Scalable Distributed Kalman Filtering Through Consensus

Shrut Kirti and Anna Scaglione
Cornell University

IEEE Upstate NY Workshop on Communications, Sensors, and Networking

Nov. 9, 2007

Problem Overview

- A fundamental problem in sensor networks - *distributed detection and estimation*
- Key requirement for distributed algorithms - *scalability*
- **Our objective -**
  - Distributed Kalman filtering in a wireless sensor network setting
  - Emphasis on designing a communication architecture which provides scalability
- Many interesting applications of this. One very simple motivating example-
  - Distributed tracking of the position of an object moving on a plane

Outline of the Talk

Part *I* - Kalman Filter

- Formulation of the distributed Kalman filter (DKF) and its connection to average consensus
- *(Lack of)* scalability of traditional average consensus in wireless networks

Part *II* - Communication Architecture

- Data driven average consensus - a scalable architecture
- Simulation results

Part I: Distributed Kalman Filter

System Model

- Consider linear dynamical system with state-space model-

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{w}_k \quad (1)$$

$$\mathbf{z}_k^i = H_k^i \mathbf{x}_k + \mathbf{v}_k^i \quad (2)$$

  where $\mathbf{w}_k \sim \mathcal{N}(0, Q_k)$, $\mathbf{v}_k \sim \mathcal{N}(0, R_k)$, $\mathbf{x}_k \in \mathbb{R}^m$, and $\mathbf{z}_k^i \in \mathbb{R}^p$
  A_k, B_k, and H_k^i are known matrices
  v_k^i are independent
- Each node i makes noisy observation $\mathbf{z}_k^i$ at time step k
- **Goal** - calculate state estimate $\hat{\mathbf{x}}_k$

Distributed Kalman Filter

Olfati-Saber (2005) showed that the centralized Kalman filter recursions can be decomposed into -

$$
\begin{aligned}
M(k) &= ((nP_{k|k-1})^{-1} + S_k)^{-1} && (3) \\
\hat{\mathbf{x}}_k &= \bar{\mathbf{x}}_k + M_k(\mathbf{y}_k - S_k \bar{\mathbf{x}}_k) && (4) \\
P_{k+1} &= A_k M_k A_k^T + B_k (nQ_k) B_k^T && (5) \\
\bar{\mathbf{x}}_k &= A_k \hat{\mathbf{x}}_{k-1} && (6)
\end{aligned}
$$

where

$$
\begin{aligned}
\mathbf{y}_k &= \frac{1}{n}\sum_{i=1}^{n} y_k^i = \frac{1}{n}\sum_{i=1}^{n} (H_k^i)^T (R_k^i)^{-1} \mathbf{z}_k^i && (7) \\
S_k &= \frac{1}{n}\sum_{i=1}^{n} (H_k^i)^T (R_k^i)^{-1} (H_k^i) && (8)
\end{aligned}
$$

On the Scalability of Average Consensus

An iteration of average consensus - *each node must hear from all its neighbors*

Simple motivating example -

- Nodes (i, j) connected if $d_{ij} < r$
- Rayleigh fading channel + AWGN with noise power N_o

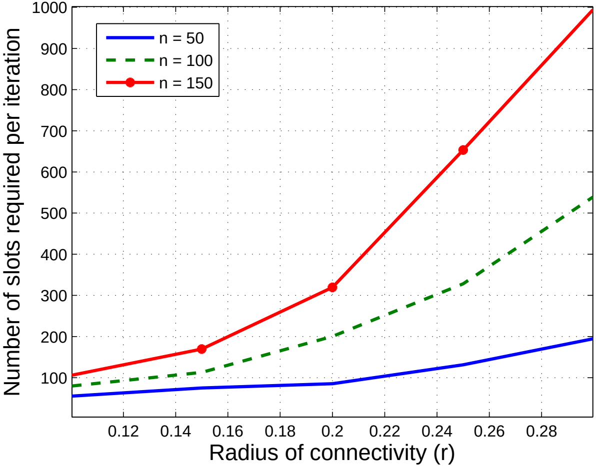
$$h_{ij} \overset{iid}{\sim} \mathcal{CN}(0, \mathcal{K}d_{ij}) \tag{9}$$

- TDMA with random access scheduling
- Node i transmits message with probability $p = 0.1$ in any slot
- Neighbor j receives message successfully if -

$$SINR_{ij} = \frac{P|h_{ij}|^2(1 + d_{ij})^{-\alpha}}{N_o + P\sum_{t:t\neq i}|h_{it}|^2(1 + d_{it})^{-\alpha}}.b_j[l] > \tau \tag{10}$$

