Distributed Estimation*

Yuzhe Xu[†1], Vijay Gupta[‡2], and Carlo Fischione[§1]

[1]School of Electrical Engineering, ACCESS Linnaeus Center, KTH Royal Institute of Technology, SE-100 44,Stockholm, Sweden
[2]Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN-46556, USA

September 17, 2012

Abstract

Distributed estimation plays an essential role in many networked applications, such as communication, networked control, monitoring and surveillance. Motivated by this, the chapter provides an overview on some of the fundamental aspects of distributed estimation over networks together with an investigation of the computational complexity and communication cost. A phenomenon being observed by a number of sensors in networks having a star and a general topology are considered. Under the assumptions of noises and linear measurements, the resulting distributed estimators are derived respectively. The limited bandwidth, communication range and message loss in the communication are considered. Distributed estimators can provide accurate estimates of the parameters of the phenomenon, while the less the limitations are in networks, the lower complexity of the estimator is.

Contents

*This work is supported by the Swedish Research Council and by the EU projects STREP Hydrobionets and NoE Hycon2

[†]yuzhe@kth.se
[‡]vgupta2@nd.edu
[§]carlofi@kth.se

1 Notation

Given a stochastic variable x, let $\mathbb{E}\,[x]$ denote its expected value, while $\mathrm{Var}[x] = \mathbb{E}\,[x - \mathbb{E}\,[x]]^2$ is its variance. With $\mathbb{E}\,_x y(x)$ we mean that the expected value is taken with respect to the probability density function (pdf) $p_x(\cdot)$ of x, where y is some function of the random variable x. Given a set of K nodes at time $n = 0, 1, 2, \ldots$, let $x_{n,k}$ denote the variable x from k-th node at time n for all $k = 1, 2, \ldots, K$. Furthermore, let X_n denote the vector $[x_{n,1}, x_{n,2}, \ldots, x_{n,K}]^T \in \mathbb{R}^K$ at time n. With $\hat{x}$ we denote the estimate of the random variable x. With $\|\cdot\|$ we denote the ℓ^2-norm of a vector or the spectral norm of a matrix. Given a matrix A, $\ell_m(A)$ and $\ell_M(A)$ denote the minimum and maximum eigenvalue (with respect to the absolute value of their real part), respectively, and its largest singular value is denoted by $\gamma(A)$. If A is a square matrix, we use $\mathrm{tr}(A)$ denote the trace of the matrix A, the sum of the elements on its diagonal. Suppose the matrix B having same size of A, $A \circ B$ is the Hadamard (element-wise) product between A and B. With $A^\dagger$ we denote the Moore-Penrose pseudo-inverse of the matrix A. With $a \preceq b$ and $a \succeq b$ denote the element-wise inequalities. With $\mathbf{I}$ and $\mathbf{1}$ we denote the identity matrix and the vector $(1, \ldots, 1)^T$, respectively, whose dimensions are clear from the context.

2 Network with a Star Topology

In this section, we assume that the network is organized as a star, where multiple sensors make measurements that are transmitted with no messages losses to a fusion center, which is assumed to be the star of the network. An example is illustrated by Fig. 1.

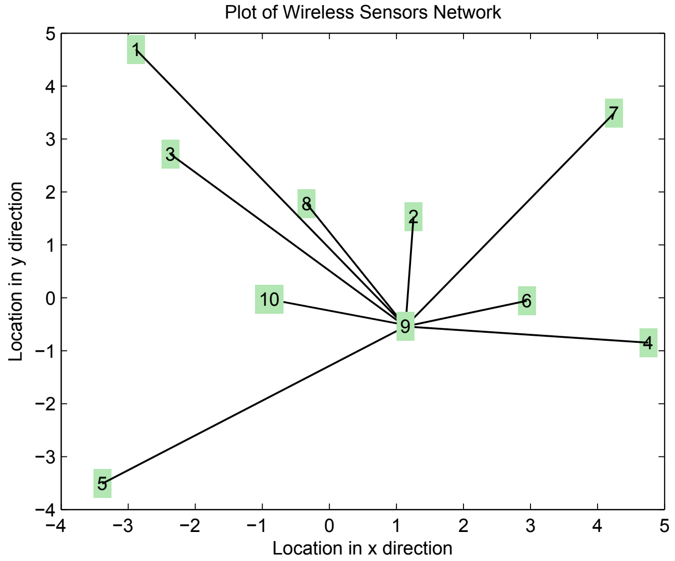

Figure 1: An example of star topology network with nodes and links (solid lines indicating that there is message communication between nodes). In this network, Node 9 can receive information from all other nodes. Thus Node 9 is the central unit.

2.1 Static Sensor Fusion

Here we study the problem of estimating a static phenomenon that is observed by a number of sensors. The observations of these sensors are then reported to a central unit that fuses them with the aim of extracting an estimate of higher accuracy.

2.1.1 Combining Estimators

In this subsection, we study distributed Minimum Mean Square Estimators (MMSE). In Appendix A.1, we recall the general result of MMSE for centralized linear estimators. Here, we rewrite those results in an alternative form.

Proposition 2.1. *Let $y = Hx + v$, where H is a matrix and v is a zero mean Gaussian noise with covariance matrix R_V independent of X. Then the MMSE estimate of X given $Y = y$ is*

$$P^{-1}\hat{x} = H^T R_V^{-1} y,$$

with P is the corresponding error covariance given by

$$P^{-1} = \left(R_X^{-1} + H^T R_V^{-1} H\right).$$

Proof. The expression for P follows by applying the matrix inversion lemma in Appendix A.2. For the estimate, consider

$$\begin{aligned}
P^{-1}\hat{x} &= \left(R_X^{-1} + H^T R_V^{-1} H\right) R_X H^T \left(H R_X H^T + R_V\right)^{-1} y \\
&= H^T \left(H R_X H^T + R_V\right)^{-1} y + H^T R_V^{-1} H R_X H^T \left(H R_X H^T + R_V\right)^{-1} y \\
&= H^T R_V^{-1} \left(H R_X H^T + R_V\right) \left(H R_X H^T + R_V\right)^{-1} y \\
&= H^T R_V^{-1} y.
\end{aligned}$$

□

This alternate form is useful because it combines local estimates directly without recourse to sending all the measurements to a central data processing unit that runs a giant estimator. This is called static sensor fusion.

2.1.2 Static Sensor Fusion for Star Topology

Proposition 2.2. *Consider a random variable x being observed by K sensors that generate measurements of the form*

$$y_k = H_k x + v_k, \qquad k = 1, \cdots, K,$$

where the noises v_k are all uncorrelated with each other and with the variable x. Denote the estimate of x based on all the n measurements by $\hat{x}$ and the estimate of x based only on the measurement y_k by $\hat{x}_k$. Then $\hat{x}$ can be calculated using

$$P^{-1}\hat{x} = \sum_{k=1}^{K} P_k^{-1}\hat{x}_k,$$

where P is the estimate error covariance corresponding to $\hat{x}$ and P_k is the error covariance corresponding to $\hat{x}_k$. Further

$$P^{-1} = \sum_{k=1}^{K} P_k^{-1} - (K-1)R_X^{-1}.$$

Proof. Denote y as the stacked vector of all the measurements y_k's, H the corresponding measurement matrix obtained by stacking all the H_k's and v the noise vector obtained by stacking all the noises v_k's. The global estimate $\hat{x}$ is given by

$$P^{-1}\hat{x} = H^T R_V^{-1} y.$$

But all the v_k's are uncorrelated with each other. Hence R_V is a block diagonal matrix with blocks R_{V_k}. Thus the right hand side can be decomposed as

$$H^T R_V^{-1} y = \sum_{k=1}^{K} H_k^T R_{V_k}^{-1} y_k.$$

But each of the terms $H_k^* R_{V_k}^{-1} y_k$ can be written in terms of the local estimates

$$P_k^{-1}\hat{x}_k = H_k^T R_{V_k}^{-1} y_k.$$

Thus

$$P^{-1}\hat{x} = \sum_{k=1}^{K} P_k^{-1}\hat{x}_k.$$

The proof for the expression for the global error covariance is similar. □

This result is useful since it allows the complexity of calculation at the fusion center to go down considerably[1]. Of course it assumes that the sensors can do some computation, but that is reasonable. The form of the global estimator shows that what we really want is a weighted mean of the local estimates. Each estimate is weighted by the inverse of the error covariance matrix. Thus more confidence we have in a particular sensor, more trust do we place in it.

2.1.3 Sequential Measurements from One Sensor

The same algorithm can be extended to the case when there are multiple measurements from one sensor. Furthermore, the processing can be done in a sequential manner. Consider a random variable evolving in time as

$$X_{n+1} = AX_n + w_n,$$

where w_n is white zero mean Gaussian noise with covariance matrix Q. The sensor generates a measurement at every time step according to the equation

$$Y_n = CX_n + v_n,$$

where v_n is again white zero mean Gaussian noise with covariance matrix R. We wish to obtain an estimate of X_n given all the measurements $\{Y_0, Y_1, \ldots, Y_n\}$. Suppose we divide the measurements into two sets:

1. The measurement Y_n.
2. The set $\mathcal{Y}$ of the remaining measurements Y_0 through Y_{n-1}.

Now note that the two sets of measurements are related linearly to X_n and further *the measurement noises are independent.* Thus we can combine the local estimates to obtain a global estimate. First we calculate the estimate of X_n based on Y_n. It is given by

$$M^{-1}\hat{X} = C^T R^{-1} Y_n,$$

where M is the error covariance given by

$$M^{-1} = R_{X_n}^{-1} + C^T R^{-1} C.$$

Let $\hat{X}_{n-1|n-1}$ be the estimate of X_{n-1} based on $\mathcal{Y}$ and $P_{n-1|n-1}$ be the corresponding error covariance. Then the estimate of X_n given $\mathcal{Y}$ is given by

$$\hat{X}_{n|n-1} = A\hat{X}_{n-1|n-1},$$

with the error covariance

$$P_{n|n-1} = AP_{n|n-1}A^T + Q.$$

[1] As an exercise, compare the number of elementary operations (multiplications and additions) for the two algorithms.

Thus the estimate of X_n given all the measurements is given by the combination of local estimates and can be seen to be

$$\begin{aligned} P_{n|n}^{-1}\hat{X}_{n|n} =& P_{n|n-1}^{-1}\hat{X}_{n|n-1} + M^{-1}\hat{X} \\ =& P_{n|n-1}^{-1}\hat{X}_{n|n-1} + C^T R^{-1} Y_n. \end{aligned}$$

The corresponding error covariance is

$$P_{n|n}^{-1} = P_{n|n-1}^{-1} + M^{-1} - R_{X_n}^{-1} = P_{n|n-1}^{-1} + C^T R^{-1} C.$$

These equations form the time and measurement update steps of the Kalman filter. Thus the Kalman filter can be seen to be a combination of estimators. This also forms an alternative proof of the optimality of the Kalman filter in the minimum mean squared sense under the stated assumptions. We will give more detail on Kalman filtering, and in particular on distributed Kalman filtering below.

2.2 Dynamic Sensor Fusion

Suppose there are multiple sensors present that generate measurements about a random variable that is evolving in time. We can again ask the question about how to fuse data from all the sensors for an estimate of the state X_n at every time step n. This is the question of dynamic sensor fusion. We will begin by seeing why this question is difficult.

To begin with, the problem can be solved if all the sensors transmit their measurements at every time step. The central node in that case implements a Kalman filter (which we will refer to from now as the *centralized* Kalman filter). However, there are two reasons why this may not be the preferred implementation.

1. The central node needs to handle matrix operations that increase in size as the number of sensors increases. We may want the sensors to shoulder some of the computational burden.

2. The sensors may not be able to transmit at every time step. Hence we may want to transmit after some local processing, rather than transmit raw measurements.

We will initially assume that the sensors can transmit at every time step and concentrate on reducing the computational burden at the central node.

