Distributed Agent-based Dynamic State Estimation over a Lossy Network

S M Shafiul Alam, Balasubramaniam Natarajan, and Anil Pahwa

WiCom Group, Department of Electrical and Computer Engineering,
Kansas State University, Manhattan, KS-66506, USA
{alam,bala,pahwa}@ksu.edu
http://ece.k-state.edu/research/communications/wicom/index.html

Abstract. In this paper, a novel distributed agent-based dynamical system estimation strategy is proposed. Each agent has a local observation space and is interested in a specific set of system state elements. The agents have the ability of two-way communication with its neighbors (i.e., agents who share at least one state element). At a particular time instant, each agent predicts its state and makes intermediate correction based on its local measurements. Information about the corrected state elements are then exchanged among the neighboring agents. Based on the final processing of these exchanged information, an agent-based Kalman consensus Filter (AKCF) and uniform weighting based diffusion Kalman filter (ADKF) are proposed in the light of well-established theory of distributed Kalman filtering. Two different systems are simulated using the proposed filters. The effect of communication is also investigated by introducing random failures in the communication link among neighboring agents. It is observed that the mean square deviation (MSD) of AKCF is lower than that of ADKF for the scenarios considered. Additionally, the results also demonstrate that the AKCF is more robust to communication link failures than the ADKF.

Keywords: Kalman consensus filter, diffusion Kalman filter, random link failure.

1 Introduction

Kalman filtering has been an effective tool for real-time estimation and tracking of dynamical processes since its first formulation by R. E. Kalman [5]. Tracking of a massive physical system (e.g. electric power system) is possible now-a-days by deploying a distributed network of communicable sensors over the occupied geographical region [4], [7]. In conventional Kalman filter, the sensors communicate with a single fusion center either directly or hierarchically to send updated measurement information in timely manner. Based on the knowledge of previous state values and overall system dynamics, the fusion center makes a minimum mean squared error (MMSE) prediction of the states. Necessary corrections are made to the predicted states based on sensor measurements. However, the underlying communication and computational burden is considerably high with a centralized Kalman filter. This issue is resolved through distributed implementation of Kalman filters across the sensor network. In this scenario, the sensors have additional responsibility of implementing local Kalman filter and inter-sensor communication in the neighborhood. The objective of each sensor is to have updated status of the overall system through local prediction and necessary correction based on the type of information (measurement and / or predicted state values) exchanged among the neighbors. Although, the fundamental concept is unchanged, the distributed Kalman filter has evolved through numerous algorithms. Among those, Kalman consensus filter (KCF) [10], [11] and diffusion Kalman filter (DKF) [1] are worth mentioning. We refer to [8] to have a glimpse of the research carried out in this regard. It should be noted that, beside the system dynamics and local observation space, the topology of active sensor network as well as the inter-sensor communication reliability play vital role in successful implementation of the distributed Kalman filters. For KCF, the effect of lossy sensor network is investigated in [12] by incorporating a Bernoulli random variable in the consensus step. In [2], the study is further extended through random addition of nonlinear dynamics as well as the quantization effect in the sensor communication. On the other hand, the effect of communication link failures and delays on the diffusion Kalman filtering still needs to be investigated. Authors in [13] propose relative variance and adaptive combination rule for a single stationary parameter estimation. These rules are used to diffuse the neighborhood information received over noisy communication links.

Dynamic state estimation in a large cyber-physical system (CPS) presents some unique challenges. A typical example is the smart electric power distribution system with thousands of end-users. For such a system, the dimension of corresponding state vector is quite large presenting high computational as well as communication burden for the sensors. As mentioned earlier, in a conventional distributed Kalman filtering setup, each sensor has to regularly store and update the global state vector and estimation error covariance matrix. Thus, it may be impractical to track the high dimensional state vector in its entirety at each communicable sensor. This constraint can be overcome specifically for sparse large-scale linear system [6]. In this case, the corresponding transition of states can be reflected on (approximately) banded matrix to spatially decompose the overall dynamics among sensors even when local measurement space projects onto global states. This idea is further extended for system specific reduced order particle filtering [9] and distributed observer design for large-scale system partitioned into disjoint areas [3]. On the contrary, a particular sensor may be interested only on the state elements, which are directly coupled to its local observation space. Some state elements may even be coupled to two or more sensors' observation space. Under these circumstances, each sensor may be relieved to track only the pertinent state elements, thus reducing the size of error covariance matrices and subsequent reduction in computational requirements. In this way, the sensors are acting as agents and the corresponding local Kalman filters are referred to as agent-based Kalman filters.

In this paper, an agent-based general formulation of KCF and DKF is proposed. One is called agent-based KCF (AKCF) and the other one is agent-based DKF (ADKF). Each agent has access to a distinct set of measurements, that are coupled to a subset of global state elements. A set of binary projection matrices are defined based on the distribution of system state elements over the observation space of the agents. These matrices map the overall system dynamics to agent-specific state-space model and also define the set of neighbors of a particular agent. AKCF and ADKF are developed by proper inclusion of these projection matrices in the basic filtering steps. The application of proposed filters are illustrated with two custom built 3-agent systems to make a comparative performance analysis. The effect of losses in the inter-agent information exchange is also investigated by allowing random and independent failure of the existing communication links.

