Comparison of consensus filter

ref: Comparison of Optimal Distributed Estimation and Consensus Filtering (2016 information fusion)

The sensors observe the state of a dynamic system and send the measurements to the processors and compute state estimates from the available data. Distributed state estimation involves two steps: communication between nodes and fusion of local and non-local data at each node.

Covariance-based fusion

Covariance-based state estimate fusion approaches model the dependent errors between the local estimates by cross-covariances and use them to compute the optimal combination of the local estimates. Algorithms inclue maximum likelihood, minimum variance, best linea unbiased estimate, and covariance intersection.

CI is an algorithm for combine two or more estimates of state variables in a Kalman filter when the correlation between them is unknown.
(https://en.wikipedia.org/wiki/Covariance_intersection)

Items of information a and b are known and are to be fused into information item c, We known a and b have mean/covariance, but the correlation is not known. The CI update gives mean and covariance as follows:

$$P^{-1} = wP_1^{-1} + (1-w)P_2^{-1}$$

$$M = P * (wP_1^{-1}M_1 + (1-w)P_2^{-1}M_2)$$

where w is computed to minimize a selected norm, e.g.., logdet or trace.

Consider here the fusion of two estimates using either the Kalman filter or the data fusion technique known as Covariance Intersection (CI), under the following circumstances

1. The two estimates are independent In this case the Kalman filter produces an optimal fused estimate while CI produces a consistent, though suboptimal, estimate
2. The two estimates are completely correlated In this case CI produces an optimal fused estimate while Kalman produces an inconsistent estimate.
3. The two estimates are partially correlated In this case CI produces a consistent, though suboptimal,estimate while the Kalman filter produces an inconsistent estimate

Consensus filter

Consensus filter does not attempt to compute the optimal estimate given all the sensor measurements or a constrained optimal estimate given the estimates to be fused. It uses average consensus to compute the estimates at each node. More specifically, each node computes an estimate and broadcasts to its neighbors. Each neighbor then upadte its estimates with a consensus algorithm. Algorithms include measurement weighted consensus,

information weighted consensus, and hybrid consensus.

> R. Olfati-Saber, “Distributed Kalman filtering for sensor networks,” Proc. 46th IEEE Conf. Decision Control, 2007.
> M. Kamgarpour and C. Tomlin, “Convergence properties of a decentralized Kalman filter,” Proc. 47th IEEE Conf. Decision Control, 2008.
> G Battistelli and Chisci, “Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability,’ Automatica, vol. 50, no. 3, pp. 707–718, Mar. 2014.
> G Battistelli, L Chisci, G Mugnai, A. Farina, and A. Graziano, “Consensus-based algorithms for distributed filtering,” in Proc. 51st IEEE Conf. Decision Control, 2012, pp. 794–799.
> G Battistelli, L Chisci, G Mugnai, A. Farina, and A. Graziano, “Consensus-based linear and nonliner filtering,” IEEE Trans. Autom. Control, vol. 60, no. 5, May 2015, pp. 1410–1415

The consensus on measurements(CM) filter uses a consensus approach to estimate the information in the local cumulative measurements of node i. ,$b_{k,l}^{i}$ and the information matrix $B_{k,l}^{l}$, After consensus is reached, the state estimate and error covariance P are update using these information.

> G Battistelli, L Chisci, G Mugnai, A. Farina, and A. Graziano, “Consensus-based linear and nonliner filtering,” IEEE Trans. Autom. Control, vol. 60, no. 5, May 2015, pp. 1410–1415.

Define the measurement information $i = H^T R^{-1} z$ and measurement information matrix $I = H^T R^{-1} H$.

Kalman Consensus Filter and its Extension

Problem Formulation

Consider a sensor network with N_c sensors. There are no specific assumptions on the overlap between the FOVs of the sensors. The comunication in the network can be represented using an undirected connect graph $\mathcal{G} = (\mathcal{C}, \mathcal{E})$.

Each node also maintains a prior state estimate $x_i^-(t)$ and covariance $P_i^-(t)$ for each target.

kalman Consensus Filter(KCF)

Algorithm 1 KCF at C_i at time step t

---

1) Given measurement z_i

2) Compute information vector and matrix

$$u_i = H^T R^{-1} z_i$$
$$U_i = H^T R^{-1} H_i$$

3) Broadcast u_i, U_i to neighbors and receive u_j, U_j from neighbors

4) Fuse the information vectors and matrics

$$b_i = \sum_{i \in \mathcal{N}_i} u_i$$
$$B_i = \sum_{i \in \mathcal{N}_i} U_i$$

5) Compute Kalman Consensus estimate

$$P_i = (P_i^- + B_i)^{-1}$$
$$x_i = x_i^- + P_i(b_i - B_i x_i^-) + \gamma(P_i^-)^{-1} \sum_{j \in \mathcal{N}_i} x_j^- - x_i^-$$

6) Predict for next time step

The state estimates of all the nodes get the same weight in the summation. Since naive nodes do not have observations of the target, their estimates are often highly erroneous.

The information matrix measurement update consider the node's own information matrix and the local neighborhood's measurement covariance. It does not account for cross covariance between the estimates by the node and its neighbors.

The naive nodes may lag behind by a significant time. This happens because naive nodes do not have direct access to new observation of a target, the only way they can get unpdated information about a target is through a node might be multiple iterations away from getting new information about a target. This information imbalance can cause large oscillations.

Generalized Kalman Consensus Filter(GKCF)

Algorithm2 GKCF at C_i at time step t

---

1) Given measurement z_i

2) Compute information vector and matrix

$$u_i = H^T R^{-1} z_i$$
$$U_i = H^T R^{-1} H_i$$

3) Broadcast u_i, U_i to neighbors and receive u_j, U_j from neighbors

4) Fuse the information vectors and matrics

$$b_i = \sum_{i \in \mathcal{N}_i} u_i$$
$$B_i = \sum_{i \in \mathcal{N}_i} U_i$$

5) Initialize consensus variables

$$v_i[0] = P_i^- x_i^-$$
$$V_i[0] = P_i^-$$

6) For K iterations run average consensus on $v_i[k] and V_i[k]$ and then compute updated estimate

$$x_i^- = V_i[K]^{-1} v_i[K]$$
$$P_i^- = V_i[K]$$

7) Compute GKCF estimate

$$P_i = P_i^- + B_i$$
$$x_i = x_i^- + P_i(b_i - B_i x_i^-)$$

6) Predict for next time step