2.2.1 Transmitting Local Estimates

Our first guess would be to generate a local estimate at each sensor that extracts all the relevant information out of the local measurements and then to combine the estimates using methods outlined above. However, in general, it is not possible to use above method. Consider K sensors being present with the k-th sensor generating a measurement of the form

$$y_{n,k} = C_k x_n + v_{n,k}.$$

Suppose we denote by Y_k the set of all the measurements from the sensor k that can be used to estimate the state x_n, i.e., the set $\{y_{0,k}, y_{1,k}, \ldots, y_{n,k}\}$. We wish to see if the local estimates formed by the sets Y_k's can be combined to yield the optimal global estimate of x_n. We can think of two ways of doing this:

1. We see that the set Y_i is linearly related to $x(k)$ through an equation of the form
$$\begin{bmatrix} y_{n,k} \\ y_{n-1,k} \\ \vdots \\ y_{0,k} \end{bmatrix} = \begin{bmatrix} C_k \\ C_k A^{-1} \\ \vdots \end{bmatrix} x_n + \begin{bmatrix} v_{n,k} \\ v_{n-1,k} - CA^{-1} w_{n-1} \\ \vdots \end{bmatrix}.$$
However we notice that the process noise w appears in the noise vector. Thus even though the measurement noises $v_{n,k}$'s may be independent, the noise entering the sets Y_k become correlated and hence the estimates cannot be directly combined. Of course, if the process noise is absent, the estimates can be combined in this fashion (see, e.g, [1] where the optimality in this special case was established. For a general discussion about the effects introduced by the process noise see, e.g. [2, 3, 4, 5, 6]).

2. We see that x_n can be estimated once the variables x_0, w_0, …, w_{n-1} are estimated. Now Y_k is linearly related to these variables through
$$\begin{bmatrix} y_{n,k} \\ y_{n-1,k} \\ \vdots \\ y_{0,k} \end{bmatrix} = \begin{bmatrix} C_i A^k & C_i A^{k-1} & \cdots & C \\ C_i A^{k-1} & \cdots & C & 0 \\ \vdots & & & \end{bmatrix} \begin{bmatrix} w_{n-1} \\ w_{n-2} \\ \vdots \\ x_0 \end{bmatrix} + \begin{bmatrix} v_{n,k} \\ v_{n-1,k} \\ \vdots \\ v_{0,k} \end{bmatrix}.$$
Now the measurement noises for different sensors are uncorrelated and the estimates can be combined. However, the vector being transmitted from either of the sensors is increasing in dimension as the time step n increases. Moreover the computation required is increasing since a matrix of size growing with time needs to be inverted at every time step. Hence this is not a practical solution.

Thus we see that it is not straight-forward to combine local estimates to obtain the global estimate. We can ask the question if it is possible at all to obtain the global estimate from the local estimates. Thus imagine that the local estimates $\hat{x}_{n,k}$ were being combined in the optimal fashion. Is it possible to generate the global estimate $\hat{x}_n$? As noted above, for the special case when there is no process noise, this is indeed true. However, in general, it is not possible.

Proposition 2.3. *(From [7]) Suppose two sets of measurements Y_1 and Y_2 are used to obtain local estimates $\hat{x}_1$ and $\hat{x}_2$. Let*
$$\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = L \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} \triangleq LY.$$
Then the global estimate $\hat{x}$ can be obtained from the local estimates $\hat{x}_1$ and $\hat{x}_2$ if and only if
$$R_{YY} L^T \left(L R_{YY} L^T \right)^{-1} L R_{YX} = R_{YX}.$$

Proof. The global estimate generated from the measurements is given by

$$\hat{x} = R_{XY} R_{YY}^{-1} Y.$$

If it is generated from the local estimates, it is given by

$$\hat{x} = R_{XY} L^T \left(L R_{YY} L^T \right)^{-1} LY.$$

The result is thus obvious. □

If L is invertible, the condition is satisfied and hence the global estimate can be generated from the local estimates. In general, however, L would be a fat matrix and hence the condition will not be satisfied. We thus have two options:

1. Find the best possible global estimator from the space spanned by the local estimates. This is left as an exercise.

2. Find the extra data that should be transmitted that will lead to the calculation of the global estimate. We will now describe some such schemes. For these and more such strategies see, e.g., [1],[6]–[19].

2.2.2 Distributed Kalman Filtering

For this section we will assume that the sensors are able to transmit information to the central node at every time step. We will use the following *information form* of the Kalman filter update equations.

Proposition 2.4. *Consider a random variable evolving in time as*

$$x_{n+1} = Ax_n + w_n.$$

Suppose it is observed through measurements of the form

$$y_n = Cx_n + v_n.$$

Then the measurement updates of the Kalman filter can be given by this alternate information form.

$$\begin{aligned} P_{n|n}^{-1} \hat{x}_{n|n} &= P_{n|n-1}^{-1} \hat{x}_{n|n-1} + C^T R^{-1} y_n \\ P_{n|n}^{-1} &= P_{n|n-1}^{-1} + C^T R^{-1} C. \end{aligned}$$

Proof. The equations were derived in section 2.1.3. □

The basic result about the requirements from the individual sensors can be derived using the above result.

Proposition 2.5. *The global error covariance matrix and the estimate are given in terms of the local covariances and estimates by*

$$\begin{aligned} P_{n|n}^{-1} &= P_{n|n-1}^{-1} + \sum_{k=1}^{K} \left(P_{n,k|n-1}^{-1} - P_{n,k|n}^{-1} \right) \\ P_{n|n}^{-1} \hat{x}_{n|n} &= P_{n|n-1}^{-1} \hat{x}_{n|n-1} + \sum_{k=1}^{K} \left(P_{n,k|n}^{-1} \hat{x}_{n,k|n} - P_{n,k|n-1}^{-1} \hat{x}_{n,k|n-1} \right). \end{aligned}$$

Proof. Proof follows by noting that the global estimate is given by

$$\begin{aligned} P_{n|n}^{-1}\hat{x}_{n|n} &= P_{n|n-1}^{-1}\hat{x}_{n|n-1} + C^T R^{-1} y_n \\ P_{n|n}^{-1} &= P_{n|n-1}^{-1} + C^T R^{-1} C. \end{aligned}$$

Since R is block diagonal, the terms $C^T R^{-1} y_n$ and $C^T R^{-1} C$ are decomposed into the sums

$$\begin{aligned} C^T R^{-1} y_n &= \sum_{k=1}^{K} C_k^T R_k^{-1} y_{n,k} \\ C^T R^{-1} C &= \sum_{k=1}^{K} C_k^T R_k^{-1} C_k. \end{aligned}$$

Noting the for the k-th sensor, the estimate and the error covariance are given by

$$\begin{aligned} P_{n,k|n}^{-1}\hat{x}_{n,k|n} &= P_{n,k|n-1}^{-1}\hat{x}_{n,k|n-1} + C_k^T R_k^{-1} y_{n,k} \\ P_{n|n}^{-1} &= P_{n|n-1}^{-1} + C_k^T R_k^{-1} C_k, \end{aligned}$$

the result follows immediately. □

Based on this result we now give two architectures for dynamic sensor fusion.

1. In the first, rather obvious, architecture, the individual sensors transmit the local estimates $\hat{x}_{n,k|n}$. The global fusion center combines the estimates using the theorem given above. Note that the terms $\hat{x}_{n|n-1}$ and $\hat{x}_{n,k|n-1}$ can be calculated by the fusion node by using the time update equation

$$\hat{x}_{n|n-1} = A\hat{x}_{n-1|n-1}.$$

   Similarly all the covariances can also be calculated without any data from the sensor nodes. This method is simple, especially at the sensor level. However, the fusion node has to do a lot of computation.

2. This method makes the computation at the fusion node simple at the expense of more data transmitted from the sensor node. The essential point is the observation as developed, e.g., in [20, 21] that the term $P_{n|n-1}^{-1}\hat{x}_{n|n-1}$ can be written in terms of contributions from individual sensors, i.e.,

$$P_{n|n-1}^{-1}\hat{x}_{n|n-1} = \sum_{k=1}^{K} z_{n,k}. \tag{1}$$

   This can be proved using straight-forward algebraic manipulation as follows.

$$\begin{aligned} P_{n|n-1}^{-1}\hat{x}_{n|n-1} =& P_{n|n-1}^{-1} A\hat{x}_{n-1|n-1} \\ =& P_{n|n-1}^{-1} A P_{n-1|n-1} P_{n-1|n-1}^{-1}\hat{x}_{n-1|n-1} \\ =& P_{n|n-1}^{-1} A P_{n-1|n-1} \Big(P_{n-1|n-2}^{-1}\hat{x}_{n-1|n-2} \\ &+ \sum_{k=1}^{K} \left(P_{n-1,k|n-1}^{-1}\hat{x}_{n-1,k|n-1} - P_{n-1,k|n-2}^{-1}\hat{x}_{n-1,k|n-2} \right) \Big) . \end{aligned}$$

Thus $z_i(k)$ evolves according to the relation

$$\begin{aligned} z_{n,k} = {} & P^{-1}_{n|n-1} A P_{n-1|n-1} z_{n,k|n-1} \\ & + \left(P^{-1}_{n-1,k|n-1} \hat{x}_{n-1,k|n-1} - P^{-1}_{n-1,k|n-2} \hat{x}_{n-1,k|n-2} \right), \end{aligned} \quad (2)$$

which depends only on the k-th sensor's data. The covariances do not depend on the data and can be calculated anywhere. Hence each sensor transmits the quantity

$$\gamma_{n,k} = \left(P^{-1}_{n,k|n} \hat{x}_{n,k|n} - P^{-1}_{n,k|n-1} \hat{x}_{n,k|n-1} \right) + z_{n,k} \quad (3)$$

and the fusion node just calculates the sum of these quantities. Thus at expense of more data transmitted from the sensor nodes, we have made the central node very simple.

3 Non-Ideal Networks with Star Topology

In this section, we will give some strategies or algorithms for sensors to perform distributed estimation if the communication network suffers from limited bandwidth, transmit range, and message loss. We consider various cases in the sequel.

3.1 Sensor Fusion in Presence of Message Loss

This research direction considers the following problem. Consider multiple sensors as above with a central fusion center. The sensors transmit data to the fusion center across an analog erasure link that drops messages stochastically. More formally, an analog erasure link accepts as input a real vector $i(n) \in \mathbf{R}^t$ for a bounded dimension t. At every time n, the output $o(n)$ is given by

$$o(n) = \begin{cases} i(n) & \text{with probability } 1-p \\ \emptyset & \text{otherwise.} \end{cases}$$

- The case when $o(n) = \emptyset$ is referred to as an erasure event. It implies that the channel drops the messages and the receiver does not receive any information apart from that an erasure event has occurred.
- This model assumes that the erasure events occur according to a Bernoulli process with erasure probability $1-p$. Other models, in which such events occur according to a Markov chain or other more general processes, can be considered.
- If the transmitter also knows that an erasure event has occurred, then we say that the receiver transmits an acknowledgement to the transmitter. Such an acknowledgement may always be available, may itself be transmitted across an erasure channel so that it is stochastically available, or may not be available at all.

The basic effect of the sensors transmitting across such channels is that information from the sensors is not available at the fusion center at every time step. This fact also requires some care in how the performance of the estimator is defined. Consider a realization of the erasure process such that at time n, the last transmission from sensor k was received at the fusion center at time n_k. Obviously, there is no algorithm that can provide a better estimate than the MMSE estimate of $x(n)$ given measurements $\{y_{0,1}, \cdots, y_{n_1,1}\}$, $\{y_{0,2}, \cdots, y_{n_2,2}\}$, $\cdots$, $\{y_{0,K}, \cdots, y_{n_K,K}\}$ (where we assume K sensors are present). Denote the error covariance of this estimator by P_n^{opt}. Due to the stochastic erasure process, it may be more convenient to consider the expected value of this covariance $E[P_n^{\text{opt}}]$ where the expectation is taken with respect to the erasure processes. Several questions arise:

1. What information should the sensors transmit to enable the fusion center to achieve the covariance P_n^{opt} at every time step?

2. If this covariance is not achievable, what is the best covariance that any algorithm can achieve?

3. Clearly, the error covariance at the fusion center degrades as the erasure probabilities increase. What are the conditions on the erasure probabilities so that any algorithm can achieve stability of the estimate error covariance, i.e., ensure that the expected error covariance remains bounded as $n \to \infty$?

We discuss below some recent work on these questions, although a complete solution is unavailable at this time.

It should be clear that an algorithm may lead to stability of the error covariance without being optimal in the sense of achieving the covariance P_n^{opt}. In other words, the requirement in the third question posed above is less strenuous than the requirement in the first question. The third question was answered in [21] which presented conditions on erasure probabilities and the process matrices for stability. We present the result below for the case when two sensors transmit data to the fusion center across individual analog erasure links with Bernoulli erasures with erasure probabilities $1 - p_k$, $k = 1, 2$. Various generalizations are available in the cited reference.