1.1 Contributions

We summarize the contributions of our work as follows:

- Introduce binary projection matrices to form agent-wise local state-space model.
- Define Agent neighborhood based on the sharing of state elements.
- Develop AKCF and ADKF where the observation space of each agent is (generally) underdetermined.
- Investigate the effect of communication over the performance of agent-based tracking of the dynamical system.

The rest of the paper is organized as follows. In section 2, general system model is given and projection matrices are defined accordingly. Detailed description of the proposed AKCF and ADKF are given in section 3 and compared with typical distributed Kalman filter in terms of computation and communication requirements. The modeling of communication effect is discussed in section 4. In the next section the proposed filters are compared for perfect and lossy communications over two multi-agent systems.

2 System Model

We consider a system whose dynamics can be modeled in discrete time as 1^{st} order Gauss-Markov Process, i.e.,

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{w}_{t-1}; t = 0, 1, 2, ... \tag{1}$$

where, the overall system state is represented by the n-dimensional state vector $\mathbf{x}_t$ at time instant t. The initial values of the state vector elements at $t = -1$ follow Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Unlike [6], the state transition matrix $\mathbf{F} \in \mathbb{R}^{n\times n}$ is a general square matrix with spectral radius less than unity. The process noise $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$. The underlying physical system is observed by N agents. The linear observation model for the k^{th} agent is,

$$\mathbf{y}_{t,k} = \mathbf{H}_k \mathbf{x}_{t,k} + \mathbf{v}_{t,k}; k = 1, 2, ..., N \tag{2}$$

where, the n_k-dimensional vector $\mathbf{x}_{t,k}$ is Agent k's local state vector $-$ a subset of $\mathbf{x}_t$. The observation matrix $\mathbf{H}_k \in \mathbb{R}^{m_k \times n_k}$, $(m_k \leq n_k)$ and the measurement noise $\mathbf{v}_{t,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$. Unlike conventional distributed Kalman filtering, a particular agent k attempts to estimate only the local state vector $\mathbf{x}_{t,k}$ instead of $\mathbf{x}_t$. To incorporate this scenario, we introduce a binary projection matrix $\mathbf{T}_k$ such that,

$$\mathbf{x}_{t,k} = \mathbf{T}_k \mathbf{x}_t; k = 1, 2, ..., N. \tag{3}$$

It should be noted that, $\mathbf{T}_k$ is an $n_k \times n$ matrix $(n_k < n)$. The static set of *physical* neighbors for the k^{th} agent is defined based on the overlap/sharing of state elements. Mathematically,

$$\mathcal{S}_k = \{i : \mathbf{P}_{i,k}\mathbf{x}_{t,i} \text{ projects onto } \mathbf{L}_{i,k}\mathbf{x}_{t,k}, \forall t\} \tag{4}$$

where, $\mathbf{P}_{i,k}$ and $\mathbf{L}_{i,k}$ are $n_k \times n_i$ and $n_k \times n_k$ binary projection matrices, respectively. By default, $\mathbf{P}_{k,k} = \mathbf{L}_{k,k} = \mathbf{I}_{n_k}$.

Using the projection matrix $\mathbf{T}_k$, the system dynamics of equation (1) can be written as,

$$\mathbf{x}_{t,k} = \mathbf{T}_k \mathbf{F} \mathbf{x}_{t-1} + \mathbf{w}_{t-1,k}, \tag{5}$$

where, $\mathbf{w}_{t-1,k} = \mathbf{T}_k \mathbf{w}_{t-1}$. Therefore, $\mathbf{w}_{t-1,k} \in \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$. It should be noted that, $\mathbf{Q}_k = \mathbf{T}_k \mathbf{Q} \mathbf{T}_k^\top$. The desired dynamical model for k^{th} agent corresponds to,

$$\mathbf{x}_{t,k} = \mathbf{F}_k \mathbf{x}_{t-1,k} + \mathbf{w}_{t-1,k}, \tag{6}$$

where, $\mathbf{F}_k \mathbf{T}_k = \mathbf{T}_k \mathbf{F}$. Following equation (35) of [6], $\mathbf{F}_k = \mathbf{T}_k \mathbf{F} \mathbf{T}_k^\dagger$. Here, $(\cdot)^\dagger$ refers to the pseudo-inverse of a full row-rank matrix.

3 Proposed Method

The agent-based dynamic state estimation procedure is developed with minimum mean square error (MSE) as the metric of interest. The state vector estimated by the k^{th} agent at discrete time instant i is defined as,

$$\hat{\mathbf{x}}_{i,k|j} = \mathbb{E}\left[\mathbf{x}_{i,k} | \mathbf{y}_{0,k}, \mathbf{y}_{1,k}, ..., \mathbf{y}_{j,k}\right]. \tag{7}$$

The corresponding error covariance matrix is,

$$\mathbf{M}_{i,k|j} = \mathbb{E}\left[(\mathbf{x}_{i,k} - \hat{\mathbf{x}}_{i,k|j})(\mathbf{x}_{i,k} - \hat{\mathbf{x}}_{i,k|j})^\top\right]. \tag{8}$$

The first five steps of estimation are performed according to traditional Kalman filtering, which are represented according to the definitions given in equations (7) and (8). For the k^{th} agent,

