, we apply Lemma 2.1 to $r(B' + Y_1 D')$. In particular, there exists a linear transformation $W = Y_1 P'$ (P' is an invertible matrix computable in polynomial time) such that

$$r(B' + Y_1 D') = r\left(\tilde{B}_1 + W_1, \tilde{B}_2\right) \qquad \text{(A.7)}$$

for some matrices $\tilde{B}_1$ and $\tilde{B}_2$ which are computable in polynomial time from B' and D'. Here, $W = [W_1, W_2]$ is some partition of the matrix W. Under the same linear transformation $W = Y_1 P'$, we have

$$r([A'_1 + Y_1, A'_2]) = r\left(\left[A'_1 + W(P')^{-1}, A'_2\right]\right)$$
$$= r([A'_1 P' + W, A'_2])$$
$$= r([A'_1 P' + [W_1, W_2], A'_2]),$$

where the second step follows from the invertibility of P'. Thus, we have

$$\min_{W_2} r([A'_1 + Y_1, A'_2]) = \min_{W_2} r([A'_1 P' + [W_1, W_2], A'_2])$$
$$= r([(A'_1 P')_1 + W_1, 0, A'_2])$$
$$= r\left(\left[\tilde{A}_1 + W_1, \tilde{A}_2\right]\right), \qquad \text{(A.8)}$$

where the second step follows from choosing $W_2 = (A'_1 P')_2$, which is obtained by partitioning $A'_1 P' = [(A'_1 P')_1, (A'_1 P')_2]$ according to the partition $W = [W_1, W_2]$; and in the last step we have let $\tilde{A}_1 = (A'_1 P')_1$ and $\tilde{A}_2 = [0, A'_2]$. We now combine (A.7) with (A.8) to obtain

$$\min_{Y_1} \{r([A'_1 + Y_1, A'_2]) + r(B' + Y_1 D')\}$$
$$= \min_{W_1, W_2} \left\{ r([A'_1 + Y_1, A'_2]) + r\left(\left[\tilde{B}_1 + W_1, \tilde{B}_2\right]\right) \right\}$$
$$= \min_{W_1} \left\{ \left(\min_{W_2} r([A'_1 + Y_1, A'_2]) \right) + r\left(\left[\tilde{B}_1 + W_1, \tilde{B}_2\right]\right) \right\}$$
$$= \min_{W_1} \left\{ r\left(\left[\tilde{A}_1 + W_1, \tilde{A}_2\right]\right) + r\left(\left[\tilde{B}_1 + W_1, \tilde{B}_2\right]\right) \right\}.$$

This, together with (A.6), implies

$$\min_X \{r(A - XC) + r(B - XD)\}$$
$$= \min_X \{r(A + XC) + r(B + XD)\}$$
$$= \min_{W_1} \left\{ r([\tilde{A}_1 + W_1, A'_2]) + r\left(\left[\tilde{B}_1 + W_1, \tilde{B}_2\right]\right) \right\},$$

as desired. Q.E.D.

References

[1] H. Abelson, "Towards a theory of local and global computation," *Theoret. Comput. Sci.*, vol. 6, pp. 41–67, 1978.

[2] ——, "Lower bounds on information transfer in distributed computations," *J. Assoc. Comput. Mach.*, vol. 27, no. 2, pp. 384–392, 1980.

[3] A. Borodin and I. Munro, *The Computational Complexity of ALgebraic and Numeric Problems*. New York: American Elsevier, 1975.

[4] L. Blum, M. Shub, and S. Smale, "On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines," *Bull. Amer. Math. Soc.*, vol. 21, no. 1, pp. 1–47, 1989.

[5] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[6] C. Y. Chong, "Hierarchical estimation," in *Proc. 2nd MIT/ONR Workshop Command, Control, Commun.*, Monterey, CA, 1979.

[7] H. R. Hashemipour, R. Roy, and A. J. Laub, "Decentralized structures for parallel Kalman filtering," *IEEE Trans. Automat. Contr.*, vol. AC-33, pp. 88–94, 1988.

[8] L. Hurwicz, "Optimality and informational efficiency in resource allocation processes," in *Mathematical Methods in the Social Sciences, 1959*, K. J. Arrow, S. Karlin and P. Suppes, Eds. Stanford, CA: Stanford Univ., 1960, pp. 27–48.

[9] ——, "On informational decentralization and efficiency in resource allocation," in *Studies in Mathematical Economics*, J. Reiter, Ed. Math. Assoc. Amer., 1985.

[10] J. Ja'Ja, "The VLSI complexity of selected graph problems," *J. Assoc. Comput. Mach.*, vol. 31, no. 2, pp. 377–391, 1984.

[11] R. J. Lipton and R. Sedgewick, "Lower bounds for VLSI," in *Proc. 13th STOC*, 1981, pp. 300–307.

[12] D. G. Luenberger, *Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley, 1984.

[13] Z.-Q. Luo, "Communication complexity of computing a collection of rational functions," in *Advances in Computing and Information*, S. G. Akl and W. W. Koczkodaj, Eds. New York: Springer-Verlag, 1990, pp. 453–462.

[14] Z.-Q. Luo and J. N. Tsitsiklis, "On the communication complexity of distributed algebraic computation," *J. Assoc. Comput. Mach.*, vol. 40, no. 5, Nov. 1993, pp. 1019–1047.

[15] ——, "On the communication complexity of solving a polynomial equation," *SIAM J. Comput.*, vol. 20, pp. 936–950, 1991.

[16] K. Mount and S. Reiter, "The informational size of message spaces," *J. Econom. Theory*, pp. 161–192, 1974.

[17] C. H. Papadimitriou and M. Sipser, "Communication complexity," in *Proc. 14th STOC*, 1982, pp. 196–200.

[18] S. Roy and R. A. Iltis, "Decentralized linear estimation in correlated measurement noise," *IEEE Trans. Aerospace Electron. Syst.*, vol. 27, no. 6, pp. 939–941, Nov. 1991.

[19] J. L. Speyer, "Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem," *IEEE Trans. Automat. Contr.*, vol. AC-24, pp. 266–269, Apr. 1979.

[20] J. N. Tsitsiklis and Z.-Q. Luo, "Communication complexity of convex optimization," *J. Complexity*, vol. 3, pp. 231–243, 1987.

[21] A. S. Willsky, M. G. Bello, D. A. Castanon, B. C. Levy, and G. C. Verghese, "Combining and updating of local estimates and regional maps along sets of one-dimensional tracks," *IEEE Trans. Automat. Contr.*, vol. AC-27, pp. 799–813, Aug. 1982.

[22] A. C. Yao, "Some complexity questions related to distributed computing," in *Proc. 11th STOC*, 1979, pp. 209–213.