Theorem 3.1 (From [21]). *Consider a process evolving as*

$$x_{n+1} = Ax_n + w_n$$

being observed using two sensors that generate measurements of the form

$$y_{n,k} = C_k x_n + v_{n,k}, \qquad i = 1, 2$$

where w_n and $v_{n,k}$ are white zero mean independent noises. Let the sensors transmit information through a real vector with bounded dimension to a fusion center across analog erasure channels with Bernoulli erasures with erasure probabilities p_1 and p_2 respectively. Denote by $\rho(A_k)$ the spectral radius of the unobservable part of matrix A when the pair (A, C_k) is written in the observer canonical form and by $\rho(A)$ the spectral radius of matrix A. Assume that the pair $(A, [C_1^T, C_2^T]^T)$ is observable.

1. *Irrespective of the information transmitted by the sensors, and the algorithm used by the fusion center, the quantity $E[P_n^{\text{opt}}]$ is not bounded as $n \to \infty$ if at least one of the following inequalities is not satisfied:*

$$p_1\rho(A_2)^2 \leq 1 \tag{4}$$

$$p_2\rho(A_1)^2 \leq 1 \tag{5}$$

$$p_1p_2\rho(A)^2 \leq 1. \tag{6}$$

2. *Conversely, if the inequalities* (4)–(6) *are satisfied, then there is an algorithm such that the corresponding expected error covariance at the fusion center is bounded as time increases.*

Thus, this result solves the third problem posed above. It is important to note that the necessity of the inequalities (4)–(6) holds irrespective of the availability of acknowledgements at the sensors. The necessity part of the result follows from system theoretic considerations. The sufficiency part of the result is proved by constructing an algorithm that guarantees stability of the estimator error covariance, even though the error covariance is not P_n^{opt} (i.e., the algorithm is not optimal in the sense of achieving the minimal error covariance at every step). Perhaps somewhat surprisingly, the algorithm is based on the sensors transmitting local estimates of the process state based on their own measurements. Specifically, sensor 1 transmits the estimate $\hat{x}_{n,1}^1$ of the modes of the process observable only from sensor 1, and $\hat{x}_{n,1}^2$ of the modes observable from both sensors. Similarly sensor 2 transmits the estimate $\hat{x}_{n,2}^1$ of the modes of the process observable only from sensor 2, and $\hat{x}_{n,2}^2$ of the modes observable from both sensors. The fusion center maintains an estimate $\hat{x}_n^1$ of the modes observable only from sensor 1, $\hat{x}_n^2$ of the modes observable only from sensor 2, and $\hat{x}_n^3$ of the modes observable from both sensors. At any time step, it updates the modes as follows:

$$\hat{x}_n^1 = \begin{cases} \hat{x}_{n,1}^1 & \text{transmission successful from sensor 1} \\ A\hat{x}_{n-1}^1 & \text{otherwise} \end{cases}$$

$$\hat{x}_n^2 = \begin{cases} \hat{x}_{n,2}^1 & \text{transmission successful from sensor 2} \\ A\hat{x}_{n-1}^2 & \text{otherwise} \end{cases}$$

$$\hat{x}_n^2 = \begin{cases} \hat{x}_{n,1}^2 & \text{transmission successful from sensor 1} \\ \hat{x}_{n,2}^2 & \text{transmission successful from sensor 2 but not from sensor 1} \\ A\hat{x}_{n-1}^1 & \text{otherwise.} \end{cases}$$

The estimate of the state x_n can be formed from the three components $\hat{x}_n^k$. Given this algorithm, the sufficiency of the inequalities (4)–(6) for stability of the expected error covariance can then be proved.

Given Proposition 2.3, it is not surprising that this algorithm cannot lead to the calculation of the optimal global estimate at the fusion center. In fact, the optimal information processing algorithm at the sensors remains unknown in most cases. Most recent advances (e.g. [20]–[23] build from the basic algorithm identified in equations (1)–(3). Thus the sensors transmit the quantity $\gamma_{n,k}$ at every time step and the fusion center sums these quantities to generate the estimate $\hat{x}_n$. If there are no erasures, this estimate is indeed the global estimate with the optimal error covariance P_n^{opt}. However, if there are erasures, then the calculation of $\gamma_{n,k}$ requires some global knowledge. In particular, the quantity $P_{n-1|n-1}$ in (2) at each sensor requires the knowledge of the last time step at which the transmission from every sensor to the fusion center was successful. Notice that the data transmitted by other sensors is not required, merely the confirmation of successful transmission is enough.

One mechanism for such global knowledge can be acknowledgements transmitted from the fusion center. If such acknowledgements are available, then it was shown in [21] that minor modifications of the algorithm outlined in equations (1)–(3) will generate the optimal global estimate at the fusion center. Depending on the problem scenario, such an assumption may or may not be realistic. If acknowledgements are also transmitted across an analog erasure link, [23] presented some further modifications to the algorithm that guaranteed that the estimation error covariance degraded continuously as a function of the probability of loss of acknowledgement. However, the optimal algorithm when acknowledgements are not available, or only available intermittently, is not known.

Other special cases where such global knowledge is available can be if only one of the sensors transmits across an analog erasure link [20] or if only one sensor transmits at any time [22]. Once again, in these cases, the optimal global estimate can be calculated. However, it remains unknown if the optimal global estimate can be calculated outside of these cases, or if it cannot be calculated, then what is the best performance that is achievable.

3.2 Sensor Fusion with Limited Bandwidth

3.2.1 Static Sensor Fusion

Consider a limited bandwidth communication network, in which K sensors measure an unknown parameter $\theta \in [-U, U]$. The measurement x_k, from k-th sensor, is corrupted by noise n_k, which is assumed independent, zero mean, and with a pdf $p(u)$, namely $\Pr(n_k = u) = p(u)$.

$$x_k = \theta + n_k \qquad \text{for } k = 1, 2, \ldots, K\,. \tag{7}$$

Depending on the distribution of the noise, and on the amount of information that it is transmitted, there can be the cases studied in the following subsections:

An ϵ-estimator with known noise pdf Here an ϵ-estimator is defined as an estimator providing estimates with MSE lower than ϵ^2. Assume the limited bandwidth forces each sensor to send just one bit messages $m_k(x_k)$ to the fusion center. The message is defined as

$$m_k(x_k) = \begin{cases} 1, & \text{if } x_k \in S_k \\ 0, & \text{if } x_k \notin S_k \end{cases}, \tag{8}$$

where S_k is a subset of $\mathbb{R}$ and is independent of the noise pdf. Let $\mathbb{R}_+$ denote the subset of $\mathbb{R}$ for all positive real number.

Example 3.2. *(From [24]) Suppose that the noise is uniformly distributed over the interval $[-U, U]$. Let $S_k = \mathbb{R}_+$ for all k. Suppose a linear fusion function that gives the estimator $\hat{\theta}$ as*

$$\hat{\theta} := f(m_1, \ldots, m_k) = -U + \frac{2U}{K} \sum_{k=1}^{K} m_k \,.$$

Then, the estimator is unbiased:

$$\begin{aligned}\mathbb{E}\,[\hat{\theta}] &= -U + \frac{2U}{K} \sum_{k=1}^{K} \mathbb{E}\,[m_k] \\ &= -U + \frac{2U}{K} K \frac{U+\theta}{2U} = \theta \,.\end{aligned}$$

Furthermore, since m_k's are independent,

$$\begin{aligned}\mathbb{E}\,[\hat{\theta} - \theta]^2 &= \frac{4U^2}{K^2}\,\mathbb{E}\left[\sum_{k=1}^{K}\left(m_k - \frac{U+\theta}{2U}\right)\right]^2 \\ &= \frac{4U^2}{K^2} \sum_{k=1}^{K} \mathbb{E}\,\left[m_k - \mathbb{E}\,[m_k]\right]^2 \leq \frac{U^2}{K} \,,\end{aligned}$$

where we used that the variance of a binary random variable is bounded above by 1/4. It indicates that, even with the binary message constraint, a total number of $K = U^2/\epsilon^2$ sensors are still sufficient to perform an ϵ-estimator for θ.

Generally, if the $p(u)$ is given, we can still choose the message function as Eq.(8) with $S_k = \mathbb{R}_+$ for all k. Then

$$\begin{aligned}\Pr(m_k = 1) = \Pr(n_k > -\theta) = \textstyle\int_{-\theta}^{\infty} p(u)du \,, \\ \Pr(m_k = 0) = \Pr(n_k \leq -\theta) = \textstyle\int_{-\infty}^{-\theta} p(u)du \,.\end{aligned}$$

Then the expectation value for $\mathbb{E}\,[m_k]$ is obtained by

$$\mathbb{E}\,[m_k] = \int_{-\theta}^{\infty} p(u)du = 1 - F(-\theta), \quad k = 1, 2, \ldots, K \,,$$

where $F(\cdot)$ is the cumulative distribution function (cdf) of the noise. If one chooses the final fusion function for $\hat{\theta}$ as introduced in [24], then

$$\hat{\theta} := f(m_1, \ldots, m_k) = -F^{-1}\left(1 - \frac{1}{K}\sum_{k=1}^{K} m_k\right) \,, \tag{9}$$

where F^{-1} is the inverse of F. By the strong law of large numbers, it follows

$$\begin{aligned}\lim_{K\to\infty} \hat{\theta} &= -F^{-1}\left(1 - \lim_{K\to\infty} \frac{1}{K}\sum_{k=1}^{K} m_k\right) \\ &= -F^{-1}(1 - \mathbb{E}\,[m_k]) = -F^{-1}(F(-\theta)) = \theta \,.\end{aligned}$$

Suppose the noise pdf $p(u)$ is known and bounded over $[-U, U]$, then $\hat{\theta}$ obtained by Eq.(9) is an ϵ-estimate of θ implying a total number of $\mathcal{O}(1/\epsilon^2)$ sensors, by the following theorem:

Theorem 3.3. *(From [24]) Suppose the noise pdf $p(u)$ is known and bounded from below by $\mu > 0$ over $[-U, U]$. Let $K \geq 1/(4\mu^2\epsilon^2)$. Then the decentralized estimation scheme* (8) *and* (9) *produces an ϵ-estimator of θ.*

Proof. Notice that

$$\begin{aligned} |F(-\theta) - F(-\theta')| &= |1 - F(-\theta) - (1 - F(-\theta'))| \\ &= \left| \int_{-\theta}^{-\theta'} p(u)du \right| \geq \mu|\theta - \theta'| \qquad \forall \theta, \theta' \in [-U, U] \\ \Rightarrow |F^{-1}(v) - F^{-1}(v')| &\leq \frac{1}{\mu}|v - v'| \qquad \forall v, v' \in [0, 1] \,, \end{aligned}$$

Then,

$$\begin{aligned} |\hat{\theta} - \theta| &= \left| -F^{-1}\left(1 - \frac{1}{K}\sum_{k=1}^{K} m_k\right) + F^{-1}(1 - \mathbb{E}\,(m_k)) \right| \\ &\leq \frac{1}{\mu}\left| \frac{1}{K}\sum_{k=1}^{K} m_k - \mathbb{E}\,(m_k) \right| \\ \Rightarrow \mathbb{E}\,[\hat{\theta} - \theta]^2 &\leq \frac{1}{\mu^2} E\left[\frac{1}{K}\sum_{k=1}^{K} m_k - \mathbb{E}\,(m_k)\right]^2 \leq \frac{1}{4\mu^2 K} \,. \end{aligned}$$

Thus, the variance of the estimator given by Eq.(9) is lower than ϵ^2 as long as

$$K \geq \frac{1}{4\mu^2\epsilon^2} \,.$$

which concludes the proof. □

A universal ϵ-estimator for unknown noise pdf The ϵ-estimator introduced in Section 3.2.1 needs the explicit pdf $p(u)$ for the noise. However, sometimes for a large number of sensors, to characterize the measurement noise distribution would cost too much, or could be even impossible in a dynamic environment. To cope with these situations, a distributed estimator providing accurate estimates regardless the noise pdf under the bandwidth constraint is required. In this subsection, we summarize a universal distributed estimator for unknown noise pdf.

The idea, proposed in [24], is to represent the estimates in binary form by quantizing the sensor measurements into the corresponding bit positions. Specifically, it tries to quantize 2^{-i} of the sensors' measurements into the i-th most significant bit (MSB), e.g. 1/4 of the sensors quantize their measurement to the second MSB. Then it can be shown that the statistics average of these message functions $(m_1 + m_2 + \cdots + m_K)/K$ is a unbiased estimator for θ, while its MSE is upper bounded by $4U^2/K$.