- Initialization:
$$\hat{\mathbf{x}}_{-1,k|-1} = \boldsymbol{\mu}_k, \mathbf{M}_{-1,k|-1} = \boldsymbol{\Sigma}_k; \tag{9}$$
where $\boldsymbol{\mu}_k = \mathbf{T}_k\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}_k = \mathbf{T}_k\boldsymbol{\Sigma}\mathbf{T}_k^\top$.
- Prediction:
$$\hat{\mathbf{x}}_{t,k|t-1} = \mathbf{F}_k\hat{\mathbf{x}}_{t-1,k|t-1} \tag{10}$$
- Minimum Prediction MSE:
$$\mathbf{M}_{t,k|t-1} = \mathbf{F}_k\mathbf{M}_{t-1,k|t-1}\mathbf{F}_k^\top + \mathbf{Q}_k \tag{11}$$
For this process to be stable the spectral radius of $\mathbf{F}_k$ has to be less than 1. The next two steps are based on k^{th} agent's observation model.
- Minimum MSE:
$$\mathbf{M}_{t,k|t} = \left(\mathbf{M}_{t,k|t-1}^{-1} + \mathbf{H}_k^\top\mathbf{R}_k^{-1}\mathbf{H}_k\right)^{-1} \tag{12}$$
- Intermediate Correction:
$$\hat{\mathbf{b}}_{t,k} = \hat{\mathbf{x}}_{t,k|t-1} + \mathbf{M}_{t,k|t}\mathbf{H}_k^T\mathbf{R}_k^{-1}\left(\mathbf{y}_{t,k} - \mathbf{H}_k\hat{\mathbf{x}}_{t,k|t-1}\right) \tag{13}$$

The steps described so far constitute the "computation phase" of agent-based Kalman filtering with a complexity of $\mathcal{O}(n_k^3)$ (including matrix inversion). This phase does not require any neighborhood communication. At the last step of estimation, $\hat{\mathbf{b}}_{t,k}$ is used along with the exchanged information from the neighbors to arrive at the final estimate of individual agents' local state values. And this step constitute the single "communication phase" in the agent-based formulation. Once the neighborhood information is exchanged, the final correction in local state estimates can be performed using either consensus [11] or uniformly diffusing the exchanged information [1]. These approaches are called agent-based Kalman consensus filter (AKCF) and Diffusion Kalman filter (ADKF), respectively. The mathematical representation of this final correction step is given below,

- Final Correction: AKCF
$$\hat{\mathbf{x}}_{t,k|t} = \hat{\mathbf{b}}_{t,k} + \epsilon\mathbf{M}_{t,k|t}\sum_{i\in\mathcal{S}_k}\left(\mathbf{P}_{i,k}\hat{\mathbf{x}}_{t,i|t-1} - \mathbf{L}_{i,k}\hat{\mathbf{x}}_{t,k|t-1}\right) \tag{14}$$
where, $0 < \epsilon \leq 1$. Larger value of ϵ allows greater contribution of consensus and vice versa.
- Final Correction: ADKF
$$\hat{\mathbf{x}}_{t,k|t} = \mathbf{D}_k\sum_{i\in\mathcal{S}_k\cup\{k\}}\mathbf{P}_{i,k}\hat{\mathbf{b}}_{t,i} \tag{15}$$
where, $\mathbf{D}_k = diag\left(d_k[1], d_k[2], \cdots, d_k[n_k]\right)$. For uniform weighting,
$$d_k[j] = \frac{1}{\text{Number of Agents that share the state } \mathbf{x}_k[j]} \tag{16}$$
Here, $\mathbf{x}_k[j]$ represents the j^{th} element of agent k's local state vector.

The above equation mathematically interpret the uniform diffusion when there is no loss or failure in the communication network. Certainly, each agent is not required to know the *number* of agents sharing its local state elements rather the awareness about its neighborhood. Intuitively, each agent counts the number of neighborhood information it receives for a particular state element and decides the uniform weighting accordingly.

It should be noted that in conventional distributed Kalman filters, each local sensor monitors the whole dynamical process with a computational complexity $\mathcal{O}(n^3)$. Also, the calculations in equations (12) and (13) require two additional communication phases. The first phase is required to fuse the weighted Grammian ($\mathbf{H}_i^\top \mathbf{R}_i^{-1} \mathbf{H}_i$) from all the neighbors. Fusion of weighted measurement ($\mathbf{H}_i^\top \mathbf{R}_i^{-1} \mathbf{y}_{t,i}$) from all the neighbors necessitates the second communication phase. The proposed method curtails these fusions, reduces the dependency over the neighborhood measurement information exchange and computational complexity is drastically reduced.

4 Effect of Communication

In the proposed method of agent-based filtering, it is evident that only the information about relevant state elements are being exchanged among neighbors. This is illustrated in Fig. 1. In AKCF, the k^{th} agent exchanges information about the predicted state elements obtained in equation (10) with its neighbors. Whereas, in ADKF, the intermediate correction vector obtained in equation (13) is exchanged. Therefore, the inter-agent two way information exchange plays an important role in agent-based Kalman filtering and can be hampered if the underlying communication link fails. These circumstances can be simulated by

