On Scalable Distributed Sensor Fusion

KC Chang
Dept. of SEOR
George Mason University
Fairfax, VA 22030, USA
kchang@gmu.edu

Chee-Yee Chong and Shozo Mori
BAE Systems
Advanced Information Technologies
Los Altos, CA 94022, USA
{chee.chong, shozo.mori}@baesystems.com

Abstract - The theoretic fundamentals of distributed information fusion are well developed. However, practical applications of these theoretical results to dynamic sensor networks have remained a challenge. There has been a great deal of work in developing distributed fusion algorithms applicable to a network centric architecture. In general, in a distributed system such as ad hoc sensor networks, the communication architecture is not fixed. In those cases, the distributed fusion approaches based on pedigree information may not scale due to limited communication bandwidth. In this paper, we focus on scalable fusion algorithms and conduct analytical performance evaluation to compare their performance. The goal is to understand the performance of these algorithms under different operating conditions. Specifically, we evaluate the performance of Channel filter fusion, Chernoff fusion, Shannon Fusion, and Bhattacharyya fusion algorithms. We also compare their performance to Naïve fusion and “optimal” centralized fusion under a specific communication pattern.

Keywords: distributed fusion, net-centric, autonomous sensor, scalable algorithm, fusion performance analysis.

1 Introduction

A distributed data fusion system consists of a network of sensors, each capable of local processing and fusion of sensor data. The nature of distributed-network fusion processing involves three factors that define network performance: the structure of the network architecture, reliability of communication links within the network, and applicable fusion algorithms. The structure of the network and the reliability of communication links are conditions that cannot be predicted yet they largely define the impact of fusion algorithm performance.

There has been a great deal of work in developing distributed fusion algorithms applicable to a network centric architecture. However, in general, in a distributed system such as ad hoc sensor networks, the communication architecture is changing dynamically. In those cases, the distributed fusion algorithm based on information graph approach [10] may not scale due to its requirements to carry long pedigree information for decorrelation.

In this paper, we will focus on several scalable fusion algorithms and analytically compare their performance through steady-state estimate error prediction. To demonstrate our performance analysis approach, we use a nominal three-node fusion processing scenario with cyclic communications as shown in Figure 1. We conduct extensive simulation to validate the theoretical predictions. We have chosen this network structure because of its complexity due to multiple paths of information propagation, and the availability of the optimal analytical solution that can be derived and used as a performance baseline.

Specifically, we consider the fusion algorithms listed below and compare their performance against the optimal information fusion solution:

- Naïve Fusion
- Channel Filter
- Chernoff Fusion
- Shannon Fusion
- Bhattacharyya Fusion

Our goal is to investigate how these different fusion algorithms perform for a specific scenario under limited communication bandwidth. This is part of a wider objective to understand the system trades involved in a general decentralized ad hoc sensor networks. This paper is organized as follows. Section 2 briefly describes the set of scalable distributed fusion algorithms to be considered in this paper. Section 3 derives the analytical fusion performance evaluation in terms of steady-state mean square error. Section 4 summarizes the technical findings of the study and Section 5 presents some future research directions.

Figure 1. A Three Sensor Cyclic Communication Scenario
2 Scalable Fusion Algorithms

The theoretic fundamentals of distributed information fusion are well documented and have been studied in depth [6-10]. It is noted, however, that practical applications of these theoretical results to non-deterministic information flow has remained a challenge. The main difficulty is the need to identify and remove common information from data sets to be fused, while minimizing the amount of data exchanged between agents.

The basic fusion process as described in [10] follows from set theory, where the combination of \( n \) event probabilities \( \Phi(I_j) \) given the information \( I_j \) can be represented as:

\[
\Phi\left(\bigcup_{j=1}^{n} I_j\right) = \frac{1}{C} \prod_{j=1}^{n} S_j
\]

where \( S_j \) represents the combination of \( i \) event probabilities such that, \( S_j = \prod_{i=1}^{n} \Phi(I_j) \), \( S_1 = \prod_{i=1}^{n} \Phi(I) \), \( S_2 = \Phi(I_1 \cap I_2) \), \ldots, \( S_n = \Phi(I_1 \cap I_2 \cap \ldots \cap I_n) \).

The alternating multiplication and division of joint probabilities from (1) removes conditional dependencies from the data sets in the form of shared information.