The procedure of this distributed estimation scheme is described as follows [25, 26]:

1. Each measurement, x_k, in node k is quantized into the i-th MSB with probability 2^{-i}, being converged to a binary message. Then this message

is sent to the fusion center. This step can be described as following

$$\Pr(a = i) = \begin{cases} 2^{-i} & i = 1, 2, 3, \ldots \\ 0 & \text{otherwise} \end{cases} \tag{10a}$$

$$m_k(x, a) = [b(2U + x; a); a] \,, \tag{10b}$$

where the value of the random variable a indicates the position for MSB, and the notation $b(z; a)$ denotes the i-th MSB of a real number z.

2. The fusion center recursively computes the average of all received binary messages that are distinct (determined by, say, the sender's ID), and uses it as estimator of θ.

   Suppose the fusion center, which also has measurement capability, has received a total of j independent messages. Based on these messages, it can first form the sets

   $$\mathcal{N}_i = \{k | a_k = i, 1 \le k \le j\}, \quad i = 1, 2, 3, \ldots \,. \tag{11}$$

   Then, based on the received messages and its own observation x, the center can be proceed to form

   $$y_i = b(2U + x; i) + \sum_{k \in \mathcal{N}_i} b(2U + x_k; a_k), \quad i = 1, 2, 3, \ldots \,, \tag{12}$$

   and perform the estimate of θ

   $$\hat{\theta}_j = f_j(x, m(x_1, a_1), \ldots, m(x_j, a_j)) = -2U + 4U \sum_{i=1}^{\infty} \frac{2^{-i}}{|\mathcal{N}_i| + 1} y_i \tag{13}$$

Theorems 3.4 and 3.5 show that this distributed estimator is unbiased and has an expected MSE of $4U^2/K$, where K is the number of sensors in the network:

Theorem 3.4. *Let $f_j(x, m(x_1, a_1), \ldots, m(x_j, a_j))$ be defined by Eq.(13). Then for all $0 \le j \le K - 1$*

$$\mathbb{E}\left[f_j(x, m(x_1, a_1), \ldots, m(x_j, a_j))\right] = \theta, \quad \forall \theta \in [-U, U], \quad \forall p \in \mathcal{M}_U \,, \tag{14}$$

where the expectation is taken with respects to the distribution of a and unknown noise, and where

$$\mathcal{M}_U = \left\{ p(u) : \int_{-U}^{U} p(u) du = 1, \int_{-U}^{U} u p(u) du = 0, p(u) \ge 0, Supp(p) \subseteq [-U, U] \right\} .$$

Proof. From Eq.(12) and (13), using that x_k is i.i.d to each others, we obtain

$$\begin{aligned}
&\mathbb{E}\left[f_j(x, m(x_1, a_1), \ldots, m(x_j, a_j))\right] \\
&\quad = -2U + 4U \sum_{i=1}^{\infty} \mathbb{E}\left[\frac{2^{-i}}{|\mathcal{N}_i| + 1} \left(b(2U + x; i) + \sum_{k \in \mathcal{N}_i} b(2U + x_k; a_k) \right) \right] \\
&\quad = -2U + 4U \sum_{i=1}^{\infty} 2^{-i}\, \mathbb{E}\left[b(\theta + 2U + n; i)\right] \\
&\quad = -2U + \mathbb{E}\left[\theta + 2U + n\right] = \theta \,,
\end{aligned}$$

where note that every number u in $[0, 4U]$ can be represented in binary as

$$u = 4U \sum_{i=1}^{\infty} 2^{-i} b(u; i) \,,$$

which concludes the proof. □

Theorem 3.5. *Let $\hat{\theta}$ be the distributed estimator of Eq.*(13)*. Then*

$$\mathbb{E}\,[\hat{\theta}_j - \theta]^2 \leq \frac{4U^2}{j+1}$$

Proof. Similarly, from Eq.(12) and (13), using that x_k is i.i.d to each others,

$$\begin{aligned}
&\mathbb{E}\,[(\hat{\theta}_j - \theta)^2 | a_1, \ldots, a_i] \\
&\quad = 16U^2 \sum_{i=1}^{\infty} \mathrm{Var}\left[\frac{2^{-i}}{|\mathcal{N}_i| + 1} \left(b(2U + x; i) + \sum_{k \in \mathcal{N}_i} b(2U + x_k; a_k) \right) \right] \\
&\quad = 16U^2 \sum_{i=1}^{\infty} 2^{-2i} \frac{\mathrm{Var}[b(2U + x; i)]}{|\mathcal{N}_i| + 1} \\
&\quad \leq 4U^2 \sum_{i=1}^{\infty} 2^{-2i} \frac{1}{|\mathcal{N}_i| + 1} \,,
\end{aligned}$$

where in the last step follows from that the upper bound of $\mathrm{var}(b(2U + x; a))$ is $1/4$. Furthermore, notice that

$$\Pr(\mathcal{N}_i = r) = \left(\begin{array}{c} i \\ r \end{array} \right) 2^{-ir} (1 - 2^{-i})^{(j-r)}, \quad 0 \leq r \leq j \,,$$

and

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{|\mathcal{N}_i + 1|} \right] &= \sum_{r=0}^{j} \frac{1}{r+1} \left(\begin{array}{c} i \\ r \end{array} \right) 2^{-ir} (1 - 2^{-i})^{(j-r)} \\
&= \frac{1}{i+1} \frac{1 - (1 - 2^{-i})^{j+1}}{2^{-i}}
\end{aligned}$$

Therefore, the MSE is

$$\begin{aligned}
\mathbb{E}\,[\hat{\theta}_j - \theta]^2 &= \mathbb{E}\,[\,\mathbb{E}\,[(\hat{\theta}_j - \theta)^2 | a_1, \ldots, a_i]] \\
&\leq 4U^2 \sum_{i=1}^{\infty} 2^{-2i}\, \mathbb{E}\left[\frac{1}{|\mathcal{N}_i| + 1} \right] \\
&\leq \frac{4U^2}{j+1} \,,
\end{aligned}$$

which concludes the proof. □

Remark 3.6. *The average message length is $\sum_{i=1}^{\infty} 2^{-i}(1 + \lceil \log(a) \rceil)$ and is upper bounded by 2.5078 [26].*

3.2.2 Dynamic Sensor Fusion: Sign of innovations-KF

An alternative solution in the presence of limited bandwidth is based on the Kalman filter. First, recall that generally distributed Kalman filter includes a prediction step and a correction step. Consider the system

$$\begin{aligned} x_n &= A_n x_{n-1} + w_n \\ y_{n,k} &= C_{n,k}^T x_n + v_{n,k} \,, \end{aligned}$$

where the driving input w_n is normally distributed with zero mean and variance Q_n and the observation noise $v_{n,k}$ is zero mean AWGN and independent across sensors with noise R [25]. In this case, we have $R = \sigma_v \mathbf{I}$. Suppose that $\hat{x}_{n-1|n-1}$ and $P_{n-1|n-1}$ are available at time n, the predicted estimate $\hat{x}_{n|n-1}$ and its corresponding covariance matrix $P_{n|n-1}$ are given by

$$\hat{x}_{n|n-1} = A_n \hat{x}_{n-1|n-1} \tag{15a}$$

$$P_{n|n-1} = A_n P_{n-1|n-1} A_n^T + Q_n \,. \tag{15b}$$

The innovation sequence

$$\tilde{y}_n := y_n - C_n^T \hat{x}_{n|n-1} \,,$$

is chosen to obtain the corrected estimate $\hat{x}_{n|n}$. To deal with the limited bandwidth, the sign of the innovation (SOI) is used to ensure that the required exchange of information among sensors is possible under one bit message constraint.

$$m(n) := \mathrm{sign}[\mathrm{\tilde{y}_n}] = \mathrm{sign}[\mathrm{y_n} - \mathrm{\tilde{y}_{n|n-1}}] \,. \tag{16}$$

Due to the sign non-linearity, $p[x_n|m_{0:n-1}]$ is non-Gaussian and computation of the exact MMSE estimate requires numerical integrations and propagation of the posterior pdf. However, base on customary simplifications made in nonlinear filtering, we can approximate the MMSE with following correction recursions [27]:

$$\hat{x}_{n|n} = \hat{x}_{n|n-1} + m_n \frac{(\sqrt{2/\pi}) P_{n|n-1} C_n}{\sqrt{C_n^T P_{n|n-1} C_n + \sigma_v^2}} \tag{17a}$$

$$P_{n|n} = P_{n|n-1} - \frac{(2/\pi) P_{n|n-1} C_n^T P_{n|n-1}}{C_n^T P_{n|n-1} C_n + \sigma_v^2} \,. \tag{17b}$$

Even at a minimal communication cost, the SOI-KF is strikingly similar to the clairvoyant KF [25]. To prove it, let us rewrite the SOI-KF correction as

$$\hat{x}_{n|n} = \hat{x}_{n|n-1} + \frac{P_{n|n-1} C_n}{C_n^T P_{n|n-1} C_n + \sigma_v^2} \tilde{m}_{n|n-1} \,, \tag{18}$$

where

$$\tilde{m}_{n|n-1} := \sqrt{(2/\pi)\, \mathbb{E}\,[\tilde{y}_{n|n-1}^2]}\, m_n \,.$$

Notice that the units of $\tilde{m}_{n|n-1}$ and $\tilde{y}_{n|n-1}$ are the same, and

$$\begin{aligned} \mathbb{E}\,[\tilde{m}_{n|n-1}] &= \mathbb{E}\,[\tilde{y}_{n|n-1}] = 0 \\ \mathbb{E}\,[\tilde{m}_{n|n-1}]^2 &= \frac{2}{\pi}\, \mathbb{E}\,[\tilde{y}_{n|n-1}]^2 \,, \end{aligned}$$

which indicates that Eq.(18) is identical to the KF update if replacing $\tilde{m}_{n|n-1}$ with the innovation $\tilde{y}_n$. It is not difficult to show that the MSE increases when using the SOI-KF is as much as the KF would incur when applied to a model with $\pi/2$ higher observation noise variance [25, 27].

4 Network with Arbitrary Topology

The results above assumed the presence of a star topology in which one central node had access to local estimates from every other node. It was essentially a two step procedure: first all the nodes transmit local estimates or local measurements to the central node and then the central node calculates and transmits the weighted sum of the local estimates back. However, what is required is a weighted average. Thus, we can generalize the approach to an arbitrary graph at the expense of more time being employed. The generalization is along the lines of average consensus algorithms that have been recently considered by many people (see, e.g., [28]–[30]). An example of arbitrary topology networks is illustrated in Fig 2. For now, we will only cover the basics.

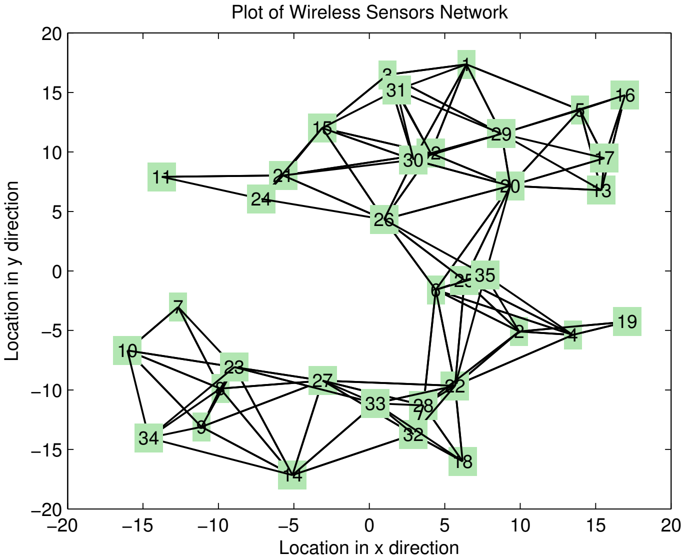

Figure 2: An example of arbitrary topology networks with nodes and links (solid lines indicating that there is message communication between nodes). In this network, there is no node acting as fusion center.

4.1 Static Sensor Fusion with Limited Communication Range

Due to the limited communication range, some of the sensors can not send message to the fusion center. In such cases, we can treat the networks as the static sensor fusion for arbitrary graphs without a fusion center.

