In Proceedings of the American Control Conference, Seattle, WA, May 2017

On distributed linear filtering with noisy communication

Anthony J. Savas[1], Vaibhav Srivastava[2], and Naomi Ehrich Leonard[1]

***Abstract*— We consider distributed filtering of a scalar linear stochastic process under communication corrupted by Gaussian noise. We investigate how communication noise degrades the performance of an existing distributed algorithm and develop a novel algorithm that mitigates these problems. We rigorously investigate the properties of the new distributed estimator and discuss optimal tuning of (fixed) gains that minimize the asymptotic error covariance. We demonstrate the effectiveness of our algorithm through numerical simulations.**

I. INTRODUCTION

Distributed estimation is a problem of fundamental interest in a variety of problems ranging from robotic networks, transportation networks, power networks, and synthetic biological networks. With increasing deployment of networked multiagent systems the algorithms for distributed estimation are of increasing importance. Some of the desired features of these algorithms include scalability, adaptability, and resilience.

In this paper, we investigate the problem of distributed filtering in a networked multiagent system of a scalar linear stochastic process under communication corrupted by Gaussian noise. There is a significant and growing literature on distributed filtering in networked systems [1], [2], [3], [4], [5], [6], [7], [8]. Typically, however, these works assume no communication noise. We design algorithms that are robust to the communication noise in such networks.

Distributed filtering in a networked multiagent system is designed to allow each individual agent to improve its estimate of the state of a dynamical system by sharing measurements or estimates through a communication network. In consensus-based distributed filtering, agents update their estimates with measurements or estimates communicated from others using linear consensus dynamics [9], [10]. Olfati-Saber [3] considered distributed linear filtering with two consensus dynamics: one for weighted measurements and one for precision matrices, see [11] for related work. Distributed linear filtering in continuous time was examined in [4]. Spanos *et al.* [12] investigated the distributed least-squares estimation problem using consensus dynamics. Speranzon *et al.* [13] studied distributed linear filtering of a noisy time-varying signal using adaptive time-varying consensus.

In the context of robotic networks, cooperative Kalman filtering techniques have been used to explore noisy scalar fields in the plane [8]. Lynch *et al.* [14] studied the problem of information maximization in a scalar uncertain field using optimal filtering and consensus techniques.

We investigate the problem of consensus-based distributed filtering under *noisy communication*. The robustness of consensus dynamics under noisy communication has been studied in [7], [15] and in the context of decision-making [16], [17].

The consensus algorithm has its root in the sociology literature and is the same as the famous DeGroot model [18]. The modification to the consensus protocols that we propose to mitigate effects of communication noise has similarities with the DeGroot-Friedkin model in sociology [19]. The analysis in this paper suggests that the DeGroot-Friedkin model may have superior robustness properties under noisy communication.

To address the problem of distributed filtering of a scalar linear stochastic process under noisy communication, we build upon the algorithm proposed by Carli *et al.* in [6]. They propose an algorithm comprising discrete-time sampling of the noisy process and a fixed number of consensus rounds between sampling instances. We develop a new algorithm that mitigates the effect of communication noise on the performance of the distributed filter. The major challenge consensus-based strategies face under noisy communication is the presence of integrator dynamics in consensus protocols which aggregate noise over time leading to large variances and poor estimation performance. Here, we design novel consensus dynamics that alleviate this problem.

The major contributions of this paper are threefold. First, we examine the algorithm proposed in [6] for distributed filtering of a scalar linear stochastic process and show how the performance of this algorithm degrades under noisy communication. Second, we build upon [6] to develop a novel algorithm that mitigates the effects of noisy communication. Third, we rigorously analyze the new algorithm and discuss methods to tune its parameters to optimize performance.

The remainder of the paper is organized as follows. In §II, we formally pose the distributed linear filtering problem. In §III, we recall the distributed filtering algorithm from [6] and study its performance under noisy communication. In §IV, we develop a novel algorithm to provide robustness to communication noise. We analyze this algorithm and illustrate in §V, and we conclude in §VI.

II. PROBLEM SETUP

Consider the following scalar linear stochastic process

$$x(k+1) = ax(k) + w(k), \quad x(0) = X_0, \tag{1}$$

This research has been supported in part by ONR grant N00014-14-1-0635, ARO grant W911NF-14-1-0431 and NSF grant ECCS-1135724.