While the removal of duplicate information is straightforward in the theoretical formulation, identification of duplicate information for distributed estimation systems can be difficult in practical implementations. The difficulty is due to the need to recognize correlated information resulting from past fusion events and know the values of their data sets. The Information Graph (IG) technique presented in [6-7,10] provides an analytical tool for identifying duplicate information in distributed estimation systems. The approach is a symbolic representation of the collection, propagation, and fusing of data among a set of fusion agents. An example of an Information Graph is shown in Figure 1, where a simple cyclical communications pattern is demonstrated. Each numbered row of symbols represents the events of a given agent. Within each time step, each agent may perform time updates of estimates, receive sensor data, perform measurement updates, transmit the local estimate to other agents, and fuse estimates received from other agents.

The difficulty with the information graph approach is that it is communication pattern dependent—it needs to consider all relevant common priors, and to remove the common information at these nodes from the current track update. Determining these nodal connections over a varying network can be difficult and time-consuming. For example, in the simple three sensor cyclic communication network shown in Figure 1, the resulting formula for the fusion between the first two sensors at time \( k \) is [10],

\[
p(x) = \frac{1}{c} \frac{P_1(x) p_2(x) p_{1,2}(x)}{\mathbb{P}(x)}
\]

where \( c \) is the normalization constant, \( p(x) \) is the conditional probability at node \( s1 \) after fusion, and \( p_{1,2}(x) \) is the conditional probability at node \( si \) and time \( k \) before fusion. In the case when all probability densities are Gaussian, the fusion formula becomes (see Figure 1),

\[
P_{k+1} = P_{k+1}(x) + P_{k+1}(x) - P_{k+1}(x) + P_{k+1}(x)
\]

In general, in order to construct the "optimal" fusion formula, it may require carrying long pedigree information \(^2\) that might not be practical in an environment with limited communication bandwidth [15].

To address the scalability issue, each of the fusion algorithms described below have been developed for autonomous sensors in arbitrary network conditions. All of these approaches are sub-optimal in general, but provide adequate performance when basic assumptions are met.

**Channel Filter:** The Channel filter approach [11][12] is simpler than the information graph fusion in that only the first order redundant information is considered. Each channel is defined by a pair of transmitting agent and receiving agent. The transmitting agent for a particular channel is responsible for removing redundant information, as such, it needs only keep track of the previous transmission from itself to the receiving node.

However, in a dynamic ad hoc network, the transmitting data may never reach the receiving end due to link uncertainty. Therefore, another idea is to have the receiving agent of a particular channel be responsible for removing the redundant information. In this way, the receiving agent only needs to keep track of the previous data transmitted to or received from the channel at the previous communication time and remove it when combining the current estimates. There is no need to maintain long histories of previous activity. In a sense, this can be considered as a first order approximation to the information graph approach.

Specifically, the Channel filter fusion equation is given as,

\[
p(x) = \int \frac{P_1(x) p_2(x) / \mathbb{P}(x)}{\int p_1(x) p_2(x) / \mathbb{P}(x) dx}
\]

where \( p_1(x) \) and \( p_2(x) \) are the two probability density functions to be fused (one local and the other received

---

1 Information graph approach is optimal when the underlying system is deterministic.

2 Information includes communication and fusion events history as well as past fusion data.
from a particular channel) and \( p(x) \) is the density function received from the same channel at the previous communication time and is the common “prior information” to be removed in the fusion formula. When both \( p_1(x) \) and \( p_2(x) \) are Gaussian density with mean and covariance \( \hat{x}_1, P_1 \) and \( \hat{x}_2, P_2 \) respectively, the fused state estimation and corresponding covariance error can be written as:

\[
P^{-1} = P_1^{-1} + P_2^{-1} - \bar{P}^{-1}
\]

(5)

\[
P^{-1}\hat{x} = P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2 - \bar{P}^{-1}\bar{x}
\]

While simpler, it is obvious that dependent information could be lost in Channel filter when compared to the information graph approach. On the other hand, if the time between when that redundancy occurred and the current processing time is noticeably long, the impact could be minimum.