Consider K nodes each with access to a scalar value being connected according to an arbitrary (but time-invariant) graph. Suppose we want each node to calculate the average of all the numbers. One way to do that is if each node implements the dynamical system

$$x_{n+1,k} = x_{n,k} + h \sum_{j \in \mathcal{N}_i} \left(x_{n,j} - x_{n,k} \right) \, ,$$

where $x_{n,k}$ is the value for state x_k at time n, and h is a small positive constant. On stacking the states of all the nodes, the entire system evolves as

$$X_{n+1} = (I - hL)X_n,$$

where $X_n = [x_{n,1}, \dots, x_{n,K}]^T$, and L is the *Graph Laplacian* matrix. If the underlying graph is connected, L has the following properties:

1. It is a symmetric positive-definite matrix. Thus the dynamics is stable (assuming h is small enough) and reaches a steady-state.
2. Each row sum is 0. Thus any vector with identical components is an equilibrium.
3. Each column sum is 0. Thus the sum of entries X_n is conserved at every time step.

Because of these three properties, it is easy to see that each entry must converge to the average of the sum of the initial conditions. This algorithm can then be readily extended for calculating *weighted* averages of *vectors* [31, 32]. If the initial values are given by the vectors $x_{0,k}$, each node calculates the following:

$$x_{n+1,k} = x_{n,k} + hW_k^{-1} \sum_{j \in \mathcal{N}_i} \left(x_{n,j} - x_{n,k} \right) \, ,$$

where $\mathcal{N}_i$ denotes the set of sensors connected to i-th sensor. In our case, we let $x_{0,k}$ to be the local estimate values and W_k to be inverse of the local estimation error covariance, and obtain the required weighted sum.

4.2 Dynamic Sensor Fusion

4.2.1 With Limited Communication Range

Consider a WSN with $K > 1$ sensor nodes placed at random and static positions in space. At every time instant, each sensor in the network takes a noisy measurement $y_k(n)$ of a scalar signal $x(n)$ for $n \in \mathbb{N}_0$ and for all $k = 1, \dots, K$ as described by Eq.(19):

$$x_n = ax_{n-1} + \delta_{n-1} \tag{19a}$$

$$y_{n,k} = c_k x_n + v_{n,k} \, , \tag{19b}$$

where $v_{n,k}$ for all k is normal distribution with zero means and variance $\sigma^2_{v_{n,k}}$ respectively and $\mathbb{E}\left(v_{n,i}v_{n,j}\right) = 0$ for all $n \in \mathbb{N}_0$, $i \neq j$. Moreover, δ_n models the disturbance for the scalar signal.

In this case, each sensor has limited communication range. To estimate the dynamic state x_n in the networked manner, we assume every sensor k computes an estimate $\hat{x}_{n,k}$ of x_n by taking a linear combination of its own and of its neighbours' estimates and measurements. Define $\hat{\mathbf{x}}_n = (\hat{x}_{n,1}, \dots, \hat{x}_{n,K})^T$ and similarly $\mathbf{y}_n = (y_{n,1}, \dots, y_{n,K})^T$, then each node computes

$$\hat{x}_{n,k} = a\boldsymbol{\kappa}_{n,k}^T \hat{\mathbf{x}}_{n-1} + \mathbf{h}_{n,k}^T \left(\mathbf{y}_n - a\mathbf{C}\hat{\mathbf{x}}_{n-1}\right) , \tag{20}$$

with $\hat{\mathbf{x}}_0 = \mathbf{y}_0$, $\mathbf{C}$ is a diagonal matrix $\mathrm{diag}(\{c_1, \dots, c_K\})$ with $\boldsymbol{\kappa}_{n,k}^T \in \mathbb{R}^{K\times 1}$, in which the j-th element is the weight coefficient used by node k for information coming from node j at time n, as seen from node k with respect to all nodes of the network.

Let denote $\mathbf{e}_n = (e_{n,1}, \dots, e_{n,K})^T$, with $e_{n,k} = x_n - \hat{x}_{n,k}$, $k = 1, \dots, K$, be the vector of the estimation errors. Assume that $\boldsymbol{\kappa}_{n,k}^T \mathbf{1} = 1$, then for each node k, the error dynamics can be obtained by

$$\begin{aligned} e_{n,k} =& a(\boldsymbol{\kappa}_{n,k} - \mathbf{C}\mathbf{h}_{n,k})^T \mathbf{e}_{n-1} + (\boldsymbol{\kappa}_{n,k} - \mathbf{C}\mathbf{h}_{n,k})^T \delta_{n-1} - \mathbf{h}_{n,k}^T \mathbf{v}_n \\ =& a\mathbf{g}_{n,k}^T \mathbf{e}_{n-1} + \mathbf{g}_{n,k}^T \delta_{n-1} - \mathbf{h}_{n,k}^T \mathbf{v}_n , \end{aligned}$$

with $\mathbf{g}_{n,k} = \boldsymbol{\kappa}_{n,k} - \mathbf{C}\mathbf{h}_{n,k}$ and where $\mathbf{v}_n = (v_{n,1}, \dots, v_{n,K})^T$.

Define $\mathbf{G}_n$ the matrix with k-th row given by the vector $\mathbf{g}_{n,k}$, for $k = 1, \dots, K$. The average estimation error of the estimator (20) is bounded throughout the network provided that a condition on the maximum singular value of the matrix $\mathbf{G}_n$, $\gamma_{\max}(\mathbf{G}_n)$, holds:

Proposition 4.1. *Assume that*

(i) $\gamma(\mathbf{G}_n) \le \gamma_{\max} < \min(1, 1/a)$ *for all* $n \in \mathbb{N}_0$, *where* $\gamma(\cdot)$ *is the singular value of matrix.*

(ii) $\delta_n = d_n + w_n$, *where* $|d_n| < \Delta$ *represents the disturbances and* $w_n \sim \mathcal{N}(0, \sigma_w^2)$ *is Gaussian noise for all* $n \in \mathbb{N}_0$.

Then the correlation function of the estimation error, computed with respect to the measurement noise and message losses, is

$$\lim_{n\to+\infty} \|\,\mathbb{E}_{\mathbf{w},\mathbf{v}} \mathbf{e}_n\| \le \frac{\Delta\sqrt{K}\gamma_{\max}}{1-\gamma_{\max}} . \tag{21}$$

Proof. The dynamics of $\mathbf{e}_n$ are given by a stochastic time-varying linear system. Consider the function $V_n = \|\,\mathbb{E}_{\mathbf{v},\mathbf{w}} \mathbf{e}_n\|$. Simple algebra gives that

$$\begin{aligned} V(t) &\le \|a\mathbf{G}_n\| V_n + \|\mathbf{G}_n\| \Delta\sqrt{K} \\ &\le (a\gamma_{\max})^n V_0 + \gamma_{\max} \frac{1-\gamma_{\max}^{n-1}}{1-\gamma_{\max}} \Delta\sqrt{K} , \end{aligned}$$

from where, taking the limit $n \to +\infty$, the proposition follows. □

The previous proposition is useful because gives us a constraint on the weights so that the estimation error is stable. Moreover, we compute the weights so that the estimation error variance is minimized under the stability constraint. Every node computes the weights by solving at each time step the following optimization problem

$$\min_{\mathbf{g}_{n,k}, \mathbf{h}_{n,k}, \psi_{n,k}} \quad \mathbf{g}_{n,k}^T \Gamma_{n-1,k} \mathbf{g}_{n,k} + \mathbf{h}_{n,k}^T \mathbf{Q}_{n-1} \mathbf{h}_{n,k} \tag{22a}$$

$$\text{s.t.} \quad \left(\mathbf{g}_{n,k} + \mathbf{C}\mathbf{h}_{n,k}\right)^T \mathbf{1} = 1 \tag{22b}$$

$$\|\mathbf{g}_{n,k}\|^2 \le \psi_{n,k} , \tag{22c}$$

where

$$\begin{aligned}\Gamma_{n,k} &= (a^2\mathbf{P}_{n,k} + \sigma_w^2\mathbf{I})\\ \mathbf{P}_{n,k} &= \mathbb{E}\,(\mathbf{e}_{n,k} - \mathbb{E}\,\mathbf{e}_{n,k})(\mathbf{e}_{n,k} - \mathbb{E}\,\mathbf{e}_{n,k})^T\,,\end{aligned}$$

while $\mathbf{Q}_n = \Sigma_n$ with Σ which is a diagonal matrix, $\mathrm{diag}(\{\sigma_{v_1}^2, \dots, \sigma_{v_K}^2\})$. If the σ_w^2 is unknown, let $\Gamma_{n,k} = a^2\mathbf{P}_{n,k}$.

In this optimization problem, the objective function is the average variance of the estimation error at node k. The first constraint is motivated by assumption $\boldsymbol{\kappa}_{n,k}^T\mathbf{1} = 1$, whereas the last constraint is a consequence of Proposition 4.1 and the Proposition III.1.in [33]. Specifically, the last constraint in problem (22) guarantees that $\gamma(\mathbf{G}_n) \le \gamma_{\max}$, provided that there exists some positive scalars $\psi_{n,k}$, $k = 1, \dots, K$, such that

$$S_k(\boldsymbol{\psi}_{n,k}) = \psi_{n,k} + \sqrt{\psi_{n,k}} \sum_{j\in\Theta_k} \sqrt{\psi_{n,j}} - \gamma_{\max} \le 0\,, \tag{23}$$

where $\Theta_k = \{j \ne k : \mathcal{N}_k \cap \mathcal{N}_j \ne \emptyset\} \cup \{\mathcal{N}_k\}$, which is the collection of communicating nodes located at two hops distance from node k plus communicating neighbours of k at time n.

The optimal solution to problem (22) is obtained in two steps: first, $\psi_{n,k}$ is assumed fixed and the problem is solved by applying Lagrange dual theory for the variables $\mathbf{g}_{n,k}$ and $\mathbf{h}_{n,k}$, thus achieving expressions of $\mathbf{g}_{n,k}$ and $\mathbf{h}_{n,k}$ as function of $\psi_{n,k}$. Finally, these expressions are used in the cost function, which is then minimized in the valuable $\psi_{n,k}$. Details follows in the sequel:

By the first step, given a covariance matrix $\mathbf{P}_{n-1,k}$, the weights that solve the optimization problem (22) are

$$\mathbf{g}_{n,k} = \frac{(\Gamma_{n-1,k} + \lambda_{n,k}\mathbf{I})^{-1}\,\mathbf{1}}{\mathbf{1}^T\left((\Gamma_{n-1,k} + \lambda_{n,k}\mathbf{I})^{-1} + \mathbf{C}\mathbf{Q}^{-1}\mathbf{C}\right)\mathbf{1}}\,, \tag{24}$$

$$\mathbf{h}_{n,k} = \frac{\mathbf{Q}^{-1}\mathbf{C}\mathbf{1}}{\mathbf{1}^T\left((\Gamma_{n-1,k} + \lambda_{n,k}\mathbf{I})^{-1} + \mathbf{C}\mathbf{Q}^{-1}\mathbf{C}\right)\mathbf{1}}\,, \tag{25}$$

$$\lambda_{n,k} = \begin{cases} 0 & \text{if } \left[\mathbf{g}_{n,k}^T\mathbf{g}_{n,k}\right]_{\lambda_{n,k}=0} \le \psi_{n,k} \\ \lambda_{n,k}^* & \text{otherwise} \end{cases}\,. \tag{26}$$

Here $\lambda_{n,k}^*$ is determined by equation

$$\left[\mathbf{g}_{n,k}^T\mathbf{g}_{n,k}\right]_{\lambda_{n,k}^*} = \psi_{n,k}\,. \tag{27}$$

The value of $\lambda_{n,k}^*$ is in the interval

$$\left[0, \max\left(0, \frac{1}{\mathbf{1}^T\mathbf{C}\mathbf{Q}^{-1}\mathbf{C}\mathbf{1}}\sqrt{\frac{1}{\psi_{n,k}}} - a^2\ell(\mathbf{P}_{n-1,k})\right)\right]\,,$$

where $\ell(\mathbf{P}_{n,k})$ is the minimum eigenvalue of matrix $P_{n,k}$. Then $\lambda_{n,k}^*$ can be computed by a simple bisection algorithm [33].