[1]A.J. Savas and N.E. Leonard are with the Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ, USA {asavas, naomi}@princeton.edu

[2]V. Srivastava is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA vaibhav@egr.msu.edu

for each $k \in \mathbb{Z}_{\geq 0}$, where $a \in \mathbb{R}$ is a constant, $\{w(k)\}_{k \in \mathbb{Z}_{\geq 0}}$ is a sequence of i.i.d. zero-mean Gaussian noise with variance $q \in \mathbb{R}_{>0}$, and X_0 is a Gaussian random variable with mean x_0 and variance σ. Suppose a sensor samples this process at each time k to obtain a noisy measurement

$$y(k) = x(k) + n(k), \quad \text{for each } k \in \mathbb{Z}_{\geq 0}, \tag{2}$$

where $\{n(k)\}_{k \in \mathbb{Z}_{\geq 0}}$ is a sequence of i.i.d. zero-mean Gaussian noise with variance $r \in \mathbb{R}_{>0}$. The estimation of state $x(k)$ in (1) using measurements $y(k)$ in (2) is the standard scalar Kalman filtering problem [20].

We consider the problem of distributed estimation of the state $x(k)$ using multiple communicating agents. For simplicity, we assume $a = 1$ but our analysis is generalizable to the case $a \neq 1$. Specifically, we consider the estimation of the white noise process

$$x(k+1) = x(k) + w(k), \quad x(0) = X_0. \tag{3}$$

We consider a multiagent network in which agents can communicate over a fixed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{1, \ldots, N\}$ is the vertex set, $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ is the edge set, and N is the total number of agents. We assume that the graph is undirected and connected in the sense that there exists a path from each node to every other node. We assume that each agent $i \in \{1, \ldots, N\}$ samples the process (3) at times k and collects a noisy measurement $y_i(k)$ of the process $x(k)$ defined by

$$y_i(k) = x(k) + n_i(k), \quad \text{for each } i \in \{1, \ldots, N\}, \tag{4}$$

where $\{n_i(k)\}_{k \in \mathbb{Z}_{\geq 0}}$ are i.i.d. zero-mean Gaussian noises with variance r. We further assume the noise sequences $n_i(k)$ are independent for different $i \in \{1, \ldots, N\}$. We can write (4) in vector form as

$$\boldsymbol{y}(k) = x(k)\mathbf{1}_N + \boldsymbol{n}(k), \tag{5}$$

where $\boldsymbol{y}(k)$ and $\boldsymbol{n}(k)$ are the N-column vectors of $y_i(k)$'s and $n_i(k)$'s, respectively, and $\mathbf{1}_N$ is the N-column vector of all ones.

We focus on consensus-based dynamics for distributed estimation [9], [10], [21]. However, in contrast to standard approaches to this problem we assume that the communication among agents is noisy. We recall that in consensus dynamics each agent at each (discrete) time averages its state with its neighbors in the communication graph [9], [18], [22]. Here we assume that each agent receives a noisy estimate of the state of each of its neighbors and it uses these noisy estimates in the consensus dynamics. Let Q be the consensus matrix, i.e., the matrix of the (convex) weights an agent i assigns to its neighbor j. Then the consensus dynamics with noisy communication are

$$\boldsymbol{z}(l+1) = Q\boldsymbol{z}(l) + \sigma_c \boldsymbol{u}(l), \tag{6}$$

where $\boldsymbol{z}(l)$ is the vector of states of agents at times $l \in \mathbb{Z}_{\geq 0}$, σ_c^2 is the variance of the communication noise, and $\{\boldsymbol{u}(l)\}_{l \in \mathbb{Z}_{\geq 0}}$ is the sequence of i.i.d. N-variate zero-mean Gaussian random vectors with covariance $\mathcal{I}_N$, where $\mathcal{I}_N$ is the identity matrix of order N. Here for simplicity, we have assumed that the communication noise for each agent has the same variance σ_c^2.

It is well-known that the matrix Q is row stochastic and for a connected undirected graph is irreducible, i.e., the matrix Q has only one simple eigenvalue at unity and every other eigenvalue is inside the unit disk [9], [18], [21], [22]. Moreover, the eigenvalue at unity corresponds to the eigenvector $\frac{1}{\sqrt{N}}\mathbf{1}_N$. We assume that Q is doubly stochastic and denote its eigenvalues as $\{\lambda_0, ..., \lambda_{N-1}\}$ with $\lambda_0 = 1$.

