Distributed Estimation

Yilin Mo

November 11, 2014

We model a network composed of m agents as a graph $G = \{V, E\}$. $V = \{1, 2, \dots, m\}$ is the set of vertices representing the agents. $E \subseteq V \times V$ is the set of edges. $(i, j) \in E$ if and only if sensor i and j can communicate directly with each other. We will always assume that G is undirected, i.e. $(i, j) \in E$ if and only if $(j, i) \in E$. We further assume that there is no self loop, i.e., $(i, i) \notin E$.

1 Static Case

Let $x \in \mathbb{R}^n$ be the state. We assume that $x \sim \mathcal{N}(0, \Sigma)$. Each sensor make a measurement of x:

$$y_i = h_i x + v_i.$$

We will assume that $v_i \sim \mathcal{N}(0, R_i)$. $x, v_1, \dots, v_m$ are jointly independent from each other.

The optimal state estimate of x given y is

$$\hat{x} = \mathbb{E}(x|y) = \Sigma H^T (H \Sigma H^T + R)^{-1} y,$$

with error covariance

$$P = \Sigma - \Sigma H^T (H\Sigma H^T + R)^{-1} H \Sigma = \left(\Sigma^{-1} + H^T R^{-1} H\right)^{-1} = \left(\Sigma^{-1} + \sum_{i=1}^m h_i^T R_i^{-1} h_i\right)^{-1}$$

Furthermore, we have

$$\begin{aligned} P^{-1}\hat{x} &= (\Sigma^{-1} + H^T R^{-1} H)\Sigma H^T (H\Sigma H^T + R)^{-1} y \\ &= \left(H^T R^{-1} R (H\Sigma H^T + R)^{-1} + H^T R^{-1} H \Sigma H^T (H \Sigma H^T + R)^{-1}\right) y \\ &= H^T R^{-1} y = \sum_{i=1}^m h_i^T R_i^{-1} y_i. \end{aligned}$$

Let

$$S = \frac{1}{m}\sum_{i=1}^m h_i^T R_i^{-1} h_i,$$

and

$$z = \frac{1}{m} h_i^T R^{-1} y_i.$$

S and z can be computed via consensus algorithm.

Remark 1. *S is of dimension $n \times n$ and z is of dimension n.*

Thus, each sensor can compute the state estimate and the corresponding error covariance matrix:

$$P = \frac{1}{m} \left[(m\Sigma)^{-1} + S \right]^{-1},$$

and

$$\hat{x} = \left[(m\Sigma)^{-1} + S \right]^{-1} z.$$

2 Dynamic Case: Distributed Kalman Filter

Kalman Filter:

1. **Initialization:**
$$\hat{x}(0|-1) = 0,\ P(0|-1) = \Sigma. \tag{1}$$

2. **Prediction:**
$$\hat{x}(k+1|k) = A\hat{x}(k),\ P(k+1|k) = AP(k)A^T + Q. \tag{2}$$

3. **Correction:**
$$\hat{x}(k+1) = \hat{x}(k+1|k) + P(k+1|k)C^T(CP(k+1|k)C^T + R)^{-1}(y(k+1) - C\hat{x}(k+1|k)), \tag{3}$$
$$P(k+1) = P(k+1|k) - P(k+1|k)C^T(CP(k+1|k)C^T + R)^{-1}CP(k+1|k). \tag{4}$$

2.1 Type I Filter: Fusion of Sensory Data

Similar to the static case, we have

$$\begin{aligned} P(k+1) &= \left(P(k+1|k)^{-1} + C^T R^{-1} C\right)^{-1} \\ &= \left(P(k+1|k)^{-1} + \sum_{i=1}^{m} c_i^T R_i^{-1} c_i\right)^{-1}, \end{aligned}$$

and

$$\begin{aligned} P(k+1)^{-1}\hat{x}(k+1) - P(k+1)^{-1}\hat{x}(k+1|k) &= C^T R^{-1}(y(k+1) - C\hat{x}(k+1|k)), \\ &= \sum_{i=1}^{m} c_i^T R_i^{-1} \left(y_i(k+1) - c_i\hat{x}(k+1|k)\right) \\ &= \sum_{i=1}^{m} c_i^T R_i^{-1} y_i(k+1) - \left(\sum_{i=1}^{m} c_i^T R_i^{-1} c_i\right)\hat{x}(k+1|k). \end{aligned}$$

Let us define

$$S = \frac{1}{m} c_i^T R_i^{-1} c_i,$$

and

$$z(k+1) = \frac{1}{m} \sum_{i=1}^{m} c_i^T R_i^{-1} y_i(k).$$

Then

$$P(k+1) = \frac{1}{m} \left[(mP(k+1|k))^{-1} + S \right]^{-1},$$

and

$$\hat{x}(k+1) = \hat{x}(k+1|k) + \left[(mP(k+1|k))^{-1} + S \right]^{-1} (z(k+1) - S\hat{x}(k+1|k))$$