The weights $\mathbf{g}_{n,k}$ and $\mathbf{h}_{n,k}$, whose expression are given in (24) and (25), depend on the thresholds $\psi_{n,k}$, through the values $\lambda_{n,k}$, and on the error covariance matrix $\mathbf{P}_{n-1,k}$. In case of perfect communication, each node could estimate efficiently the error covariance matrix from data. In particular, let $\hat{\mathbf{P}}_{n-1,k}$ the estimation of the covariance matrix computed by node k, then

$$\hat{\mathbf{P}}_{n-1,k} = \frac{1}{n} \sum_{\tau=0}^{n-1} \left(\hat{\boldsymbol{\epsilon}}_{\tau,k} - \hat{\mathbf{m}}_{\tau,n}\right) \left(\hat{\boldsymbol{\epsilon}}_{\tau,k} - \hat{\mathbf{m}}_{\tau,k}\right)^T \ , \tag{28}$$

where

$$\hat{\mathbf{m}}_{n,k} = \frac{1}{n} \sum_{\tau=0}^{n} \hat{\boldsymbol{\epsilon}}_{\tau,k} \ ,$$

is the sample mean. The vector $\hat{\boldsymbol{\epsilon}}_{n,k}$ is the vector of the estimation errors of the neighboring nodes available at node k, which is obtained by a Tichonov regularization approach, as discussed in [33] with a different matrix $\mathbf{A}$ given by

$$\mathbf{A} = \begin{pmatrix} \mathbf{1} & \mathbf{I} \\ \mathbf{C1} & \mathbf{0} \end{pmatrix} .$$

Now let us compute the values of $\psi_{n,k}$ that solve the optimization problem (22). By substituting (24) and (25) in the cost function of (22), we see that the larger is $\psi_{n,k}$, the lower is the cost function. In other words, the larger is $\psi_{n,k}$, the lower is the estimation error variance at node k. Since $\psi_{n,k}$ must be maximized for $k = 1, \ldots, K$, it follows that $\psi_{n,k}$, $k = 1, \ldots, K$, is given by the solution to the following multi-criterion optimization problem

$$\max_{\boldsymbol{\psi}_{n,k}} \quad \boldsymbol{\psi}_{n,k} \tag{29}$$

$$\text{s.t.} \quad \mathbf{S}(\boldsymbol{\psi}_n) \preceq 0 \tag{30}$$

$$\boldsymbol{\psi}_{n,k} \succ 0 \, ,$$

where $\mathbf{S}(\boldsymbol{\psi}_n) = (S_1(\boldsymbol{\psi}_{n,1}), \ldots, S_K(\boldsymbol{\psi}_{n,K}))^T$.

Notice that the cost function is a vector whose components are coupled by the constraints (30). Thus the problem is a multi-criterion optimization problem and each threshold $\psi_{n,k}$, $k = 1, \ldots, K$, must be optimized simultaneously. This problem is a Fast-Lipschitz optimization problem [34]. The solution is given in [33].

Now let us investigate the performance of this estimator in MSE.

Proposition 4.2. *The optimal value of $\boldsymbol{\kappa}_{n,k}$ and $\mathbf{h}_{n,k}$ are such that the error variance at node k satisfies*

$$\mathbb{E}\left[e_k - \mathbb{E}\, e_k\right]^2 < \frac{1}{\mathbf{1}^T \left(\Omega_k + \left(\sum_{j \in \mathcal{N}_k} \frac{1}{\mathbf{1}^T \Omega_j \mathbf{1}} + \frac{1}{\mathbf{1}^T \Omega_k \mathbf{1}} \sqrt{\frac{1}{\psi_{n,k}}}\right)^{-1}\right) \mathbf{1}} \, ,$$

where $\Omega_i = \mathbf{C}_k \mathbf{Q}_k^{-1} \mathbf{C}_k$.

Proof. First, by using (24) and (25) in the expression of the estimation error variance it follows that

$$\begin{aligned} \mathbb{E}\left[e_{n,k} - \mathbb{E}\, e_{n,k}\right]^2 &\leq \frac{1}{\mathbf{1}^T \left(\Gamma_{n-1,k} + \lambda_{n,k} \mathbf{I}\right)^{-1} \mathbf{1} + \mathbf{1}^T \Omega_k \mathbf{1}} \\ &< \frac{1}{\mathbf{1}^T \Omega_k \mathbf{1}} \, . \end{aligned}$$

Then notice that

$$\mathrm{tr}\Gamma_{n,k} = \sum_{j\in\mathcal{N}_k} \mathbb{E}\,(e_{n-1,j} - \,\mathbb{E}\, e_{n-1,j})^2 < \sum_{j\in\mathcal{N}_k} \frac{1}{\mathbf{1}^T\Omega_j\mathbf{1}}\,.$$

Thus we have

$$\begin{aligned}
&\ell_{\max}(\Gamma_{n-1,k} + \lambda_{n,k}\mathbf{I}) \\
&\qquad < \sum_{j\in\mathcal{N}_k} \frac{1}{\mathbf{1}^T\Omega_j\mathbf{1}} + \max\left(0, \frac{1}{\mathbf{1}^T\Omega_k\mathbf{1}}\sqrt{\frac{1}{\psi_{n,k}}} - a^2\ell_{\min}(\mathbf{P}_{n-1,k})\right) \\
&\qquad < \sum_{j\in\mathcal{N}_k} \frac{1}{\mathbf{1}^T\Omega_j\mathbf{1}} + \frac{1}{\mathbf{1}^T\Omega_k\mathbf{1}}\sqrt{\frac{1}{\psi_{n,k}}}\,.
\end{aligned}$$

Since

$$\mathbf{1}^T(\Gamma_{n-1,k} + \lambda_{n,k}\mathbf{I})^{-1}\mathbf{1} \geq \frac{1}{\mathbf{1}^T(\ell_{\max}(\Gamma_{n-1,k} + \lambda_{n,k}\mathbf{I})\mathbf{1}}\,,$$

we have that

$$\begin{aligned}
\mathbb{E}\,(e_{n,k} - \,\mathbb{E}\, e_{n,k})^2 &\leq \frac{1}{\mathbf{1}^T\left(\Gamma_{n-1,k} + \lambda_{n,k}\mathbf{I}\right)^{-1}\mathbf{1} + \mathbf{1}^T\Omega_k\mathbf{1}} \\
&< \frac{1}{\mathbf{1}^T\left(\Omega_k + \left(\sum_{j\in\mathcal{N}_k}\frac{1}{\mathbf{1}^T\Omega_j\mathbf{1}} + \frac{1}{\mathbf{1}^T\Omega_k\mathbf{1}}\sqrt{\frac{1}{\psi_{n,k}}}\right)^{-1}\right)\mathbf{1}}\,.
\end{aligned}$$

□

The previous proposition guarantees that the estimation error at each time n, in each node k, is always upper-bounded by the Cramer-Rao lower bound.

4.2.2 With Message Losses

Suppose that over a link, messages may be dropped because of bad channel conditions or radio interference. Let $\phi_{n,kj}$, with $k \neq j$, be a binary random variable associated to the message losses from sensor k to j at time n. For $k \neq j$, we assume that the random variables $\phi_{n,kj}$ are independent with probability mass function:

$$\begin{aligned}
\Pr(\phi_{n,kj} = 1) &= p_{kj}\,, \\
\Pr(\phi_{n,kj} = 0) &= q_{kj} = 1 - p_{kj}\,,
\end{aligned}$$

where $p_{kj} \in [0, 1]$ denotes the successful message reception probability.

Example 4.3. *Suppose each sensor computes an estimates $\hat{x}_i(n)$ by taking a linear combination of its own and of its neighbors' estimates and measurements. Then with the message losses, the estimator can be written as*

$$\hat{x}_{n+1,k} = \hat{x}_{n,k} + hW_k^{-1}\sum_{j\in\mathcal{N}_k}\phi_{n,kj}\left(\hat{x}_{n,j} - \hat{x}_{n,k}\right)\,,$$

where $\mathcal{N}_k$ denotes the set of neighbours of sensor k plus the node itself.

Example 4.4. *(From [35]) Suppose that the prediction phase of the Kalman Filter is independent of the observation process as Eq.*(15)*. The measurement update is stochastic as the received measurements are determined by* $\phi_{n,kj}$*.*

Then the covariance update may have following compact expression

$$P_n^{(K)} = A_n P_{n-1} A_n^T + Q_n - \sum_{k=1}^{K} \prod_{i=1}^{k} \phi_{n,ip} \prod_{j=1}^{K-k} (1-\phi_{n,jp}) \Omega_n^{(k)}(P_{n-1}) \, , \tag{31}$$

where we use the simplified notation $P_n = P_{n|n-1}$*,* $R^{(k)} = (\sum_{i=1}^{k} \sigma_i^{-1})^{-1}$*, in which* σ_n *is the variance for noise* v_n*, and*

$$\Omega_n^{(k)}(P_n) = A_n P_n C_n^T (C_n P_n C_n^T + R^{(k)})^{-1} C_n P_n A_n^T \, .$$

Intuitively, the more sensors report, the better estimation performance becomes.

Let us extend the results in Section 4.2.1 with the message losses for the scenario described by Eq.(19). Considering the message losses, each node computes estimates, instead of Eq.(20), by

$$\hat{x}_{n,k} = a\boldsymbol{\kappa}\boldsymbol{\varphi}_{n,k}^T \hat{\mathbf{x}}_{n-1} + \mathbf{h}\boldsymbol{\varphi}_{n,k}^T \left(\mathbf{y}_n - a\mathbf{C}\hat{\mathbf{x}}_{n-1}\right) \, , \tag{32}$$

where

$$\boldsymbol{\kappa}\boldsymbol{\varphi}_{n,k} = \boldsymbol{\kappa}_{n,k} \circ \boldsymbol{\varphi}_{n,k} = \left(\kappa_{n,1}, \kappa_{n,2}, \ldots, \kappa_{n,K}\right)^T \circ \left(\varphi_{n,k1}, \varphi_{n,k2}, \ldots, \varphi_{n,kK}\right)^T \, ,$$

where $\circ$ is the element-wise product between two matrices, with $\boldsymbol{\varphi}_{n,k} \in \mathbb{R}^{K\times 1}$ denotes the vector of the message reception process realization of the process $\boldsymbol{\phi}_{n,k}$ at time n, as seen from node k with respect to all nodes of the network. Specifically, let the jth element of $\boldsymbol{\varphi}_{n,k}$, with $j \neq k$, be $\varphi_{n,kj}$. Notice that at a given time instant, the j-th component of $\boldsymbol{\varphi}_{n,k}$ is zero if no data messages are received from node j. Let $\mathcal{N}_{\boldsymbol{\varphi}_k} = \{j \in \mathcal{N}_k : \varphi_{n,kj} \neq 0\}$, namely such a set collects the nodes communicating with node k at time n. The number of nodes in the set is $|\mathcal{N}_{\boldsymbol{\varphi}_k}| = \boldsymbol{\varphi}_{n,k}^T \boldsymbol{\varphi}_{n,k}$. The vector $\mathbf{h}\boldsymbol{\varphi}_{n,k} \in \mathbb{R}^{K\times 1}$ is constructed from the elements $\mathbf{h}_{n,k}$, similarly to $\boldsymbol{\kappa}\boldsymbol{\varphi}_{n,k}$.