Poor Scalability of Average Consensus in Wireless

Network congestion



Part II: Scalable Communication Architecture

Assumptions

Wireless channel model -

- Broadcast channel with Rayleigh fading and AWGN
- Half-duplex constraint

$$r_i[l] = \begin{cases} \sum_{j=1}^{n} h_{ij}[l] s_j[l] + w_i[l], & \text{if } i \text{ silent;} \\ 0, & \text{if } i \text{ transmits.} \end{cases} \quad (11)$$

- The channel is reciprocal $\mathbb{E}\{|h_{ij}|^2\} = \mathbb{E}\{|h_{ji}|^2\}$
- Fixed transmit power P_t

Also assume synchronized node operation

Data Driven Average Consensus

Consider usual average consensus -

$$\theta_i(t+1) = \theta_i(t) + \epsilon \underbrace{\sum_{j \in \mathcal{N}_i(t)} a_{ij}(t)(\theta_j(t) - \theta_i(t))}_{u_i(t)} \tag{12}$$

such that $\lim_{t \to \infty} \theta_i(t) = \frac{1}{n} \sum_{i=1}^{n} \theta_i(0)$

1^{st} **key point - need to quantize!**

$$\tilde{\theta}_i(t) = \arg\min_{q_l} |\theta_i(t) - q_l| \tag{13}$$

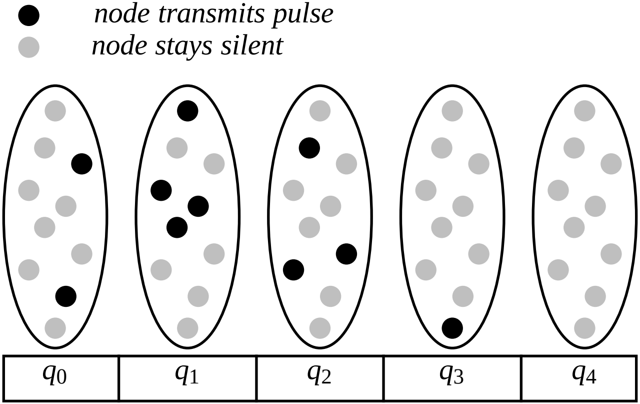
Then quantized consensus update becomes -

$$\tilde{u}_i(t) = \sum_{l=0}^{|\mathcal{Q}|-1} (q_l - \theta_i(t)) \sum_{j=1}^{n} a_{ij}(t) \delta[q_l - \tilde{\theta}_j(t)]$$

$$V_{ti}(q_l) \triangleq \sum_{j=1}^{n} a_{ij}(t) \delta[q_l - \tilde{\theta}_j(t)] \tag{14}$$

Data Driven Scheduling

Simple example of idea



$|\mathcal{Q}| = 5$ channel access slots

Data Driven Average Consensus

- Let transmitted signal be -

$$s_i[|\mathcal{Q}_t|t + l] = e^{j\phi_i}\delta[q_l - \tilde{\theta}_i(t)] \quad l = 0, \ldots, |\mathcal{Q}_t| - 1 \tag{15}$$

- 2^{nd} **key point - approximation of network information!**

$$\text{Network information:} \qquad V_{ti}(q_l) \triangleq \sum_{j=1}^{n} a_{ij}(t)\delta[q_l - \tilde{\theta}_j(t)] \tag{16}$$

  can be approximated as -

$$\hat{V}_{ti}(q_l) = \begin{cases} |r_i\,[l + |\mathcal{Q}_t|]|^2 - N_0, & \text{if } q_l \neq \tilde{\theta}_i(t) \\ 0, & \text{else.} \end{cases}$$

- Nodes use their local states to cooperatively form a vector code whose length is a function of the desired precision in consensus but is independent of the number of nodes
- The channel is assigned exclusively to the particular state value not to the node

Some Results in Brief

Some justification of approximation -

Lemma

$$\mathbb{E}\{\hat{\mathbf{u}}(t)\} = \tilde{\mathbf{u}}(t) \tag{17}$$

$$\hat{u}_i(k) = \sum_{l=0}^{|\mathcal{Q}|-1} (q_l - \theta_i(t))\hat{V}_{ki}(q_l), \ \tilde{u}_i(k) = \sum_{l=0}^{|\mathcal{Q}|-1} (q_l - \theta_i(t))V_{ki}(q_l) \tag{18}$$

Convergence of the algorithm -

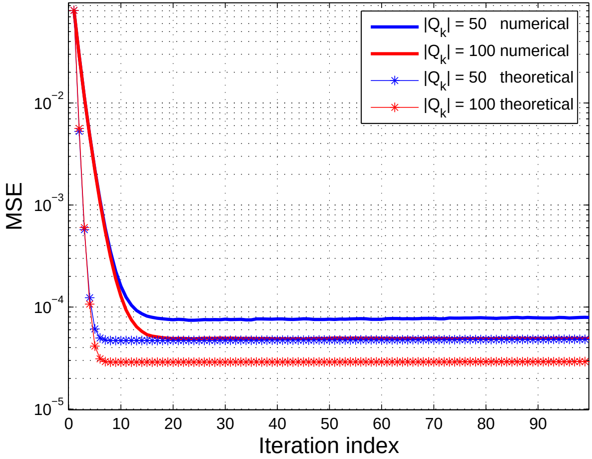
Lemma

The data driven algorithm is guaranteed to converge to the quantized true average under the condition that $|Q| < \infty$.