Naïve Fusion: Naïve fusion is the simplest fusion approach where it is assumed that the dependency between density functions is negligible. This fusion approach represents the simplest type and it could be unreliable. The Naïve fusion formula can be written as:

\[
p(x) = \frac{p_1(x)p_2(x)}{p_1(x)p_2(x)dx}
\]

(6)

For Gaussian case, the fused state estimation and corresponding covariance error are shown as:

\[
P^{-1} = P_1^{-1} + P_2^{-1}
\]

\[
P^{-1}\hat{x} = P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2
\]

(7)

Note that the fused track covariance is the inverse of the sum of the inverses of the local track covariance matrices. Thus, due to the lack of common prior information, the fused covariance could be much smaller, which can lead to overconfidence. Also when the common prior has very large covariance, (7) is equivalent to (5).

Chernoff Fusion: When the dependency between two distributions is unknown, one idea is to use the Chernoff information [13]. The fusion formula is based on the following:

\[
p(x) = \frac{p_1^w(x)p_2^{1-w}(x)}{p_1^w(x)p_2^{1-w}(x)dx}
\]

(8)

where \( w \in [0,1] \) is an appropriate parameter which minimizes a chosen criteria. When the criterion to be minimized is the Chernoff information as defined in the denominator of (8), we call it Chernoff fusion. It can be shown that the resulting fused density function that minimizes the Chernoff information is the one “halfway” between the two original densities in terms of the Kullback Leibler distance [13, p.312]. In the case when both \( p_1(x) \) and \( p_2(x) \) are Gaussian, the resulting fused density is also Gaussian with mean and covariance obtained as [14],

\[
P^{-1} = wP_1^{-1} + (1-w)P_2^{-1}
\]

\[
P^{-1}\hat{x} = wP_1^{-1}\hat{x}_1 + (1-w)P_2^{-1}\hat{x}_2
\]

(9)

This formula is identical to the covariance intersection (CI) fusion technique [11-12]. Therefore, the CI technique can be considered as a special case of (8). In theory, Chernoff fusion can be used to combine any two arbitrary density functions in a log-linear fashion. However, the resulting fused density may not preserve the same form as the original ones. Also in general, obtaining the proper weighting parameter to satisfy a certain criterion may involve extensive search or computation.

Shannon Fusion: A special case of (8) is when the parameter \( w \) is chosen to minimize the determinant of the fused covariance [14-15]. In Gaussian case, it is equivalent to minimize the Shannon information of the fused density [14]. This is because the Shannon information defined as, \( I_s = -\int p(x)\ln p(x)dx \), can be shown to be equal to \( I_s = -\frac{1}{2}\ln \left(\frac{1}{2\pi e}\right) + \frac{1}{2} \) when \( p(x) \) is Gaussian with covariance \( P \). We call this special case the Shannon fusion. Note that with (9), the Shannon information is a convex function of the parameter \( w \) and therefore the maximum is located at the extreme points (either \( w=0 \) or \( w=1 \)). Moreover, in scalar case where both \( p_1 \) and \( p_2 \) are scalar, the minimum of Shannon information is also located at the extremes.

Bhattacharyya Fusion: Another special case of (8) is when the parameter \( w \) is set to be 0.5. In this case, the denominator of (8) becomes \( B = \int \sqrt{p_1(x)p_2(x)}dx \), which is the Bhattacharyya bound. We call the resulting fusion formula, \( p(x) = \frac{1}{\pi} \int \sqrt{p_1(x)p_2(x)}dx \), the Bhattacharyya fusion. When both \( p_1(x) \) and \( p_2(x) \) are Gaussian, the fusion equation can be written as:

\[
P^{-1} = \frac{1}{2}(P_1^{-1} + P_2^{-1})
\]

\[
P^{-1}\hat{x} = \frac{1}{2}(P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2)
\]

(10)

Therefore, in Gaussian case, Bhattacharyya fusion is similar to the Naïve fusion except the resulting fused covariance is twice as big as that of the Naïve fusion. Note that the fusion equation can be rewritten as

\[
P^{-1} = \frac{1}{2}(P_1^{-1} + P_2^{-1}) = (P_1^{-1} + P_2^{-1}) - \frac{1}{2}(P_1^{-1} + P_2^{-1})
\]

\[
P^{-1}\hat{x} = \frac{1}{2}(P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2)
\]

(11)

Comparing to the Channel filter (Equation (5)), this formula is equivalent to replace the common prior
information by the “average” of the two set of information to be fused. Namely, \( P^{-1} = (P_{1}^{-1} + P_{2}^{-1}) \) and \( P^{-1} = (P_{1}^{-1} P_{2}^{-1} P_{1}^{-1}) \). In other words, instead of removing the common prior information from the previous communication as in the Channel filter case, the common information of Bhattacharyya fusion is approximated by the “average” of the two locally available information sets.

