Anthony Savas
December 16, 2016
Pre-Generals Committee Meeting

On distributed Kalman filtering with noisy communication.

My research alongside Professor Naomi Leonard and Professor Vaibhav Srivastava (Michigan State U.) considers Kalman filtering of a scalar linear stochastic process under noisy communication. Specifically we've been investigating how communication noise degrades the performance of an existing distributed estimation algorithm from the literature and have developed a novel algorithm which mitigates these problems.

We consider the following scalar linear stochastic process

$$\dot{x}(t) = v(t), \quad x(0) = X_0. \tag{1}$$

where $v(t)$ is a white noise process with variance $q \in \mathbb{R}_{>0}$, and X_0 is a Gaussian random variable with mean x_0 and variance σ. We then consider the problem of distributed state estimation of the state $x(t)$ using multiple communicating agents. These agents can communicate over a fixed interaction graph which is undirected and connected in the sense that there exists a path from each node to every other node. We assume that each agent $i \in \{1, \dots, N\}$ samples the process (1) at times k, $k \in \mathbb{Z}_{\geq 0}$, and collects a noisy measurement $y_i(k)$ of the process $x(k)$ defined by

$$y_i(k) = x(k) + n_i(k), \quad \text{for each } i \in \{1, \dots, N\}, \tag{2}$$

where $\{n_i(k)\}_{k\in\mathbb{Z}_{\geq 0}}$ are uncorrelated zero mean Gaussian noises with variance r. We further assume the noise sequences $n_i(k)$ are independent for different $i \in \{1, \dots, N\}$.

We consider the two-stage strategy proposed in [1] for estimation problems of the type given. During the first stage, at time k each agent i computes the estimate of process $x(t)$ 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. The first stage is given by

$$\hat{\boldsymbol{x}}(k|k) = (1-\ell)\hat{\boldsymbol{x}}(k|k-1) + \ell \boldsymbol{y}(k), \tag{3}$$

where $\ell \in (0,1)$ is the Kalman gain. The second stage comprises m steps of consensus using local estimates $\hat{x}_i(k|k)$. The consensus steps ensure that the local estimates of each agent converge towards the average of the group. However, we consider the case when the consensus step itself has associated noise which we represent as

$$\hat{\boldsymbol{x}}\Big(k + \frac{h}{m}\Big|k\Big) = Q\hat{\boldsymbol{x}}\Big(k + \frac{(h-1)}{m}\Big|k\Big) + \sigma_c \boldsymbol{u}\Big(k + \frac{h}{m}\Big), \tag{4}$$

where $h \in \{1, \dots, m\}$, Q is the doubly stochastic consensus matrix, $\boldsymbol{u}(k+h/m)$ is the N-variate Gaussian noise with zero mean and covariance $\mathcal{I}_N$ and σ_c^2 is the communication noise variance. The estimation error at time k is defined by

$$\tilde{\boldsymbol{x}}(k|k-1) = x(k)\mathbf{1}_N - \hat{\boldsymbol{x}}(k|k-1). \tag{5}$$

We numerically investigated the performance of this two-stage algorithm under under noisy communication for a set of three agents $\{1, 2, 3\}$ communicating over an undirected line graph. We took the convexity parameter to be $\ell = 0.5$ and performed $20,000$ Monte-Carlo simulations to estimate the trace of the error covariance matrix. Fig. 1 shows the trace of the error covariance matrix for $k = 4$ as a function of the number of consensus steps m. It can be seen that for large enough values of σ_c the trace of the error covariance actually increases as more consensus steps are performed, suggesting that the two-stage estimation algorithm is not stabilizing, i.e., the trace of the error covariance diverges as the number of consensus steps are increased.