Remark 2. *Infinite amount of communication is needed in order to reach consensus on S and $z(k+1)$.*

2.2 Type II Filter: Consensus on the State Estimate

For each sensor i:

1. **Initialization:**

$$\hat{x}_i(0|-1) = 0, \ P_i(0|-1) = \Sigma. \tag{5}$$

2. **Prediction:**

$$\hat{x}_i(k+1|k) = A\hat{x}_i(k), \ P_i(k+1|k) = AP_i(k)A^T + Q. \tag{6}$$

3. **Correction:**

$$\begin{aligned} \hat{x}_i(k+1) = {} & \hat{x}_i(k+1|k) + P_i(k+1|k)C^T(CP_i(k+1|k)C^T + R)^{-1}(y_i(k+1) - C\hat{x}_i(k+1|k)) \\ & + \varepsilon P_i(k+1)^{-1} \sum_{j \in \mathcal{N}_i} [\hat{x}_j(k+1|k) - \hat{x}_i(k+1|k)], \end{aligned} \tag{7}$$

$$P_i(k+1) = P_i(k+1|k) - P_i(k+1|k)C^T(CP_i(k+1|k)C^T + R)^{-1}CP_o(k+1|k). \tag{8}$$

Remark 3. *Each sensor runs its own Kalman filter and they try to fuse the state estimation by running consensus.*

2.3 Type III Filter: Constant Gain Strategy

We know that KF has the same asymptotic performance as a constant gain filter. Hence, we may be able to forget the covariance update.

Consider a simple case, where $x \in \mathbb{R}$ is a scalar. $A = 1$, $Q = q$, $c_i = 1$ and $R_i = r$.

The KF will eventually have the same asymptotic performance as the following estimator:

$$\hat{x}(k+1) = (1-\alpha)\hat{x}(k) + \alpha\frac{\mathbf{1}^T y(k+1)}{m},$$

where $\alpha \in \mathbb{R}$ is the optimal gain.

Now we consider a distributed strategy:

1. Each sensor computes the local $\hat{x}_i(k)$ based on its local measurement:

$$\hat{x}_i(k) = (1-\alpha)\hat{x}_i(k|k-1) + \alpha y_i(k).$$

Denote $\hat{x}(k)$ to be

$$\hat{x}(k) \triangleq \begin{bmatrix} \hat{x}_1(k) \\ \vdots \\ \hat{x}_m(k) \end{bmatrix}$$

2. We then run consensus for l times. Hence,

$$\hat{x}^+(k) = M^l \hat{x}(k),$$

where M is the consensus matrix.

3. For the prediction, since $A = 1$, we have

$$\hat{x}(k+1|k) = \hat{x}^+(k).$$

Define $P(k) \triangleq \mathrm{Cov}(\hat{x}(k) - x(k)\mathbf{1})$, $P^+(k) \triangleq \mathrm{Cov}(\hat{x}^+(k) - x(k)\mathbf{1})$.
Thus,

$$P^+(k) = M^l P(k) M^{lT}.$$

and

$$\begin{aligned} P(k+1) &= (1-\alpha)^2 P^+(k) + (1-\alpha)^2 q\mathbf{1}\mathbf{1}^T + \alpha^2 r I \\ &= (1-\alpha)^2 M^l P(k) M^{lT} + (1-\alpha)^2 q\mathbf{1}\mathbf{1}^T + \alpha^2 r I. \end{aligned}$$

We only care about the asymptotic performance. Consider the case where $k \to \infty$, we have

$$\lim_{k\to\infty} P(k) = q\mathbf{1}\mathbf{1}^T \sum_{k=1}^{\infty} (1-\alpha)^{2k} + \alpha^2 r \sum_{k=0}^{\infty} (1-\alpha)^{2k} M^{kl} M^{klT}.$$

Therefore, define the cost to be

$$J = \lim_{k\to\infty} \mathrm{tr}\left(P(k)\right) = \frac{mq(1-\alpha)^2}{1-(1-\alpha)^2} + \sum_{i=1}^{m} \frac{r\alpha^2}{1-(1-\alpha)^2|\lambda_i|^{2l}},$$

where λ_i is the ith eigenvalue of M.

Remark 4. *In general, it is difficult to jointly design M and α. If we fix M, then α can be found by numerical methods. One interesting observation is that if we increase the number of consensus steps l between each time interval, then we will increase the gain and decrease the cost J.*