In the next section, we will derive the analytical performance of Channel filter, Naïve fusion, and Bhattacharyya fusion in terms of true steady-state mean square error. We will derive the results based on the specific cyclic communication scenario as given in Figure 1. We will also conduct extensive simulation to evaluate other alternative fusion algorithms.

3 Analytical Performance Prediction

Define the mean and covariance of each node as shown in Figure 1, where at time \( k \) we define \( \hat{x} = \hat{x}_{k}^{i} \), \( P_{0} = P_{i k}^{i} \), and \( \tilde{x} = \tilde{x}_{k-1}^{i} \); \( \bar{P} = P_{i k-1}^{i} \) as the fused state estimates and the associated filter covariances at time \( k \) and \( k-1 \), \( \hat{x} = \hat{x}_{k}^{i} \), \( P_{i k}^{i} \) as the local updated state estimates and the associated filter covariances, and \( \tilde{x} = \tilde{x}_{k-1}^{i}; \tilde{P} = \tilde{P}_{i k-1}^{i} \) as the local updated state estimates and the associated filter covariances at the previous time instance \( k-1 \).

Our goal is to find the steady-state mean square error covariance of the fused estimate, namely, \( \Omega = \lim_{k \rightarrow \infty} E[(\hat{x}_{k}^{i} - x_{k}^{i})(\hat{x}_{k}^{i} - x_{k}^{i})^{T}] \). In the following, we assume that the dynamic system follows a scalar random walk model, namely, \( x_{k+1} = x_{k} + v_{k} \), where \( v_{k} \) is a zero mean Gaussian process noise with variance \( Q \). We further assume that the observation model is similar for the three sensors and is linear Gaussian, i.e., \( z_{k,i} = x_{k} + w_{k,i} \) where \( w_{k,i} \) is a zero-mean Gaussian measurement noise with variance \( R_{i} \) for sensor \( i \). In the following, we assume that the sensors have same quality, i.e., \( R_{i} = R_{i} = R_{i} = R \). Therefore, in steady state, let \( P_{i} = P \), then \( \bar{P} = \tilde{P}_{i k-1} = P_{i k} = P = P \).

(1) Channel Filter

With Channel filter, as shown in (5), the fusion equations is written as,

\[
P_{0}^{-1} = P_{1}^{-1} + P_{2}^{-1} - P_{2,i,k-1}^{-1} \]

\[
\bar{P}^{-1} \hat{x} = P_{1}^{-1} \hat{x}_{1}^{i} + P_{2}^{-1} \hat{x}_{2}^{i} - \tilde{P}_{2,i,k-1}^{-1} \tilde{x}^{i} \]

Equation (13) can be re-written as,

\[
A \triangleq P_{0}^{-1}(\hat{x} - x) = P_{1}^{-1}(\hat{x}_{1} - x) + P_{2}^{-1}(\hat{x}_{2} - x) - \tilde{P}_{2,i,k-1}^{-1}(\tilde{x}^{i} - x) = P_{1}^{-1}(\hat{x} - x) + P_{2}^{-1}(\hat{x} - (P + Q)^{-1}(\bar{x} - x))
\]

\[
\Rightarrow (\hat{x} - x) = P_{0}A
\]

\[
= P_{0}P_{1}^{-1}(\hat{x}_{1} - x) + P_{0}P_{2}^{-1}(\hat{x}_{2} - x) = P_{0}(P + Q)^{-1}(\bar{x} - x)
\]

Therefore,

\[
\Omega \triangleq E[(\hat{x} - x)^{2}] = P_{0}E(AA^{T})P_{0}^{T}
\]

In scalar case,

\[
(\hat{x} - x)^{2} = P_{0}^{2}(\hat{x} - x)^{2} + \tilde{P}_{0}^{2}(\tilde{x} - x)^{2} + \frac{P_{0}^{2}}{(P + Q)^{2}}(\bar{x} - x)^{2}
\]

\[
+ 2P_{0}^{2}(\hat{x} - x)(\tilde{x} - x) = \frac{2P_{0}^{2}(\hat{x} - x)(\bar{x} - x)}{P^{2}} \frac{2P_{0}^{2}(\hat{x} - x)(\bar{x} - x)}{P^{2}}
\]

\[
\Rightarrow \Omega = \frac{2P_{0}^{2}}{P^{2}}(B + E) + \frac{P_{0}^{2}}{(P + Q)^{2}}\frac{2P_{0}^{2}}{P^{2}}(C_{i} + C_{j})
\]