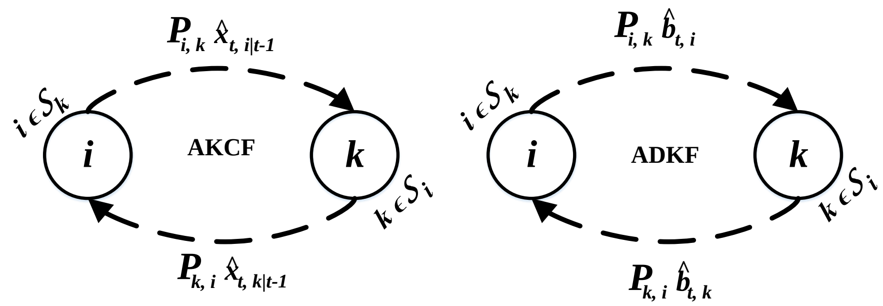

Fig. 1. Inter-agent information exchange

introducing random link failures (RLF). Mathematically, effect of RLF can be analyzed by inserting Bernoulli random variables $\gamma_{i,k}(t)$ in equations (14) and (15). These random variables have the following probability mass functions,

$$\gamma_{i,k}(t) = \begin{cases} 0 \text{ with Prob. } \rho_{i,k} \\ 1 \text{ with Prob. } 1-\rho_{i,k} \end{cases} ;\forall i,k,i \neq k \tag{17}$$

Therefore, the final correction step of AKCF with RLF is,

$$\hat{\mathbf{x}}_{t,k|t} = \hat{\mathbf{b}}_{t,k} + \epsilon \mathbf{M}_{t,k|t} \sum_{i \in \mathcal{S}_k} \gamma_{i,k}(t) \left(\mathbf{P}_{i,k} \hat{\mathbf{x}}_{t,i|t-1} - \mathbf{L}_{i,k} \hat{\mathbf{x}}_{t,k|t-1} \right) \tag{18}$$

It is evident that only the consensus part of equation (18) is affected by communication. On the contrary, the final correction step of ADKF is modified as follows,

$$\hat{\mathbf{x}}_{t,k|t} = \mathbf{D}_{t,k} \sum_{i \in \mathcal{S}_k \cup \{k\}} \mathbf{P}_{i,k} \hat{\mathbf{b}}_{t,i} \tag{19}$$

where, $\mathbf{D}_{t,k} = diag\left(d_{t,k}[1], d_{t,k}[2], \cdots, d_{t,k}[n_k]\right)$. And for uniform weighting,

$$d_{t,k}[j] = \frac{1}{c_{t,k}[j]} \tag{20}$$

where, $c_{t,k}[j]$ represents the number of *Successfully Received* estimates for $\mathbf{x}_k[j]$ at discrete time instant t.

5 Simulation and Results

We investigate the performance of the proposed filters for two 3-agent systems. The set of global state elements for the 1^{st} system `SYS1` is $\{a, b, c, d, e, f\}$ and that for the 2^{nd} system `SYS2` is $\{a, b, c, d, e, f, g, h, i, j, k, l, m\}$. The Venn diagrams in Fig. 2 show the agent-wise distribution of the state elements for the two systems. It should be noted that in `SYS2`, there exist some state elements *strictly* local to the agents, whereas, each of the state elements is shared between two agents in `SYS1`. The rest of the parameters of `SYS1` and `SYS2` are given in Appendix A and B, respectively. The proposed filters are applied to these systems as illustrative examples.

5.1 Figure of Merit

To investigate the performance of AKCF and ADKF, an estimation error vector associated with each agent is calculated. The estimation error of k^{th} agent at discrete time instant t is,
$\boldsymbol{\eta}_{t,k} = \hat{\mathbf{x}}_{t,k|t} - \mathbf{x}_{t,k}$.

A global estimation error vector is formed by stacking $\boldsymbol{\eta}_{t,k}$ from all agents, $\boldsymbol{\eta}_t = \left[\boldsymbol{\eta}_{t,1}^\top \cdots \boldsymbol{\eta}_{t,N}^\top\right]^\top$.

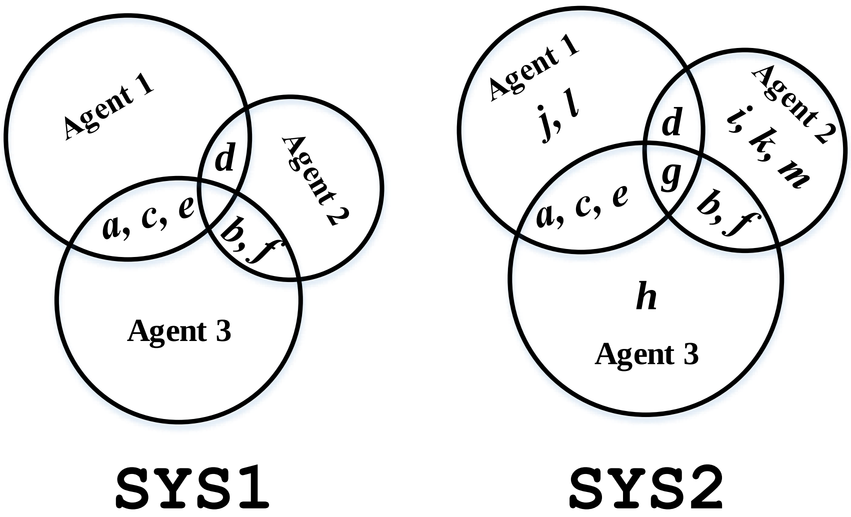

Fig. 2. Venn diagrams for `SYS1` and `SYS2`

This vector is used to define the mean square deviation (MSD) as follows,

$$\mathbf{MSD}_t = trace\left(\mathbb{E}\left[\boldsymbol{\eta}_t\boldsymbol{\eta}_t^\top\right]\right) \tag{21}$$

MSD is used as the figure of merit to compare the performance of AKCF and ADKF. The lower the MSD is, the better. In the upcoming subsections we present the performance of the proposed filters in terms of MSD obtained from simulations.