We modified the previous algorithm to mitigate the effects of noisy communication. We keep the update in the first stage of the algorithm the same as in (3). We modify the second stage 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 z within the sampling epochs as follows:

$$\boldsymbol{z}\Big(k+\frac{h}{m}\Big|k\Big) = Q\boldsymbol{z}\Big(k+\frac{(h-1)}{m}\Big|k\Big)+\sigma_c \boldsymbol{u}\Big(k+\frac{h}{m}\Big)+\hat{\boldsymbol{x}}(k|k). \tag{6}$$

In (6), each agent remembers their estimate $\hat{\boldsymbol{x}}(k|k)$ at the end of the last sampling and re-injects it at each consensus round. Loosely speaking, the intuition for

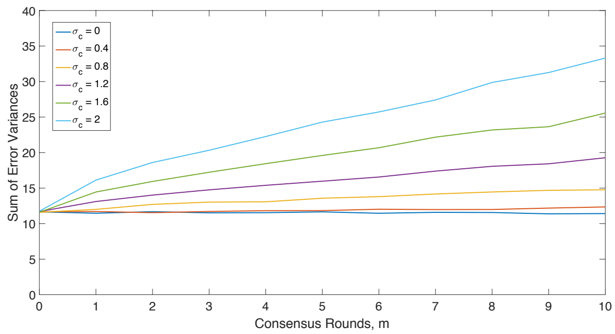

Figure 1: Influence of consensus noise on error variance across 20,000 Monte Carlo runs for algorithm (4) with $N = 3$, $r = 1$, and $q = 1$ for a simple undirected line graph. We see that the error variance is divergent as the number of consensus steps increases.

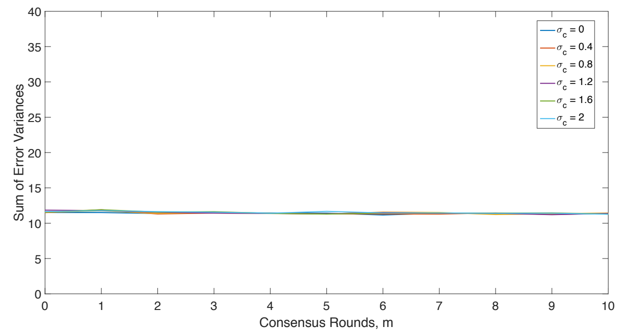

Figure 2: Influence of consensus noise on error variance across 20,000 Monte Carlo runs for the updated algorithm with $N = 3$, $r = 1$, and $q = 1$ for a simple undirected line graph. Now even as σ_c increases the error variance no longer diverges as more consensus rounds are performed.

such an update is that starting from a deterministic initial condition $\boldsymbol{z}(k|k) = \hat{\boldsymbol{x}}(k|k)$, 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 $j \in \{1, \ldots, N\}$. 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 then consensus noise will still be an issue. To mitigate its effect for small m, 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

$$\hat{\boldsymbol{x}}(k+1|k) = \zeta\hat{\boldsymbol{x}}(k|k) + (1-\zeta)\frac{\boldsymbol{z}(k+1|k)}{m+1}, \qquad (7)$$

where $\zeta \in (0, 1]$ is a constant. This algorithm has two tunable parameters ℓ and ζ, and these parameters can be chosen to minimize the asymptotic error covariance of the estimator.

We now numerically investigate the performance of the modified estimation algorithm. We again consider an undirected line graph with 3 nodes. For each value of m and σ_c we use the optimal ℓ and ζ as determined by minimizing the asymptotic error covariance. Fig. 2 shows for the modified estimation algorithm the summed error variance metric vs. the number of consensus steps m, with the color of the lines designating the value of σ_c in (6). Comparing with Fig. 1, we see that the error variance no longer increases as more consensus steps are performed. Rather, the error variance vs. consensus steps trend is much closer to the monotonically decreasing ideal. Even for larger values of σ_c we do not see the error variance increase with additional consensus, which is how an effective estimation algorithm should perform.

There are several areas for continued research. We would like to investigate centrality measures that predict the ordering of the performance of individual agents. Another interesting area is to study estimation problems in leader-follower networks in which only a subset of agents (leaders) can sample the stochastic process and investigate associated leader selection problems.

References:

[1] 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