Similarly, we extend the optimization problem in Eq.(22) to the following

$$\min_{\mathbf{g}_{n,k}, \mathbf{h}_{n,k}, \psi_{n,k}} \quad \mathbf{g}_{n,k}^T \Gamma'_{n-1,k} \mathbf{g}_{n,k} + \mathbf{h}_{n,k}^T \mathbf{Q}'_{n-1} \mathbf{h}_{n,k} \tag{33a}$$

$$\text{s.t.} \quad \left((\mathbf{g}_{n,k} + \mathbf{C}\mathbf{h}_{n,k})^T \circ \boldsymbol{\varphi}_{n,k}\right) \mathbf{1} = 1 \tag{33b}$$

$$\|\mathbf{g}_{n,k} \circ \boldsymbol{\varphi}_{n,k}\|^2 \leq \psi_{n,k} \, . \tag{33c}$$

where $\Gamma'_{n,k} = \Gamma_{n,k} \circ (\boldsymbol{\varphi}_{n,k}\boldsymbol{\varphi}_{n,k}^T)$, while $\mathbf{Q}'_n = \mathbf{Q}_n \circ (\boldsymbol{\varphi}_{n,k}\boldsymbol{\varphi}_{n,k}^T)$. The optimal weights are

$$\mathbf{g}_{n,k} = \frac{\left((\Gamma_{n-1,k} + \lambda_{n,k}\mathbf{I}) \circ \boldsymbol{\varphi}_{n,k}\boldsymbol{\varphi}_{n,k}^T\right)^{\dagger} \boldsymbol{\varphi}_{n,k}}{\boldsymbol{\varphi}_{n,k}^T \left(\left((\Gamma_{n-1,k} + \lambda_{n,k}\mathbf{I}) \circ \boldsymbol{\varphi}_{n,k}\boldsymbol{\varphi}_{n,k}^T\right)^{\dagger} + \mathbf{C}_k \mathbf{Q}_{n-1}^{-1} \mathbf{C}_k\right) \boldsymbol{\varphi}_{n,k}} \, , \tag{34}$$

$$\mathbf{h}_i(t) = \frac{\mathbf{Q}_{n-1}^{-1} \mathbf{C}_k \boldsymbol{\varphi}_{n,k}}{\boldsymbol{\varphi}_{n,k}^T \left(\left((\Gamma_{n-1,k} + \lambda_{n,k}\mathbf{I}) \circ \boldsymbol{\varphi}_{n,k}\boldsymbol{\varphi}_{n,k}^T\right)^{\dagger} + \mathbf{C}_k \mathbf{Q}_{n-1}^{-1} \mathbf{C}_k\right) \boldsymbol{\varphi}_{n,k}} \, , \tag{35}$$

$$\lambda_{n,k} = \begin{cases} 0 & \text{if } \left[\boldsymbol{\varphi}_{n,k}^T \mathbf{g}_{n,k}^T \mathbf{g}_{n,k} \boldsymbol{\varphi}_{n,k}\right]_{\lambda_{n,k}=0} \leq \psi_{n,k} \\ \lambda_{n,k}^* & \text{otherwise} \end{cases} \, . \tag{36}$$

We have following proposition for the MSE:

Proposition 4.5. *It holds*

$$\mathbb{E}_{\boldsymbol{\phi}}\,\mathbb{E}_{\mathbf{v}}[e_{n,k}-\,\mathbb{E}_{\mathbf{v}}e_{n,k}]^2 \leq \frac{a^2(\sqrt{5}-1)\sqrt{\gamma_{\max}}+2N}{(a^2+1)(\sqrt{5}-1)\sqrt{\gamma_{\max}}+2N}\cdot\sum_{i=0}^{|\mathcal{N}_k|-1}\frac{\chi(i)}{i+1}\,\sigma_{\max}^2\,.$$

Observe that the estimation error variance given by the previous proposition depends on the message loss probabilities q_{kj}, on the maximum number of neighbors for each node $|\mathcal{N}_k|$, the total number of nodes in the networks K, and the largest singular value of the matrix $\boldsymbol{\kappa}\boldsymbol{\varphi}_{n,k}$. If the number of neighbors is greater than 2, with a loss of $q_{kj}=0.3$ for all j, we have that the product of the two coefficients does not exceed 0.65 and it is only a 30% higher than the case when no packet losses are present.

5 Computational Complexity and Communication Cost

The efficiency of implementation of estimation algorithms can be characterized in terms of computational complexity and communication cost. Conventionally, the computational complexity of an algorithm is measured by the amount of basic operations such as float-point arithmetic performed. The computational complexity is commonly expressed by using the $\mathcal{O}$ notation, which describes the limiting behaviour of a function. The $\mathcal{O}$ notation suppresses the multiplicative constants and lower order terms. For example, if the time running requirement for an algorithm is at most $5n^3+100n^2$, then we say that the computational complexity is $\mathcal{O}(n^3)$. On the other hand, communication cost of an algorithm refers to the communication resources required in terms of amount and size of the exchanged messages in bytes or bits. We express the communication cost by using the $\mathcal{O}$ notation as well.

It is important to analyze the computational complexity and communication cost for distributed estimation algorithms, especially when one designs algorithms for a sensor network. In these networks, larger computational complexity requirement and communication cost always entail the high risk of slower response speed, smaller transmit rate and thus poorer performance in practice, though the theoretical performance for the algorithm might be much better. In fact, due to the limited computational capability of sensors, sometimes we have to redesign A_{alg}, or implement an approximate algorithm $\tilde{A}_{\text{alg}}$ which has lesser computational complexity but may still provide acceptable performance.

5.1 On Computational Complexity

Before deploying an algorithm for sensors networks, it is desirable to check whether the sensor nodes have enough computational capability to perform the computation as designed. Suppose that the sensors are designed to produce their estimates using some algorithm A_{alg}. We analyse A_{alg}'s worst-case computational cost requirements as a function of the size of its input (in terms of the $\mathcal{O}$-notation). Here we assume that arithmetic or basic operation with individual elements has complexity $\mathcal{O}(1)$. Thus, the computational complexity of a matrix addition, multiplication, and inversion are $\mathcal{O}(m^2)$, $\mathcal{O}(m^3)$ and $\mathcal{O}(m^3)$ respectively, where $m \times m$ is the size of the matrix.

Example 5.1. *Consider the combining estimator of Section 2.1.1. According to Proposition 2.1, some matrix inversions and multiplications to find the MMSE estimate of X are needed. Denote the largest size of the vector Y and X is M. Then the computational complexity of the estimators is $\mathcal{O}(M^3)$.*

Example 5.2. *Consider the static sensor fusion of Section 2.1.2. According to Proposition 2.2, similarly to Example 5.1, we need to perform the matrix inversions and multiplications. Notice that in this case, we need K times matrix inversions and multiplications in each iteration. Thus the computational complexity of the static sensor fusion is $\mathcal{O}(KM^3)$.*

Example 5.3. *Consider the local estimator of Section 2.2.1. According to the method, we need to perform the matrix inversions and multiplications. Notice that in this case, the size of the matrix is not M, but nM, where n is the time step. Thus the computational complexity of the estimator is $\mathcal{O}(n^3M^3)$.*

Example 5.4. *Consider the Kalman Filtering of Section 2.2.2. According to the method, we need to perform the matrix inversion and multiplications. Similar to Example 5.2, we need K times matrix inversions and multiplications per iteration. Thus computational complexity of distributed Kalman filtering is $\mathcal{O}(KM^3)$.*

Example 5.5. *Consider the computational complexity of the methods in Section 4.2.1 and 4.2.2. It is given by three components: the computational complexity of matrix operations to find the optimal weights, the computational complexity of a bisection algorithm, and the computational complexity for the estimation of the covariance matrix. To find the optimal weights $\mathbf{g}_{n,k}$ and $\mathbf{h}_{n,k}$, it is required to compute a matrix pseudo-inversion, and matrix multiplications with matrices of size $\mathcal{N}_k$.*

Matrix pseudo-inversion: *In this case, we can shrink the matrix*

$$\left((\Gamma_{n-1,k} + \lambda_{n,k}\mathbf{I}) \circ \boldsymbol{\varphi}_{n,k}\boldsymbol{\varphi}_{n,k}^T\right)$$

from the full-zero rows and columns to form a new matrix of size $\mathcal{N}_{\boldsymbol{\varphi}_k}$. Thus the complexity is still $\mathcal{O}(\mathcal{N}_{\boldsymbol{\varphi}_k}^3)$.

Matrix multiplication: *Similarly the complexity is $\mathcal{O}(\mathcal{N}_{\boldsymbol{\varphi}_k}^3)$ for these shrunk matrices.*

As a result, the computational complexity needed to find optimal weights $\mathbf{g}_{n,k}, \mathbf{h}_{n,k}$ can be obtained as

$$\mathcal{O}(\mathcal{N}_{\boldsymbol{\varphi}_k}^3) + \mathcal{O}(\mathcal{N}_{\boldsymbol{\varphi}_k}^3) + \mathcal{O}(\mathcal{N}_{\boldsymbol{\varphi}_k}^2) \sim \mathcal{O}(\mathcal{N}_{\boldsymbol{\varphi}_k}^3)\,.$$

We use the bisection method to find the optimal value for $\lambda_{n,k}$, which fulfils the following equations:

$$\left[\mathbf{g}_{n,k}^T\mathbf{g}_{n,k}\right]_{\lambda_{n,k}^*} - \psi_{n,k} = 0\,,$$

where $\lambda_{n,k}^*$ is in the interval:

$$[0, \max(0, \Lambda)]\;,$$

in which

$$\Lambda = \frac{1}{\mathbf{1}^T\mathbf{C}\mathbf{Q}^{-1}\mathbf{C}\mathbf{1}}\sqrt{\frac{1}{\psi_{n,k}} - a^2\ell(\mathbf{P}_{n-1,k})}\,.$$

Let us assume that ϵ is the required accuracy for the bisection test. Then the bisection would search at most $\log_2 \Lambda/\epsilon$ times to determine the number, since each search halves the interval. Furthermore, the complexity of searching for the bisection method is

$$\mathcal{O}\left(\mathcal{O}(\mathcal{N}^3_{\boldsymbol{\varphi}_k})\min(\log_2 \Lambda/\epsilon, \mathrm{Max}_{\mathrm{Iter}})\right)$$

in which $\mathrm{Max}_{\mathrm{iter}}$ *is a number of maximum iterations.*

From the above analysis, we can conclude that the methods used in Section 4.2.2 needs several operations. If we set $\mathrm{Max}_{\mathrm{Iter}}$ *large enough, the computational complexity is approximately $\mathcal{O}\left(\mathcal{N}^3_{\boldsymbol{\varphi}_k}\log_2 \Lambda/\epsilon\right)$.*

5.2 On Communication Cost

Due to the limited communication resources for the network, before implementing a distributed estimator, we need to analyze the communication cost as well. We define the number of messages exchanged by the sensors as the communication cost.

Example 5.6. *In the network with star topology mentioned in the Section 2, every sensor need sharing sending messages to the center fusion. Thus the total communication cost is $\mathcal{O}(K)$ for each iteration, where K is the number of sensors in the network.*

Example 5.7. *In the network with arbitrary topology mentioned in the Section 4, every sensor needs sending its messages with its neighbors. Since via wireless communication channels, sensor can broadcast its messages to all sensors inside its communication range, the total communication cost is $\mathcal{O}(K)$ per iteration.*

5.3 Summary of the computational complexity and communication cost

In this section, we have studied the computational complexities and communication cost for the distributed estimation methods summarized in the preceding sections. Now, we summarize the result in Table 1.

In Table 1, **Signal** represents the signal tracked by the sensors network, **Complexity** represents the computational complexity, whereas **Cost** represents the communication cost. It is worth mentioning that the R in the **Cost** represents the extra communication cost used for routing when the sensors only have limited communication range.

The table shows that without considering routing, the communication costs are approximately same for the networks with star and arbitrary topology. However, the computational complexities vary for different algorithms. Generally, transmitting local estimation (in Section 2.2.1) needs most running time compared to other algorithms used in star topology. Moreover, dynamic sensor fusion (in Section 4.2.2) needs more running time than the static sensor fusion.

Table 1: Summary of the time complexity and communication cost. In the table, M is the at most size of the tracked signal, n is the time step, while K is the number of the sensors in the network. R represents the extra communication cost for the routing, which will be explained later.

Topology	**Signal**	**Algorithm**	**Complexity**	**Cost**
Star	$\mathbb{R}^M$	Combining Estimators in Section 2.1.1	$\mathcal{O}(M^3)$	$\mathcal{O}(K) + R$
	$\mathbb{R}^M$	Static Sensor Fusion in Section 2.1.2	$\mathcal{O}(KM^3)$	$\mathcal{O}(K) + R$
	$\mathbb{R}^M$	Transmission Local in Section 2.2.1	$\mathcal{O}\left((n+M)^3\right)$	$\mathcal{O}(K) + R$
	$\mathbb{R}^M$	Distributed Kalman Filter in Section 2.2.2	$\mathcal{O}(KM^3)$	$\mathcal{O}(K) + R$
Arbitrary	$\mathbb{R}^1$	Static Sensor Fusion in Section 4.1	$\mathcal{O}(\mathcal{N}_k^3)$	$\mathcal{O}(K)$
	$\mathbb{R}^1$	Dynamic Sensor Fusion in Section 4.2.2	$\mathcal{O}(\mathcal{O}(\mathcal{N}_k^3) \log_2 \Lambda/\epsilon)$	$\mathcal{O}(K)$

6 Conclusion

This chapter introduced basic notions of distributed estimation theory, and some implications for the applications with and without considering the limitations in the networks. Moreover, an analysis of the computational complexity and communication cost of these distributed algorithms was performed. Generally, the less the limited capability of the network and the greater the knowledge of the physical phenomenon, the lower the complexity of the resulting estimators. Nevertheless, it is often possible to establish accurate distributed estimators in the networks. We remark that, this study we gave here is an essential overview on some key aspects of distributed estimation. Much more can be summarized (e.g., the convergence or consensus properties of the distributed estimators).