5.2 Case Study: Perfect Communication

In this scenario, all the inter-agent communication links are assumed to be working perfectly. The proposed ADKF is applied to `SYS1` with uniform weighting rule (equation (16)). AKCF is applied to `SYS1` with different values of ϵ. MSD is calculated from 1000 independent Monte Carlo trials at each time step. The comparative performance of ADKF and AKCF for `SYS1` is shown in Fig. 3. In the same way, AKCF and ADKF is applied to `SYS2` and the performance is summarized in Fig. 4. The MSD values from Fig. 3 and Fig. 4 differ from the classical distributed filtering approach [1] by showing better performance of AKCF, irrespective of the selection of ϵ within the prescribed range. In particular, the best performance of AKCF is obtained when ϵ is selected to be 0.1 and 0.01 for `SYS1` and `SYS2`, respectively. The optimum values of ϵ thus obtained is used in AKCF for the following case study.

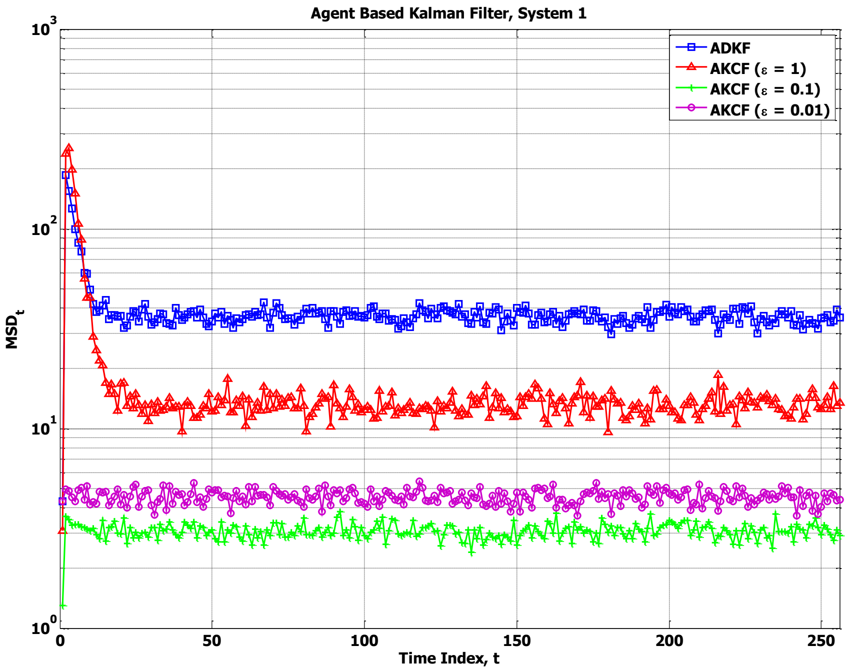

Fig. 3. AKCF versus ADKF for `SYS1`

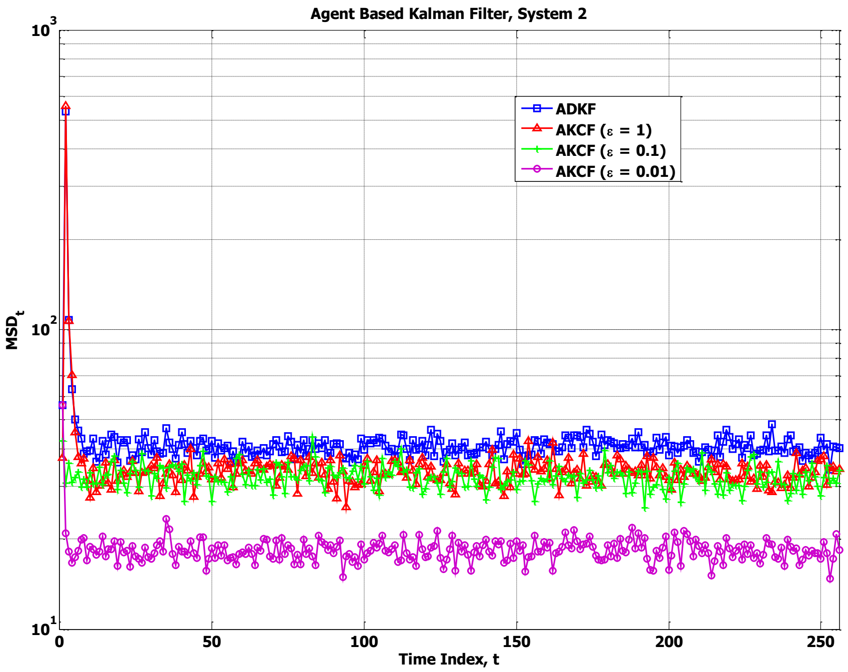

Fig. 4. AKCF versus ADKF for SYS2

5.3 Case Study: Random Link Failure

The effect of inter-agent communication is investigated for `SYS1` and `SYS2` by using equations (18-20) at the final correction steps of the proposed filters. For simplicity, the probability of link failure, $\rho_{i,k} = \rho; \forall i, k, i \neq k$. The MSD is obtained at different link failure rates, which is illustrated in Fig. 5 for `SYS1`. Based on the previous case study, the value of ϵ in AKCF is 0.1. The whole procedure