III. A STATE-OF-THE-ART DISTRIBUTED LINEAR FILTER

The estimation problem posed in §II was studied by Carli *et al.* [6] and they proposed a two-stage algorithm that we summarize in this section. We then apply their algorithm to the setting of noisy communication and observe that it is no longer stabilizing. This is to be expected since consensus dynamics have one eigenvalue at unity and consequently integrate noise. The integrated noise has asymptotically infinite variance. So any strategy that is designed for noise-free communication doesn't immediately extend to noisy communication.

A. A two-stage distributed linear filter under noise-free communication

In this section we recall the two-stage distributed linear filter proposed in [6]. During the first stage, at time k each agent i computes the estimate of process $x(k)$ given measurements until time k, i.e., $\hat{x}_i(k|k)$, by computing a convex combination of the predictive estimate of the current state using observations until time $k-1$, i.e., $\hat{x}_i(k|k-1)$ and the current observation $y_i(k)$. Formally, the first stage updates the state as

$$\hat{\boldsymbol{x}}(k|k) = (1-\ell)\hat{\boldsymbol{x}}(k|k-1) + \ell\boldsymbol{y}(k), \tag{7}$$

where $\hat{\boldsymbol{x}}(k|k)$ and $\hat{\boldsymbol{x}}(k|k-1)$ are vectors of $\hat{x}_i(k|k)$ and $\hat{x}_i(k|k-1)$, respectively, and $\ell \in [0,1]$ is the gain. Note that, unlike the optimal Kalman filter, here the gain ℓ is assumed constant. This means that the resulting filter is not necessarily optimal. However, as shown in [6] this leads to a bounded variance of estimation error and, hence, the choice of constant ℓ is stabilizing.

The second stage comprises m rounds of the consensus dynamics (6) between two consecutive time instances k and $k+1$ using local estimates $\hat{x}_i(k|k)$. The consensus dynamics ensure that the local estimate $\hat{x}_i(k+1|k)$ of each agent converges towards the average of the group $\frac{1}{N}\sum_{j=1}^{N}\hat{x}_j(k|k)$. Formally, the second stage is

$$\hat{\boldsymbol{x}}\Big(k + \frac{h}{m}\Big|k\Big) = Q\hat{\boldsymbol{x}}\Big(k + \frac{(h-1)}{m}\Big|k\Big), \quad h \in \{1, \ldots, m\}, \tag{8}$$

Here, a timescale separation between process dynamics and consensus dynamics is assumed, i.e., the communication and consensus dynamics are much faster than the process dynamics. Note that at the end of the consensus rounds, update (8) yields $\hat{x}(k+1|k)$ that can be used with update (7) to compute $\hat{x}(k+1|k+1)$. The distributed linear filtering algorithm is initialized with $\hat{\boldsymbol{x}}(0|-1) = x_0\mathbf{1}_N$ and the estimates at future times are computed recursively using (7) and (8).

Carli *et al.* [6] computed the covariance of the estimation error for the above algorithm and used it to find the optimal ℓ that minimizes the trace of the asymptotic error covariance matrix.

The above algorithm is easy to implement and under noise-free communication is stabilizing, i.e., always leads to bounded error covariance. In the next section, we investigate the performance of this algorithm under noisy communication.

B. Performance under noisy communication

We now consider the two-stage distributed linear filtering algorithm in §III-A with noisy communication in the consensus dynamics. The first stage of the algorithm remains identical to update (7). In the second stage the update (8) is replaced by noisy consensus dynamics

$$\hat{\boldsymbol{x}}\Big(k+\frac{h}{m}\Big|k\Big) = Q\hat{\boldsymbol{x}}\Big(k+\frac{(h-1)}{m}\Big|k\Big) + \sigma_c\boldsymbol{u}\Big(k+\frac{h}{m}\Big), \quad (9)$$

where $\boldsymbol{u}(k+h/m)$ is the N-variate zero-mean Gaussian noise with covariance $\mathcal{I}_N$, for each $k \in \mathbb{Z}_{\geq 0}$ and $h \in \{1,\ldots,m\}$, $\boldsymbol{u}(k+h/m)$ are independent, and σ_c^2 is the communication noise variance. The estimation error at time k is defined by

$$\tilde{\boldsymbol{x}}(k|k-1) = x(k)\mathbf{1}_N - \hat{\boldsymbol{x}}(k|k-1). \quad (10)$$

We numerically investigate the performance of the two-stage algorithm described in §III-A under noisy communication. Consider a set of three agents $\{1,2,3\}$ communicating over an undirected line graph. Let $Q = \mathcal{I}_3 - \epsilon L$, where $\mathcal{I}_3$ is the identity matrix and $\epsilon = 0.4$ is a constant.