Idea - model the algorithm as a finite markov chain.

Accuracy of Data Driven Average Consensus

Mean square error performance at SNR = $30dB$ with 50 nodes



Recall DKF recursions

DKF recursions at each node -

$$\begin{aligned} M(k) &= ((nP_{k|k-1})^{-1} + S_k)^{-1} && (19) \\ \hat{\mathbf{x}}_k &= \bar{\mathbf{x}}_k + M_k(\mathbf{y}_k - S_k\bar{\mathbf{x}}_k) && (20) \\ P_{k+1} &= A_k M_k A_k^T + B_k(nQ_k)B_k^T && (21) \\ \bar{\mathbf{x}}_k &= A_k\hat{\mathbf{x}}_{k-1} && (22) \end{aligned}$$

So each node needs to track the dynamic average -

$$\mathbf{y}_k = \frac{1}{n}\sum_{i=1}^{n} y_k^i = \frac{1}{n}\sum_{i=1}^{n}(H_k^i)^T(R_k^i)^{-1}\mathbf{z}_k^i \qquad (23)$$

Putting Things Together

Data driven *dynamic* consensus

- Need to track the dynamic average -

$$\mathbf{y}_k = \frac{1}{n}\sum_{i=1}^{n} y_k^i = \frac{1}{n}\sum_{i=1}^{n} (H_k^i)^T (R_k^i)^{-1}\mathbf{z}_k^i \tag{24}$$

- Dynamic consensus update -
  Consider the observations at node i: $u_i(t) = r(t) + v_i(t)$

$$\theta_i(t+1) = \theta_i(t) + \epsilon \sum_{j\in\mathcal{N}_i(t)} a_{ij}(t)(\theta_j(t) - \theta_i(t)) + \epsilon \sum_{j\in\mathcal{N}_i(t)\cup\{i\}} a_{ij}(t)(u_j(t) - \theta_i(t)) \tag{25}$$

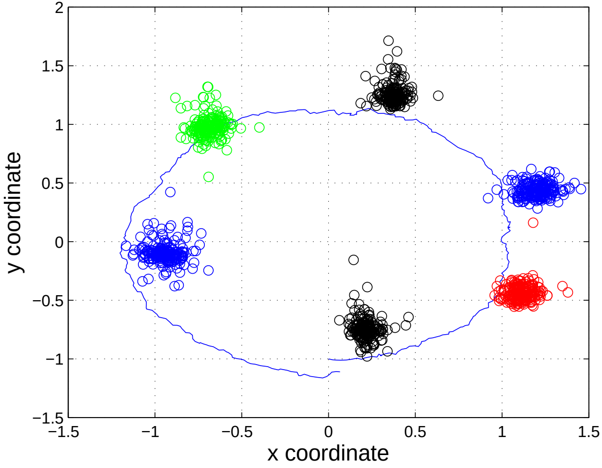
- Modify data driven algorithm for *component-wise* dynamic consensus -

$$s_i\left[l + (t-1)|\mathcal{Q}|\right] = e^{j\phi_i[l]}\left[\delta[q_l - \tilde{\theta}_i(t)] + \delta[q_l - [\tilde{\mathbf{y}}_k^j]_j]\right]$$

- Only 1 iteration of dynamic consensus is required per DKF iteration!

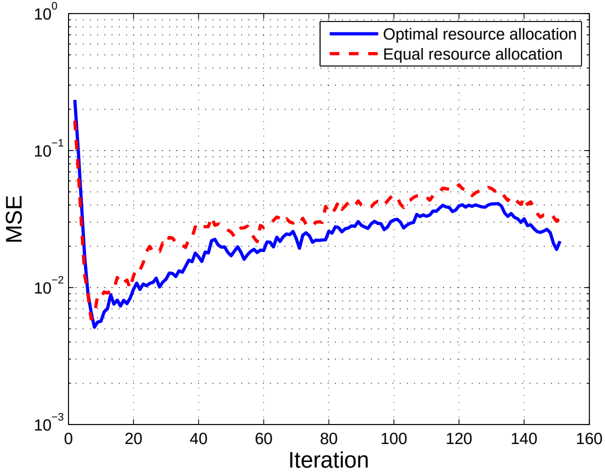
Simulation Results

Tracking the position of an object moving in circles



Simulation Results

Mean square error performance



Thank you!