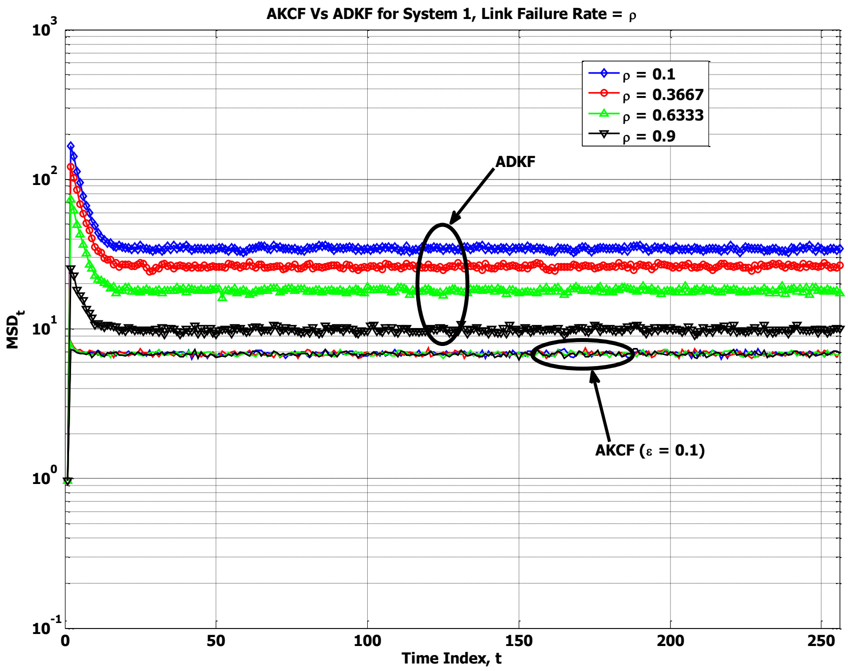

Fig. 5. Effect of communication for `SYS1`

is repeated for `SYS2` with the corresponding value of ϵ in AKCF being 0.01. Fig. 6 shows the relative performance for `SYS2` affected by imperfect communication. The effect of faulty inter-agent communication link is insignificant for AKCF as evident from Fig. 5 and Fig. 6. This is because of relatively small values of ϵ chosen for the two systems. On the other hand, a high link failure rate results in less contribution from neighboring agents in the final correction step of ADKF. It is interesting to see that ADKF performs better when the communication link is highly unreliable. While this may appear counter intuitive, it is in fact a direct consequence of the system parameter choice. These observations suggest that, for the two systems considered in our simulations, the underlying system states are more dependent upon the agent-wise observation space as compared to the system dynamics itself. Nevertheless, the steady-state MSD values are

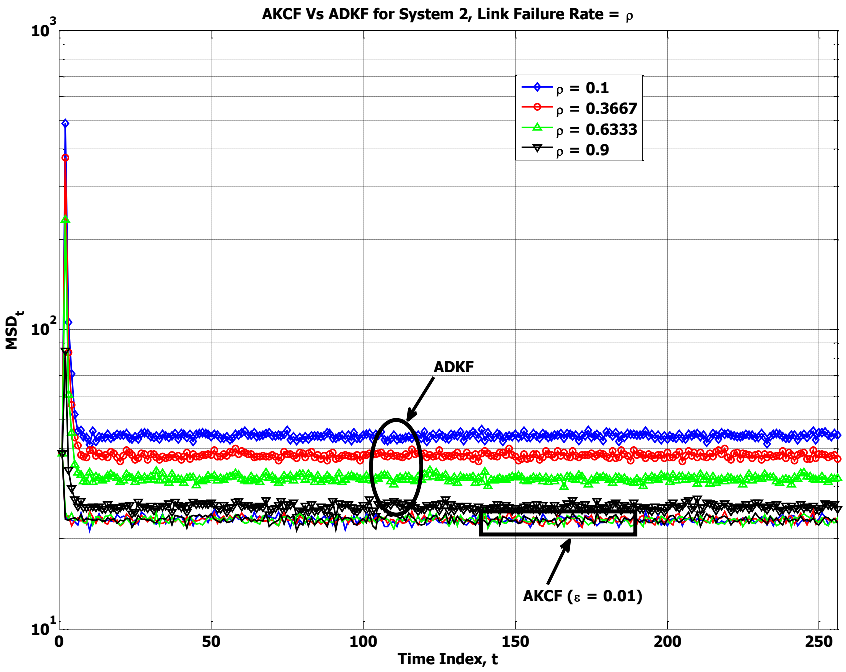

Fig. 6. Effect of communication for SYS2

smaller for AKCF, irrespective of the choice of ϵ value as well as the condition of inter-agent communication link and is more robust than ADKF. However, If the number of agents are increased for a particular dynamical process, the sharing of state elements and size of neighborhood will also increase. As a consequence, the proposed AKCF and ADKF are expected to perform better and exhibit more robustness under faulty communication network.

6 Conclusions and Future Work

An agent-based Kalman consensus and diffusion Kalman filter is proposed. Each agent is interested in a distinct subset of state elements and is able to communicate to its neighbors who share at least one state element. The proposed filtering procedures are applied to two multi-agent systems to compare their performance. The effect of communication is also observed for the agent-based Kalman filters. It is observed that AKCF performs better than ADKF for both systems even under random failure of inter-agent communication link. In the future work, it would be worthwhile to investigate the effect of coupling strength of observation space and system dynamics over the performance of AKCF and ADKF under both perfect and faulty communication link.