We examine ten time instances of the process (3), i.e., $k \in \{0,...,9\}$, and between each consecutive pair of time instances we apply m consensus rounds. We illustrate performance for values of m from 0 to 5. We assume the process noise variance is $q = 1$ and the measurement noise variance is $r = 25$.

We employ the distributed filtering algorithm (7) and (9) with convexity parameter $\ell = 0.25$. We performed $200,000$ Monte-Carlo simulations to estimate the trace of the error covariance matrix. Fig. 1 shows the trace of the error covariance matrix for $k = 4$, which can be represented as $\sum_{i=1}^{3}\mathrm{var}(\tilde{x}_i(4|3))$, as a function of the number of consensus rounds m for a range of values of σ_c. It can be seen that for large enough values of σ_c the trace of the error covariance actually increases as more consensus rounds are performed, suggesting that the two-stage estimation algorithm in §III-A is not stabilizing, i.e., the trace of the error covariance diverges as the number of consensus rounds are increased.

This is not totally unexpected because the consensus dynamics are inherently non-robust due to the presence of an eigenvalue of unity. This eigenvalue at unity acts as an integrator and integrates noise. As we integrate more and more noise the covariance of the system diverges.

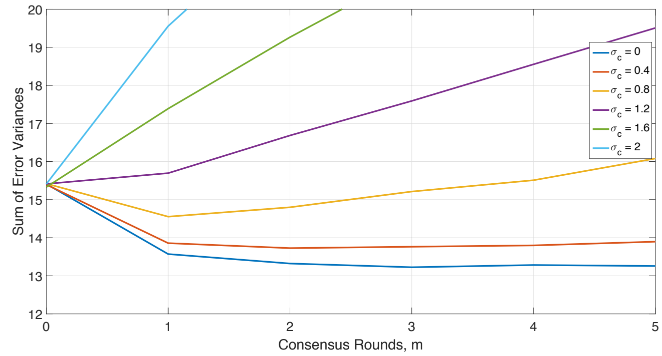

Fig. 1. Influence of communication noise in consensus dynamics on error variance across 200,000 Monte Carlo runs for distributed filtering algorithm (7) and (9) with $N = 3$, $r = 25$, and $q = 1$ for an undirected line graph. We see that the error variance diverges with the number of consensus rounds.

IV. A NOVEL TWO-STAGE DISTRIBUTED LINEAR FILTER

As discussed in the previous section, the filter in §III-A suffers under noisy communication. In this section, we modify the algorithm of Carli *et al.* [6] to mitigate the effects of noisy communication.

We keep the update in the first stage of the algorithm the same as in (7), i.e.,

$$\hat{\boldsymbol{x}}(k|k) = (1-\ell)\hat{\boldsymbol{x}}(k|k-1) + \ell\boldsymbol{y}(k), \quad (11)$$

with $\hat{\boldsymbol{x}}(0|-1) = x_0\mathbf{1}_N$.

We modify the second stage, i.e., the consensus dynamics, in the following way. We define $\boldsymbol{z}(k|k) = \hat{\boldsymbol{x}}(k|k)$ for each $k \in \mathbb{Z}_{\geq 0}$. We update $\boldsymbol{z}$ through m consensus rounds between consecutive time instances k and $k+1$ as follows:

$$\boldsymbol{z}\Big(k+\frac{h}{m}\Big|k\Big) = Q\boldsymbol{z}\Big(k+\frac{(h-1)}{m}\Big|k\Big)+\sigma_c\boldsymbol{u}\Big(k+\frac{h}{m}\Big)+\hat{\boldsymbol{x}}(k|k). \quad (12)$$