Note that in (17), \( P_{0} \) and \( P \) are the steady-state “filter” variances. They can be obtained by solving the following two equations,

\[
P_{0}^{-1} = P_{1}^{-1} + P_{2}^{-1} - \tilde{P}_{2,i,k-1}^{-1} = 2P^{-1} - (P + Q)^{-1}
\]

\[
\Rightarrow P_{0} = P(P + Q)/(P + 2Q)
\]

\[
P = (P_{0} + Q)^{-1} - KSK' = (P_{0} + Q)^{-1} - \frac{(P_{0} + Q)^{2}}{(P_{0} + Q + R)} = \frac{(P_{0} + Q)(R)}{(P_{0} + Q + R)}
\]

where \( K' = (P_{0} + Q)/(P_{0} + Q + R) \) is the steady-state Kalman gain, and \( S = (P_{0} + Q + R) \) is the steady state innovation variance. From (22) and (23), it can be easily shown that

\[
P^{2} + (2Q)P^{2} + (2Q^{2})P - (2Q^{2})R = 0
\]

A close-form real solution of the above cubic polynomial can be solved and the resulting \( P_{0} \) as a function of various \( Q \) and \( R \) is shown in Figure 2.

Note that the filter variance is not the same as the true mean square error. To obtain the true mean square error as given in (17), we will need to derive each of the three terms listed in (19)-(21). It can be shown that (the details are omitted),

\[
B = (1 - K)'\Omega + (1 - K)'Q + K' R \triangleq \lambda \Omega + \alpha
\]
Figure 2. The Steady-State Filter Variances $P$ and $P_0$ for Various $Q$ and $R$

\[ E' = (1 - K) \left\{ E \left[ (x_{k+1} - x_k)(x_{k+1} - x_k) \right] + Q \right\} \]
\[ \hat{E}_p = (1 - K) \left[ E_p + Q \right] \]
\[ E_p = \left( \frac{P}{P + Q} \right)^2 \left( 3E' + (E' + Q) \right) \left( 1 + \frac{P^2}{P(P + Q)} \right) \]
\[ C_1 = C_2 = \frac{(1 - K)(\frac{P}{P + Q})^2 B + \frac{P}{P + Q} E' + Q}{1 + (1 - K)(\frac{P}{P + Q})^2} \]
\[ \Omega = \eta(E' + B + \eta') \]
\[ D = \eta(2E') + \eta' \]

With (25)-(29), (17) can be obtained as,
\[ \Omega = \frac{2P_0^2}{P^2} (B + E') + \frac{P^2}{P(P + Q)} E' - \frac{2P_0^2}{P(P + Q)} \]
\[ \Omega = \left( \frac{2P_0^2}{P^2} B + \frac{P_0^2}{P(P + Q)} \right) E' - \frac{2P_0^2}{P(P + Q)} \]
\[ \eta = \frac{1}{4\left( E'[\hat{x}_k - x_k]^2 + E[(\hat{x}_k - x_k)^2] + 2E[(\hat{x}_k - x_k)(\hat{x}_2 - x_2)] \right)} \]

From (25), $B = \lambda \Omega + \alpha$, and from (26),
\[ E' = E[(\hat{x}_k - x_k)^2(\hat{x}_2 - x_2)] \]
\[ \Rightarrow \Omega = \frac{m_0 \alpha + m_2}{1 - m_0 \lambda} \]

Figure 3 compares the analytical mean square errors (MSE) based on (30) with the average MSE based on 1,000 Monte Carlo simulation trials. It is clear that they are in perfect agreement. Figure 3 also shows that the filter variance, $P_0$, are very close to the true MSE which indicates that the algorithm behaves well and is reasonably consistent [9].

\[(29) \text{Naive Fusion}\]

With the notations defined earlier, the Naïve fusion equations can be written as,
\[ P_0 = \left( \frac{P^{(-1)} + P^{(-1)}}{2} \right) = P/2 \]
\[ \hat{x} = P_0 \left( \frac{P^{(-1)} \hat{x}_1 + P^{(-1)} \hat{x}_2}{2} \right) \]