Acknowledgment

The authors would like to thank the NSF for providing support for this research through award No. CNS-1136040. The views expressed in this paper are of the authors.

References

1. Cattivelli, F., Sayed, A.: Diffusion strategies for distributed kalman filtering and smoothing. Automatic Control, IEEE Transactions on 55(9), 2069–2084 (2010)
2. Dong, H., Wang, Z., Gao, H.: Distributed filtering for a class of time-varying systems over sensor networks with quantization errors and successive packet dropouts. Signal Processing, IEEE Transactions on 60(6), 3164–3173 (2012)
3. Dorfler, F., Pasqualetti, F., Bullo, F.: Continuous-time distributed observers with discrete communication. Selected Topics in Signal Processing, IEEE Journal of 7(2), 296–304 (April 2013)
4. Farag, H., El-Saadany, E., Seethapathy, R.: A two ways communication-based distributed control for voltage regulation in smart distribution feeders. Smart Grid, IEEE Transactions on 3(1), 271–281 (2012)
5. Kalman, R.E., et al.: A new approach to linear filtering and prediction problems. Journal of basic Engineering 82(1), 35–45 (1960)
6. Khan, U., Moura, J.: Distributing the kalman filter for large-scale systems. Signal Processing, IEEE Transactions on 56(10), 4919–4935 (2008)
7. Lo, C.H., Ansari, N.: Decentralized controls and communications for autonomous distribution networks in smart grid. Smart Grid, IEEE Transactions on 4(1), 66–77 (2013)

8. Mahmoud, M., Khalid, H.: Distributed kalman filtering: a bibliographic review. Control Theory Applications, IET 7(4), 483–501 (2013)
9. Mohammadi, A., Asif, A.: Distributed state estimation for large-scale nonlinear systems: A reduced order particle filter implementation. In: Statistical Signal Processing Workshop (SSP), 2012 IEEE. pp. 249–252 (Aug 2012)
10. Olfati-Saber, R.: Distributed kalman filtering for sensor networks. In: Decision and Control, 2007 46th IEEE Conference on. pp. 5492–5498 (2007)
11. Olfati-Saber, R.: Kalman-consensus filter : Optimality, stability, and performance. In: Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on. pp. 7036–7042 (2009)
12. Yang, W., Wang, X., Shi, H.: Optimal consensus-based distributed estimation with intermittent communication. International Journal of Systems Science 42(9), 1521–1529 (2011)
13. Zhao, X., Tu, S.Y., Sayed, A.: Diffusion adaptation over networks under imperfect information exchange and non-stationary data. Signal Processing, IEEE Transactions on 60(7), 3460–3475 (2012)

Appendix A: `SYS1` Parameters

– State initialization: $\boldsymbol{\mu} = \begin{bmatrix}10 \ 5 \ 20 \ 2.5 \ 40 \ 1.25\end{bmatrix}^\top$; $\boldsymbol{\Sigma} = diag(.8, .2, .5, 1.3, .1, .3)$;

– Global state transition matrix: $\mathbf{F} =$

$$\begin{bmatrix} 0.3153 & 0.0090 & 0.0541 & 0.2342 & 0.1712 & 0.2162 \\ 0.0270 & 0.2883 & 0.0631 & 0.1892 & 0.2072 & 0.2252 \\ 0.2793 & 0.0811 & 0.0180 & 0.1982 & 0.2432 & 0.1802 \\ 0.0721 & 0.2523 & 0.2973 & 0.1532 & 0.0901 & 0.1351 \\ 0.2703 & 0.0450 & 0.3063 & 0.1081 & 0.1261 & 0.1441 \\ 0.0360 & 0.3243 & 0.2613 & 0.1171 & 0.1622 & 0.0991 \end{bmatrix};$$

– Process noise: $\mathbf{Q} = diag(1.8, .9, 2.7, 3.6, 1.0, .5)$;

– Agent-wise observation model:

$$\mathbf{H}_1 = \begin{bmatrix} 13.3758 & 3.2277 & 5.2820 & 18.3034 \\ 6.7976 & 0.3931 & 4.9316 & 4.1200 \\ 8.1333 & 6.3871 & 13.2687 & 13.6812 \end{bmatrix};$$

$\mathbf{R}_1 = diag(0.9217, 0.3057, 0.7316)$;

$$\mathbf{H}_2 = \begin{bmatrix} 41.4826 & 10.9732 & 27.7157 \\ 22.6780 & 28.1071 & 39.7014 \end{bmatrix};$$

$\mathbf{R}_2 = diag(0.0019, 0.2653)$;

$$\mathbf{H}_3 = \begin{bmatrix} 41.6863 & 49.7289 & 39.0787 & 26.8042 & 4.2471 \\ 13.5406 & 48.6878 & 63.23 & 7.3416 & 14.5855 \\ 15.2408 & 46.3493 & 2.1896 & 73.8971 & 73.7868 \\ 68.2435 & 38.4299 & 46.5694 & 43.9841 & 10.4298 \end{bmatrix}; \mathbf{R}_3 = diag(0.3703, 0.3765, 0.2747, 0.3410).$$