In (12), each agent $i \in \{1,\ldots,N\}$ remembers its estimate $\hat{x}_i(k|k)$ at time k and re-injects it at each consensus round. Loosely speaking, the intuition for such an update is that starting from a deterministic initial condition $\boldsymbol{z}(k|k) = \hat{\boldsymbol{x}}(k|k)$ and after m rounds of consensus the dominating component of the variance of $\boldsymbol{z}(k+1|k)$ is $m\sigma_c^2$ (see Fig. 1). By re-injecting $\hat{\boldsymbol{x}}(k|k)$ at each step, we ensure that the dominating component of the expected value of $z_i(k+1|k)$ is $\frac{m+1}{N}\sum_{j=1}^{N}\hat{x}_j(k|k)$, for each $i \in \{1,\ldots,N\}$. Finally, if we divide $\boldsymbol{z}(k+1)$ by $(m+1)$, the resulting mean is $\frac{1}{N}\sum_{j=1}^{N}\hat{x}_j(k|k)$ and variance is $m\sigma_c^2/(m+1)^2$ which goes to 0 as $m \to +\infty$. Thus, for large m we recover the performance of the noise-free algorithm. However, if m is small noise still degrades performance, so we set the update $\hat{\boldsymbol{x}}(k+1|k)$ as the convex sum of $\hat{\boldsymbol{x}}(k|k)$ and $z(k+1|k)$ as below

$$\hat{\boldsymbol{x}}(k+1|k) = \zeta\hat{\boldsymbol{x}}(k|k) + (1-\zeta)\frac{\boldsymbol{z}(k+1|k)}{m+1}, \quad (13)$$

where $\zeta \in [0,1]$ is a constant. ζ trades off the variance of the two estimators $\hat{\boldsymbol{x}}(k|k)$ and $\boldsymbol{z}(k+1|k)$. Thus, for large m we can choose ζ close to 0 and for small m we can choose ζ close to 1.

In contrast to the distributed filtering algorithm in [6] which has only one tunable parameter ℓ, our algorithm has two tunable parameters ℓ and ζ. Similar to [6], these parameters can be chosen to minimize asymptotic error covariance of the estimator. Towards this end, we analyze the error covariance of the new algorithm in the next section.

V. Analysis of the novel two-stage distributed linear filter

In this section we analyze the properties of the novel distributed linear filter proposed in §IV. We first derive an expression for the asymptotic error covariance and then analyze its properties. Our analysis follows similarly to [6].

A. Error covariance of the estimator

We define the predictive and posterior errors as

$$\tilde{\boldsymbol{x}}(k+1|k) = x(k+1)\mathbf{1}_N - \hat{\boldsymbol{x}}(k+1|k),$$
$$\text{and} \quad \tilde{\boldsymbol{x}}(k+1|k+1) = x(k+1)\mathbf{1}_N - \hat{\boldsymbol{x}}(k+1|k+1), \tag{14}$$

respectively. Let

$$P(k+1|k) = \mathbb{E}[\tilde{\boldsymbol{x}}(k+1|k)\tilde{\boldsymbol{x}}(k+1|k)^\top]$$
$$\text{and} \quad P(k+1|k+1) = \mathbb{E}[\tilde{\boldsymbol{x}}(k+1|k+1)\tilde{\boldsymbol{x}}(k+1|k+1)^\top]$$

be predictive and posterior error covariance matrices. We are now ready to state the main result of this section.

*Theorem 1 (**Asymptotic Error Covariance**):* For the scalar linear stochastic dynamics (3) and the distributed linear filtering algorithm with noisy communication defined by (11), (12) and (13), the following statements hold:

(i) the asymptotic error covariance is

$$\begin{aligned}&\lim_{k\to\infty} P(k|k-1)\\ &= \ell^2 r\sum_{i=0}^{\infty}(1-\ell)^{2i}Q^{\dagger(i+1)}(Q^{\dagger(i+1)})^\top + \frac{q}{1-(1-\ell)^2}\mathbf{1}_N\mathbf{1}_N^\top\\ &+\left(\frac{1-\zeta}{m+1}\right)^2\sigma_c^2\sum_{i=0}^{\infty}(1-\ell)^{2i}\sum_{j=0}^{m-1}Q^{\dagger i}Q^j(Q^j)^\top(Q^{\dagger i})^\top,\end{aligned} \tag{15}$$

where $Q^\dagger = \zeta\mathcal{I}_N + \left(\frac{1-\zeta}{m+1}\right)\sum_{i=0}^m Q^i$;