From (31),
\[ \Omega = E[(\hat{x} - x)^2] \]
\[ \frac{1}{4} E[(\hat{x} - x)^2] + E[(\hat{x}_2 - x_2)^2] + 2E[(\hat{x}_1 - x)(\hat{x}_2 - x)] \]

where, as defined before, $B = E[(\hat{x}_1 - x)^2] = E[(\hat{x}_2 - x)^2]$

From (25), $B = \lambda \Omega + \alpha$, and from (26),
\[ E' = E[(\hat{x}_1 - x)^2(\hat{x}_2 - x_2)] \]
\[ \Rightarrow \Omega = \frac{(1 - K)^2}{4 - 3(1 - K)} \Omega \]

Therefore, from (25), (33) and (34), we have,
\[ \Omega = \frac{1}{2} (B + E') = \frac{1}{2} \left( 1 + \mu \right) B + 2\mu Q \]
\[ \Rightarrow \Omega = \frac{1}{2} \left( 1 + \mu \right) (\lambda \Omega + \alpha) + 2\mu Q \Rightarrow \Omega = \frac{(1 + \mu) \alpha / 2 + 2\mu Q}{1 - (1 + \mu) \lambda / 2} \]

Note that in (31), $P_0$ and $P$ are the steady-state “filter” variances which are not the same as the true mean square.
errors. They can be obtained by solving the following two equations,

\[ P_0^{-1} = P_1^{-1} + P_2^{-1} = 2P^{-1} \Rightarrow P_0 = P/2 \]  
(36)

\[ P = (P_0 + Q) - KSK' = (P_0 + Q) - \frac{(P_0 + Q)^2}{(P_0 + Q + R)} = \frac{(P/2 + Q + R)}{(P/2 + Q + R)} \]  
(37)

From (36) and (37), it can be easily shown that

\[ P^2 + (2Q + R)P - 2QR = 0 \]  
\[ \Rightarrow P = \sqrt{(2Q + R)^2 + 8QR - (2Q + R)} \]  
(38)

Figure 4 compares the analytical mean square errors (MSE) based on (35) with the average MSE based on 1,000 Monte Carlo simulation trials. It is clear that they are very close to each other when process noise is not very small. However, when the process noise is extremely small (\( \leq 10^{-4} \)), the simulation results are slightly lower than the analytical prediction. This could be due to the numerical round off error caused by the small magnitude of the noise. Figure 4 also shows that the steady-state “filter” variances, \( P_0 \), are significantly smaller than the true MSE especially when the process noise is not very large. This implies that Naïve fusion is too optimistic and has poor filter consistency [9].

\[ \Omega = E\left[\hat{\tau}^2 - x^2\right] \]  
\[ = \frac{1}{4} \left[ E\left[\hat{x}_1^2 - x_1^2\right] + E\left[\hat{x}_2^2 - x_2^2\right]\right] + 2E[(\hat{x}_1 - x)(\hat{x}_2 - x)] \]  
(41)

\[ = \frac{1}{2} (B + E') \]

where, as defined before, \( B = 2\alpha + \lambda \), and

\[ E' = \frac{(1-K)^2}{4 - 3(1-K)}(B + 4Q) \pm \mu(B + 4Q) \cdot \]

Therefore, as in (35),

\[ \Omega = \frac{(1 + \mu)\alpha/2 + 2\mu Q}{1 - (1 + \mu)\lambda/2} \]  
(42)

Note that the only difference between Naïve and Bhattacharyya fusion is in (38), where \( P_0 \) and \( P \) are the steady-state “filter” variances which can be obtained by solving the following equation,

\[ P = (P_0 + Q) - KSK' = (P_0 + Q) - \frac{(P_0 + Q)^2}{(P_0 + Q + R)} = \frac{(P+QR)}{(P+QR)} \]  
(43)

From (43), it can be easily shown that

\[ P^2 + PQ - QR = 0 \Rightarrow P = \sqrt{Q^2 + 4QR - Q} \]  
(44)

Figure 5 compares the analytical mean square errors (MSE) based on (41) with the average MSE based on 1,000 Monte Carlo simulation trials. Again, they are in perfect agreement. However, as can be seen in the figure, a critical issue with this approach is that the steady-state filter variances are almost twice as large as the true MSE. This indicates that the Bhattacharyya fusion algorithm is too pessimistic and is severely inconsistent.