References

[1] A. Willsky, M. Bello, D. Castanon, B. Levy, and G. Verghese, “Combining and updating of local estimates and regional maps along sets of one-dimensional tracks,” *Automatic Control, IEEE Transactions on*, vol. 27, no. 4, pp. 799 – 813, aug 1982.

[2] Y. Bar-Shalom, “On the track-to-track correlation problem,” *Automatic Control, IEEE Transactions on*, vol. 26, no. 2, pp. 571 – 572, apr 1981.

[3] Y. Bar-Shalom and L. Campo, “The effect of the common process noise on the two-sensor fused-track covariance,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. AES-22, no. 6, pp. 803 –805, nov. 1986.

[4] J. Roecker and C. McGillem, “Comparison of two-sensor tracking methods based on state vector fusion and measurement fusion,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 24, no. 4, pp. 447 –449, jul 1988.

[5] K. Chang, R. Saha, and Y. Bar-Shalom, “On optimal track-to-track fusion,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 33, no. 4, pp. 1271 –1276, oct. 1997.

[6] S. Mori, W. Barker, C.-Y. Chong, and K.-C. Chang, “Track association and track fusion with nondeterministic target dynamics,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 38, no. 2, pp. 659 –668, april 2002.

[7] C. Y. Chong, “Hierarchical estimation,” in *Proceedings of the 2nd MIT/ONR C^3 Workshop*, 1979.

[8] C. B. C. D. Willner and K. P. Dunn, “Kalman filter algorithms for a multisensor system,” in *Proceedings of the 15th Conference on Decision and Control*, 1976.

[9] M. Hassan, G. Salut, M. Singh, and A. Titli, “A decentralized computational algorithm for the global kalman filter,” *Automatic Control, IEEE Transactions on*, vol. 23, no. 2, pp. 262 – 268, apr 1978.

[10] J. Speyer, “Computation and transmission requirements for a decentralized linear-quadratic-gaussian control problem,” *Automatic Control, IEEE Transactions on*, vol. 24, no. 2, pp. 266 – 269, apr 1979.

[11] B. C. Levy, D. A. Castañon, G. C. Verghese, and A. S. Willsky, “A scattering framework for decentralized estimation problems,” *Automatica*, vol. 19, no. 4, pp. 373 – 384, 1983.

[12] C.-Y. Chong, S. Mori, and K.-C. Chang, “Information fusion in distributed sensor networks,” in *American Control Conference, 1985*, june 1985, pp. 830 –835.

[13] H. Hashemipour, S. Roy, and A. Laub, “Decentralized structures for parallel kalman filtering,” *Automatic Control, IEEE Transactions on*, vol. 33, no. 1, pp. 88 –94, jan 1988.

[14] N. Carlson, “Federated square root filter for decentralized parallel processors,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 26, no. 3, pp. 517 –525, may 1990.

[15] B. Rao and H. Durrant-Whyte, “Fully decentralised algorithm for multisensor kalman filtering,” *Control Theory and Applications, IEE Proceedings D*, vol. 138, no. 5, pp. 413 –420, sep 1991.

[16] T. Berg and H. Durrant-Whyte, “Distributed and decentralized estimation,” in *Intelligent Control and Instrumentation, 1992. SICICI ’92. Proceedings., Singapore International Conference on*, vol. 2, feb 1992, pp. 1118 –1123.

[17] O. E. Drummond, “Tracklets and a hybrid fusion with process noise,” in *Proceedings of the SPIE, Signal and Data Processing of Small Targets*, 1997.

[18] I. Liggins, M.E., C.-Y. Chong, I. Kadar, M. Alford, V. Vannicola, and S. Thomopoulos, “Distributed fusion architectures and algorithms for target tracking,” *Proceedings of the IEEE*, vol. 85, no. 1, pp. 95 – 107, jan. 1997.

[19] C.-Y. Chong, S. Mori, W. Barker, and K.-C. Chang, "Architectures and algorithms for track association and fusion," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 15, no. 1, pp. 5 –13, jan 2000.

[20] V. Gupta, B. Hassibi, and R. M. Murray, "Optimal LQG control across packet-dropping links," *Systems and Control Letters*, vol. 56, no. 6, pp. 439–446, june 2007.

[21] V. Gupta, N. C. Martins, and J. S. Baras, "Stabilization over erasure channels using multiple sensors," *IEEE Transactions on Automatic Control*, vol. 54, no. 7, pp. 1463–1476, july 2009.

[22] P. Hovareshti, V. Gupta, and J. S. Baras, "On sensor scheduling using smart sensors," in *Proceedings of the IEEE Conference on Decision and Control (CDC), 2007*, December 2007.

[23] V. Gupta and N. C. Martins, "On fusion of information from multiple sensors in the presence of analog erasure links," in *Proceedings of the IEEE Conference on Decision and Control (CDC), 2009*, December 2009.

[24] Z. Q. Luo, "Universal Decentralized Estimation in a Bandwidth Constrained Sensor Network," *IEEE Transactions on Information Theory,* vol. 51, no. 6, pp. 2210–2219, 2005.

[25] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, and G. Giannakis, "Distributed compression-estimation using wireless sensor networks," *Signal Processing Magazine, IEEE*, vol. 23, no. 4, pp. 27 – 41, july 2006.

[26] Z. Q. Luo, "An Isotropic Universal Decentralized Estimation Scheme for a Bandwidth Constrained Ad Hoc Sensor Network," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 4, pp. 735–744, 2005.

[27] A. Ribeiro, G. B. Giannakis, and S. Roumeliotis, "SOI-KF : Distributed Kalman Filtering with Low-cost Communications using the Sign of Innovations," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, no. 2, Toulouse, 2006, pp. 153–156.

[28] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1520 – 1533, sept. 2004.

[29] W. Ren and R. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *Automatic Control, IEEE Transactions on*, vol. 50, no. 5, pp. 655 – 661, may 2005.

[30] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *Automatic Control, IEEE Transactions on*, vol. 48, no. 6, pp. 988 – 1001, june 2003.

[31] D. Spanos, R. Olfati-Saber, and R. Murray, "Dynamic consensus on mobile networks," in *Information Processing in IFAC World Congress*, 2006.

[32] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, april 2005, pp. 63 – 70.

[33] A. Speranzon, C. Fischione, K. Johansson, and A. Vincentelli-Sangiovanni, "A distributed minimum variance estimator for sensor networks," *IEEE Journal on Selected Areas in Communications*, pp. 609–621, 2008.

[34] C. Fischione, "Fast-lipschitz optimization with wireless sensor networks applications," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2319–2331, October 2011.

[35] B. Zhu, B. Sinopoli, K. Poolla, and S. Sastry, "Estimation over Wireless Sensor Networks," in *American Control Conference*, 2007, pp. 2732–2737.

[36] H. Sandberg, "Ncs: Kalman filtering." [Online]. Available: http://www.cds.caltech.edu/murray/wiki/index.php?title=NCS:_\Kalman_Filtering

A Appendix

A.1 Optimal mean square estimate of a random variable

We will be interested in Minimum Mean Square Error (MMSE) estimates. Given a random variable Y that depends on another random variable X, obtain the estimate $\hat{X}$ such that the mean square error given by $\mathbb{E}\,[X-\hat{X}]^2$ is minimized. The expectation is taken over the random variables X and Y.

Proposition A.1. *(Lemma 1 in Henrik's Kalman Filtering Lecture [36]): The minimum mean square error estimate is given by the conditional expectation* $\mathbb{E}\,[X|Y=y]$.

Proof. The arguments are standard. Consider the functional form of the estimator as $g(Y)$. Let $f_{X,Y}(x,y)$ denote the joint probability density function of X and Y. Then the cost function C is given by

$$\begin{aligned}\mathbb{E}\left[X-\hat{X}\right]^2 &= \int_x\int_y (x-g(y))^2 f_{X,Y}(x,y)dxdy\\ &= \int_y dy f_Y(y)\int_x (x-g(y))^2 f_{X|Y}(x|y)dx.\end{aligned}$$

Now consider the derivative of the cost function with respect to the function $g(y)$.

$$\begin{aligned}\frac{\partial C}{\partial g(y)} &= \int_y dy f_Y(y)\int_x 2(x-g(y))f_{X|Y}(x|y)dx\\ &= 2\int_y dy f_Y(y)(g(y)-\int_x x f_{X|Y}(x|y)dx)\\ &= 2\int_y dy f_Y(y)(g(y)-\mathbb{E}\,[X|Y=y]).\end{aligned}$$

Thus the only stationary point is $g(y)=\mathbb{E}\,[X|Y=y]$. Moreover it is easy to see that it is a minimum. □

The result holds for vector random variables as well.

MMSE estimates are important because for *Gaussian* variables, they coincide with the Maximum Likelihood (ML) estimates. Of course, for non-Gaussian random variables, other notions of optimality may be better. (Recall Moving Horizon Estimation [36]).

It is also a standard result that for Gaussian variables, the MMSE estimate is linear in the state value. Proof was given in the lecture on Kalman filtering. So we will restrict our attention to linear estimates now. Also, from now on we will assume zero mean values for all the random variables. All the results can however be generalized. The covariance of X will be denoted by R_X and the cross-covariance between X and Y by R_{XY}.

Proposition A.2. *The best linear MMSE estimate of X given $Y = y$ is*

$$\hat{x} = R_{XY} R_Y^{-1} y,$$

with the error covariance

$$P = R_X - R_{XY} R_Y^{-1} R_{YX}.$$

Proof. Let the estimate be $\hat{x} = Ky$. Then the error covariance is

$$\begin{aligned} C &= \mathbb{E}\left[(x-Ky)(x-Ky)^T\right] \\ &= R_X - KR_{YX} - R_{XY}K^T + KR_YK^T. \end{aligned}$$

Differentiating C w.r.t. K and setting it equal to zero yields

$$-2R_{XY} + 2KR_Y^{-1} = 0.$$

The result follows immediately. □

In the standard control formulations, we are also interested in measurements that are related linearly to the variable being estimated (usually the state).

Proposition A.3. *Let $y = Hx + v$, where H is a matrix and v is a zero mean Gaussian noise with covariance R_V independent of X. Then the MMSE estimate of X given $Y = y$ is*

$$\hat{x} = R_X H^T \left(HR_XH^T + R_V\right)^{-1} y,$$

with the corresponding error covariance

$$P = R_X - R_X H^T \left(HR_XH^T + R_V\right)^{-1} HR_X.$$

Proof. Follows immediately by evaluating the terms R_{XY} and R_Y and substituting in the result of Proposition A.2. □

A.2 Matrix Inversion Formula

Proposition A.4. *For compatible matrices A, B, C and D,*

$$(A+BCD)^{-1} = A^{-1} - A^{-1}B\left(C^{-1} + DA^{-1}B\right)^{-1} DA^{-1},$$

assuming the inverses exist.

Proof. Begin by considering the block matrix

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

By doing the LDU and UDL decomposition of M and equating them, we obtain

$$\begin{bmatrix} I & 0 \\ CA^{-1} & 0 \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & D - CA^{-1}B \end{bmatrix} \begin{bmatrix} I & A^{-1}B \\ 0 & I \end{bmatrix} \\ = \begin{bmatrix} I & BD^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A - BD^{-1}C & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} I & 0 \\ D^{-1}C & I \end{bmatrix}.$$

Thus inverting both sides yields

$$\begin{bmatrix} I & -A^{-1}B \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & \left(D - CA^{-1}B\right)^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -CA^{-1} & 0 \end{bmatrix} \\ = \begin{bmatrix} I & 0 \\ -D^{-1}C & I \end{bmatrix} \begin{bmatrix} \left(A - BD^{-1}C\right)^{-1} & 0 \\ 0 & D^{-1} \end{bmatrix} \begin{bmatrix} I & -BD^{-1} \\ 0 & I \end{bmatrix}.$$

Equating the $(1,1)$ block shows

$$\left(A - BD^{-1}C\right)^{-1} = A^{-1} + A^{-1}B\left(D - CA^{-1}B\right)^{-1} CA^{-1}.$$

Finally substituting $C \to -D$ and $D \to C^{-1}$, we obtain

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B\left(C^{-1} + DA^{-1}B\right)^{-1} DA^{-1}.$$

□