(ii) the trace of the asymptotic covariance matrix is

$$\begin{aligned}\operatorname{tr}\Big(\lim_{k\to\infty}P(k|k-1)\Big) &= \frac{\ell^2 r + qN + \dfrac{(1-\zeta)^2\sigma_c^2 m}{(m+1)^2}}{1-(1-\ell)^2}\\ &+\ell^2 r\sum_{h=1}^{N-1}\frac{\left|\left(\frac{1-\zeta}{m+1}\right)\bar{\lambda}_h+\zeta\right|^2}{1-(1-\ell)^2\left|\left(\frac{1-\zeta}{m+1}\right)\bar{\lambda}_h+\zeta\right|^2}\\ &+\left(\frac{1-\zeta}{m+1}\right)^2\sigma_c^2\sum_{h=1}^{N-1}\frac{\left(\dfrac{1-|\lambda_h|^{2(m-1)}}{1-|\lambda_h|^2}\right)}{1-(1-\ell)^2\left|\left(\frac{1-\zeta}{m+1}\right)\bar{\lambda}_h+\zeta\right|^2},\end{aligned} \tag{16}$$

where $\bar{\lambda}_h = \sum_{n=0}^m \lambda_h^n$.

Proof: From (14) and (11) it follows that

$$\tilde{\boldsymbol{x}}(k|k) = (1-\ell)\tilde{\boldsymbol{x}}(k|k-1) - \ell\boldsymbol{n}(k). \tag{17}$$

Then

$$P(k|k) = (1-\ell)^2 P(k|k-1) + \ell^2 r\mathcal{I}_N. \tag{18}$$

Further note that (12) can be solved explicitly to obtain

$$\boldsymbol{z}(k+1|k) = \sum_{i=0}^m Q^i\hat{\boldsymbol{x}}(k|k) + \sigma_c\sum_{i=1}^m Q^{m-i}\boldsymbol{u}\Big(k+\frac{i}{m}\Big). \tag{19}$$

Substituting $\boldsymbol{z}(k+1|k)$ in (13) and using (14), we obtain

$$\begin{aligned}\tilde{\boldsymbol{x}}(k+1|k) &= Q^\dagger\tilde{\boldsymbol{x}}(k|k) + w(k+1)\mathbf{1}_N\\ &-\left(\frac{1-\zeta}{m+1}\right)\sum_{i=1}^m Q^{m-i}\sigma_c\boldsymbol{u}\Big(k+\frac{i}{m}\Big),\end{aligned} \tag{20}$$

where $Q^\dagger = \zeta\mathcal{I}_N + \left(\frac{1-\zeta}{m+1}\right)\sum_{i=0}^m Q^i$. It follows that

$$\begin{aligned}P(k+1|k) =& Q^\dagger P(k|k)(Q^\dagger)^\top + q\mathbf{1}_N\mathbf{1}_N^\top\\ &+\left(\frac{1-\zeta}{m+1}\right)^2\sigma_c^2\sum_{i=0}^{m-1}Q^i(Q^i)^\top.\end{aligned} \tag{21}$$

Since Q is row-stochastic with $Q\mathbf{1}_N = \mathbf{1}_N$, it can be shown that $Q^\dagger\mathbf{1}_N = \mathbf{1}_N$, i.e., $Q^\dagger$ is also row-stochastic. Using row-stochasticity of $Q^\dagger$, (18) and (21) we obtain

$$\begin{aligned}P(k+1|k) &= (1-\ell)^2 Q^\dagger P(k|k-1)(Q^\dagger)^\top + \ell^2 rQ^\dagger(Q^\dagger)^\top\\ &+q\mathbf{1}_N\mathbf{1}_N^\top + \left(\frac{1-\zeta}{m+1}\right)^2\sigma_c^2\sum_{i=0}^{m-1}Q^i(Q^i)^\top.\end{aligned} \tag{22}$$

We can solve (22) with initial condition $P(0|-1)$ to obtain:

$$\begin{aligned}P(k|k-1) &= (1-\ell)^{2k}Q^{\dagger k}P(0|-1)(Q^{\dagger k})^\top\\ &+\ell^2 r\sum_{i=0}^{k-1}(1-\ell)^{2i}Q^{\dagger(i+1)}(Q^{\dagger(i+1)})^\top + q\sum_{i=0}^{k-1}(1-\ell)^{2i}\mathbf{1}_N\mathbf{1}_N^\top\\ &+\left(\frac{1-\zeta}{m+1}\right)^2\sigma_c^2\sum_{i=0}^{k-1}(1-\ell)^{2i}\sum_{j=0}^{m-1}Q^{\dagger i}Q^j(Q^j)^\top(Q^{\dagger i})^\top.\end{aligned} \tag{23}$$

Taking the limit $k\to+\infty$ and using the geometric series summation formula, we establish (i).