(3) Bhattacharyya Fusion

As in the Naïve fusion case, the Bhattacharyya fusion equations can be written as,

\[ P_0 = 2\left(P_1^{-1} + P_2^{-1}\right)^{-1} = P \]  
(39)

\[ \hat{x} = \frac{1}{2} P_0 \left(P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2\right) = \left(\hat{x}_1 + \hat{x}_2\right)/2 \]  
(40)

As in (33),
4. Simulation Results and Discussion

In addition to the theoretical analysis for Channel filter, Naïve fusion, and Bhattacharyya fusion, we conducted extensive simulation for Chernoff fusion and Shannon fusion to compare their performances against Naïve fusion and optimal centralized fusion. The results are shown in Figure 6. As can be seen, other than Naïve fusion and Shannon fusion, the remaining algorithms have very similar performance. A closer look (Figure 7) reveals that Channel filter performs close to optimal while Chernoff fusion and Bhattacharyya fusion perform slightly worse. Note that when all sensors have the same quality, Chernoff fusion converges to Bhattacharyya fusion.

We then evaluate the fusion algorithms with different sensor qualities. Instead of homogeneous quality as in the previous case, the sensor measurement error variances are set as 0.5, 1.0, and 2.0 for the three sensors respectively. The results are shown in Figure 8. Figure 8 compares the performance of Channel filter, Chernoff fusion, and Bhattacharyya fusion vs. the optimal fusion. From the figure, it is clear that Channel filter performs the best, Bhattacharyya fusion performs slightly worse, while Chernoff fusion performs the worst among the three particularly when the process noise is large.

To simulate stochastic nature of the communication link, we model the reliability of each link with a probabilistic measure. For example, a link with 0.5 reliability means that the information will pass through the channel only 50% of the time. We then test the three fusion algorithms and their robustness under various link qualities. Since all algorithms under consideration are scalable and autonomous, no additional changes are necessary in the algorithms for the test. The results in Figure 9 show that the performances are in general proportional to the communication quality which is quite intuitive. The results also show that all three algorithms are quite stable and they perform according to expectation.

It should be noted that Channel filter, while requiring a one-step memory in order to retrieve and remove the common prior information in each channel, has a rather simple implementation. On the other hand, the Chernoff fusion algorithm, in addition to its poor filter consistency, needs significantly more computation to search for the optimal weighting factor. Our preliminary experiments show that the Channel filter is at least one order of magnitude faster than the Chernoff fusion. Further investigation is needed to compare the trade-off between these promising algorithms in a more reliable manner.

5. Summary

In this paper, we focus on the analysis and comparison of several scalable algorithms for distributed fusion in a cyclic communication sensor network. Specifically, we evaluate the performance of Channel filter fusion, Chernoff fusion, Shannon Fusion, and Bhattacharyya fusion algorithms. We also compare their performance to Naïve fusion and “optimal” centralized fusion algorithms under a specific communication pattern.

The results show that these scalable algorithms, while requiring minimum communication, perform fairly well. They perform comparably to the optimal fusion algorithm and significantly better than the Naïve fusion method. In particular the Channel filter fusion, representing a first order approximation to the information graph fusion, works surprisingly well.

One of the future research directions is to extend and validate the results to more general network scenarios. In particular, to address the real world network-centric tracking and fusion problems, we will consider heterogeneous sensors with different sampling interval and error characteristics under dynamic communication topology and constraints. We also intend to develop theoretical analysis for specific algorithms whenever possible for a given network scenario.

---

3 In scalar case, Shannon fusion essentially picks the density with smaller variance. Therefore the fusion performance converges to single sensor performance when the sensor qualities are identical.
Comparing MSE with Optimal Fusion (R=1)

Figure 7. Comparing Channel Filter, Chernoff Fusion, and Bhattacharyya Fusion (R1=R2=R3=1)

Figure 8. Channel Filter, Chernoff Fusion, and Bhatcharyya Fusion vs. Optimal Fusion with Sensors of Different Qualities (R1=0.5, R2=1.0, R3=2.0)

Figure 9. Channel Filter vs. Bhattacharyya Fusion with Various Communication Link Qualities (R1=R2=R3=1)

References