Appendix B: SYS2 Parameters

– Dynamics of SYS2:

$$\boldsymbol{\mu} = \begin{bmatrix} 10\ 5\ 20\ 2.5\ 40\ 1.25\ 4\ 3.7\ 25.8\ 21.4\ 76\ 51.6\ 88.9 \end{bmatrix}^\top;$$

$$\boldsymbol{\Sigma} = diag(.8, .2, .5, 1.3, .1, .3, .4, 1, .7, 1.2, .9, 3.9, 5.7);$$

$$\mathbf{Q} = diag(1.8, 0.9, 2.7, 3.6, 1, 0.5, 0.1, 4.5, 2, 8, 5, 1.5, 0.3);$$

$$\mathbf{F} = \begin{bmatrix} 0.0842 & 0.0977 & 0.1113 & 0.1249 & 0.1385 & 0.1520 & 0.0009 & 0.0145 & 0.0281 & 0.0416 & 0.0552 & 0.0688 & 0.0824 \\ 0.0968 & 0.1104 & 0.1240 & 0.1376 & 0.1511 & 0.0118 & 0.0136 & 0.0271 & 0.0407 & 0.0543 & 0.0679 & 0.0814 & 0.0833 \\ 0.1095 & 0.1231 & 0.1367 & 0.1502 & 0.0109 & 0.0127 & 0.0262 & 0.0398 & 0.0534 & 0.0670 & 0.0805 & 0.0941 & 0.0959 \\ 0.1222 & 0.1357 & 0.1493 & 0.0100 & 0.0235 & 0.0253 & 0.0389 & 0.0525 & 0.0661 & 0.0796 & 0.0932 & 0.0950 & 0.1086 \\ 0.1348 & 0.1484 & 0.0090 & 0.0226 & 0.0244 & 0.0380 & 0.0516 & 0.0652 & 0.0787 & 0.0923 & 0.1059 & 0.1077 & 0.1213 \\ 0.1475 & 0.0081 & 0.0217 & 0.0353 & 0.0371 & 0.0507 & 0.0643 & 0.0778 & 0.0914 & 0.1050 & 0.1068 & 0.1204 & 0.1339 \\ 0.0072 & 0.0208 & 0.0344 & 0.0362 & 0.0498 & 0.0633 & 0.0769 & 0.0905 & 0.1041 & 0.1176 & 0.1195 & 0.1330 & 0.1466 \\ 0.0199 & 0.0335 & 0.0471 & 0.0489 & 0.0624 & 0.0760 & 0.0896 & 0.1032 & 0.1167 & 0.1186 & 0.1321 & 0.1457 & 0.0063 \\ 0.0326 & 0.0462 & 0.0480 & 0.0615 & 0.0751 & 0.0887 & 0.1023 & 0.1158 & 0.1294 & 0.1312 & 0.1448 & 0.0054 & 0.0190 \\ 0.0452 & 0.0588 & 0.0606 & 0.0742 & 0.0878 & 0.1014 & 0.1149 & 0.1285 & 0.1303 & 0.1439 & 0.0045 & 0.0181 & 0.0317 \\ 0.0579 & 0.0597 & 0.0733 & 0.0869 & 0.1005 & 0.1140 & 0.1276 & 0.1412 & 0.1430 & 0.0036 & 0.0172 & 0.0308 & 0.0443 \\ 0.0706 & 0.0724 & 0.0860 & 0.0995 & 0.1131 & 0.1267 & 0.1403 & 0.1421 & 0.0027 & 0.0163 & 0.0299 & 0.0434 & 0.0570 \\ 0.0715 & 0.0851 & 0.0986 & 0.1122 & 0.1258 & 0.1394 & 0.1529 & 0.0018 & 0.0154 & 0.0290 & 0.0425 & 0.0561 & 0.0697 \end{bmatrix};$$

– Agent-wise observation model:

$$\mathbf{H}_1 = \begin{bmatrix} 5.03 & 1.9 & 19.88 & 15.16 & 5.23 & 2.1 & 18.42 \\ 14.55 & 4.19 & 2.18 & 8.78 & 8.23 & 4.46 & 12.04 \\ 4.82 & 7.87 & 17.23 & 3.49 & 13.23 & 17.94 & 10.37 \\ 6.91 & 2.91 & 4.45 & 7.97 & 4.73 & 12.60 & 7.36 \end{bmatrix};$$

$$\mathbf{R}_1 = diag(0.84, 0.16, 0.29, 0.37);$$

$$\mathbf{H}_2 = \begin{bmatrix} 15.45 & 25.53 & 6.17 & 46.70 & 16.23 & 41.32 & 32.43 \\ 5.06 & 39.37 & 31.16 & 0.27 & 32.76 & 8.01 & 20.26 \\ 39.33 & 18.28 & 44.17 & 27.31 & 6.27 & 27.23 & 13.33 \\ 42.76 & 8.67 & 35.13 & 49.78 & 44.14 & 15.19 & 22.33 \\ 17.46 & 39.27 & 48.45 & 31.98 & 15.56 & 6.85 & 44.79 \\ 9.14 & 27.59 & 2.34 & 34.65 & 34.39 & 42.29 & 30.44 \end{bmatrix};$$

$$\mathbf{R}_2 = diag(0.47, 0.23, 0.82, 0.32, 0.58, 0.42);$$

$$\mathbf{H}_3 = \begin{bmatrix} 8.19 & 75.38 & 68.21 & 77.34 & 13.89 & 58.98 & 73.77 \\ 15.3 & 38.88 & 68.12 & 45.0 & 56.2 & 65.84 & 17.77 \\ 18.96 & 20.23 & 66.23 & 61.85 & 51.02 & 18.76 & 58.98 \end{bmatrix}; \mathbf{R}_3 = diag(0.45, 0.81, 0.53).$$