The second statement follows using model decomposition of Q and the result that

$$\operatorname{tr}\left(Q^{\dagger im}(Q^{\dagger im})^\top\right) = \sum_{h=0}^{N-1}\left|\left(\frac{1-\zeta}{m+1}\right)\bar{\lambda}_h+\zeta\right|^{2im}. \tag{24}$$

■

Note that the steady-state error covariance (16) is bounded and hence our distributed estimation algorithm is stabilizing in the mean squared sense. Our algorithm is not necessarily optimal since it assumes convex weights ℓ and ζ to be constant. However, given our algorithm, we can choose optimal parameters ℓ and ζ as we discuss in the next section.

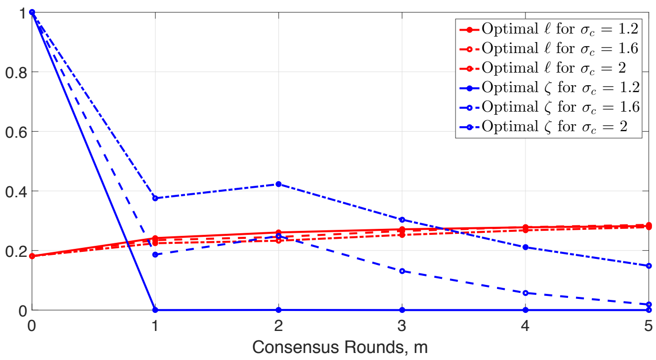

Fig. 2. Optimal ℓ and ζ as a function of consensus rounds m and σ_c with $N = 3$, $r = 25$, and $q = 1$ for an undirected line graph.

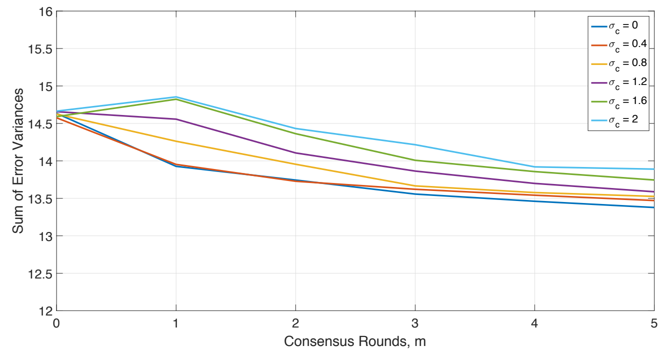

Fig. 3. Influence of communication noise in consensus dynamics on error variance across 200,000 Monte Carlo runs for the distributed filtering algorithm defined in §IV with $N = 3$, $r = 25$, and $q = 1$ for an undirected line graph. Even as σ_c increases the error variance no longer diverges as more consensus rounds are performed.

B. Tuning parameters ℓ and ζ

The algorithm defined in §IV requires two parameters ℓ and ζ to be tuned. For a given graph structure (fixed Q and N), given process, measurement and communication variance (fixed r, q, and σ_c), and a given number of consensus rounds (fixed m), we choose these parameters to minimize the asymptotic error covariance (16). In the following, let $J(\ell, \zeta) = \text{tr}\Big(\lim_{k\to\infty} P(k|k-1)\Big)$.

When determining the optimal (ℓ, ζ) which minimize J, we note a special case for $m = 0$. In this case of no consensus the modified algorithm will only involve (11) and (13). Inspecting (13) we see that the only appropriate formulation would have $\zeta = 1$. With $\zeta = 1$ and $m = 0$ we see that (16) simplifies to $J|_{m=0} = \frac{(\ell^2 r+q)N}{1-(1-\ell)^2}$. We then minimize $J|_{m=0}$ using `fmincon` in MATLAB to solve for the optimal ℓ. Likewise, for $m > 0$ we can use `fmincon` to solve for the optimal ℓ and ζ which minimize J.

The trends of optimal ℓ and ζ as a function of number of consensus rounds m and σ_c are shown in Fig. 2. Note that optimal ℓ increases with m, while ζ does not follow a monotonic trend. The initial trend of ζ is attributed to the transient consensus dynamics. As the number of consensus rounds increases, the value ζ goes to zero as discussed in §IV.

C. Numerical Simulations

We numerically investigate the performance of the modified estimation algorithm developed in §IV. We consider an undirected line graph with $N = 3$ nodes. We choose the same parameters as in Fig. 1 and again choose as our performance metric $\sum_{i=1}^{3} \text{var}(\tilde{x}_i(4|3))$. For each value of m and σ_c we use the optimal ℓ and ζ in the algorithm as determined in §V-B. Fig. 3 shows for the modified estimation algorithm the summed error variance metric versus the number of consensus rounds m, with the color of the lines designating the value of σ_c in (12). Comparing with Fig. 1, we see that the error variance no longer increases in an unbounded way as more consensus rounds are performed. Rather, the trend in error variance versus consensus rounds is much closer to the monotonically decreasing trend one expects in a distributed filter without communication noise. Even for larger values of σ_c we see the error variance decreases with additional consensus rounds, which is how an effective estimation algorithm should perform. Note the slight difference in scale between the vertical axes in Figs. 1 and 3.

VI. Conclusions

In this paper we studied consensus-based distributed linear filtering under noisy communication. We investigated how noisy communication affects the performance of the distributed filtering algorithm proposed in [6]. We showed that under noisy communication the error covariance of the estimator obtained using this algorithm diverges. We modified the algorithm from [6] to develop a novel distributed filtering algorithm that achieves a bounded asymptotic error covariance under noisy communication. We discussed how the parameters of this new algorithm can be tuned.

Future directions include examining the convexity of the asymptotic error covariance with respect to algorithm parameters, extending to vector-valued dynamical processes, and exploring the influence of the network graph on the performance.

References

[1] I. Matei and J. S. Baras. Consensus-based linear distributed filtering. *Automatica*, 48:1776–1782, 2012.

[2] S. Das and J. M. F. Moura. Consensus+Innovations Distributed Kalman Filter with Optimized Gains. *IEEE Transactions on Signal Processing*, arXiv:1605.06096v2, October 2016.

[3] R. Olfati-Saber. Distributed Kalman filter with embedded consensus filters. In *IEEE Conference on Decision and Control*, pages 8179–8184. IEEE, 2005.

[4] R. Olfati-Saber. Distributed Kalman filtering for sensor networks. In *IEEE Conference on Decision and Control*, pages 5492–5498. IEEE, 2007.

[5] U. A. Khan and J. M. F. Moura. Distributing the Kalman filter for large-scale systems. *IEEE Transactions on Signal Processing*, 56(10):4919–4935, 2008.

[6] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri. Distributed Kalman filtering based on consensus strategies. *IEEE Journal on Selected Areas in Communications*, 26(4):622–633, 2008.

[7] L. Xiao, S. Boyd, and S-J. Kim. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 67(1):33–46, 2007.

[8] F. Zhang and N. E. Leonard. Cooperative filters and control for cooperative exploration. *IEEE Transactions on Automatic Control*, 55(3):650–663, 2010.

[9] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[10] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[11] R. Olfati-Saber and J. S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. In *IEEE Conference on Decision and Control*, pages 6698–6703. IEEE, 2005.

[12] D. P. Spanos and R. M. Murray. Distributed sensor fusion using dynamic consensus. In *IFAC World Congress*. Citeseer, 2005.

[13] A. Speranzon, C. Fischione, K. H. Johansson, and A. Sangiovanni-Vincentelli. A distributed minimum variance estimator for sensor networks. *IEEE Journal on Selected Areas in Communications*, 26(4):609–621, 2008.

[14] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman. Decentralized environmental modeling by mobile sensor networks. *IEEE Transactions on Robotics*, 24(3):710–724, 2008.

[15] G. F. Young, L. Scardovi, and N. E. Leonard. Robustness of noisy consensus dynamics with directed communication. In *American Control Conference*, pages 6312–6317, 2010.

[16] I. Poulakakis, G. F. Young, L. Scardovi, and N. E. Leonard. Information centrality and ordering of nodes for accuracy in noisy decision-making networks. *IEEE Transactions on Automatic Control*, 61(4):1040-1045, 2016.

[17] V. Srivastava and N. E. Leonard. Collective decision-making in ideal networks: The speed-accuracy trade-off. *IEEE Transactions on Control of Network Systems*, 1(1):121–132, 2014.

[18] M. H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[19] N. E. Friedkin. The problem of social control and coordination of complex systems in sociology: A look at the community cleavage problem. *IEEE Control Systems*, 35(3):40–51, 2015.

[20] K. J. Åström. *Introduction to stochastic control theory*, volume 70 of *Mathematics in Science and Engineering*. Academic Press, New York, 1970.

[21] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press, 2009. Electronically available at http://coordinationbook.info.

[22] J. N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, Massachusetts Institute of Technology, November 1984.