## REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*

**1. REPORT DATE** *(DD-MM-YYYY)*
06-28-2011

**2. REPORT TYPE**
Final

**3. DATES COVERED** *(From - To)*
September 2007-June 2011

**4. TITLE AND SUBTITLE**
Distributed Fusion in Sensor Networks with Information Genealogy

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**
N00014-07-1-1211

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**
Dr. Kuo Chu Chang

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
George Mason University
4400 University Drive, MS 4C6
Fairfax, VA 22030

**8. PERFORMING ORGANIZATION REPORT NUMBER**
201405

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Office of Naval Research
100 Alabama St., SW
Suite 4R15
Atlanta, GA 30303-3104

**10. SPONSOR/MONITOR'S ACRONYM(S)**
ONR

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for public release, distribution is unlimited.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

Distributed sensor networks seek to enable adaptive and cognitive behavior in networked information systems. These networks will exhibit truly ad hoc behavior as they adapt in situ to maintain or optimize operations under various conditions. Network topologies and membership may change in response to unpredictable variations in conditions such as spectrum availability, link conditions, power and energy constraints, latency, and routing. As a distributed system of devices, networks must support truly decentralized information exchange, and fusion.

Under the ONR Grant: #N000140711211, George Mason University has been developing a distributed fusion methodology that is both analytically tractable and can be readily implemented in a distributed and autonomous manner. The method is grounded in set-theoretic derivations of information fusion where we develop information genealogy to provide a global view of distributed fusion events for each agent under adverse operating conditions. The technique requires no a priori knowledge of network topology, or communications patterns and is applicable to both low-level and high-level fusion processes with disparate sensors comprised of

**15. SUBJECT TERMS**
Distributed Sensor Network, Information Fusion

| | | | | | |
|---|---|---|---|---|---|
| **16. SECURITY CLASSIFICATION OF:** | | | **17. LIMITATION OF ABSTRACT** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON** |
| **a. REPORT** | **b. ABSTRACT** | **c. THIS PAGE** | | | Dr. Kuo Chu Chang |
| U | U | U | UU | 11 | **19b. TELEPHONE NUMBER** *(Include area code)*<br>(703) 993-1639 |

GEORGE MASON UNIVERSITY

Department of Systems Engineering and Operations Research
4400 University Drive, MS 4A6, Fairfax, Virginia 22030
Phone: 703-993-1670; Fax: 703-993-1521

June 30, 2011

Office of Naval Research
One Liberty Center
875 North Randolph Street, Suite 1177
Arlington, VA 22203-1995
Attention: Code 311: Dr. Behzad Kamgar-Parsi

RE: ONR Grant #N000140711211

Dear Dr. Kamgar-Parsi,

Enclosed please find a copy of our final report for the grant #N000140711211 for period from Oct. 2008 to June 2011.

Sincerely,

K C Chang

KC Chang
Professor, SEOR
George Mason University

Final Technical Report

ONR Grant#N000140711211

**Title:** Distributed Fusion in Sensor Networks with Information Genealogy

Report Performance Period: Sept. 2007 to June 2011

**GMU Technical POC:** Dr. Kuo Chu Chang
Systems Engineering and Operations Research
George Mason University
4400 University Dr., MS 4A6
Fairfax, VA 22030
Voice : (703) 993-1639
Fax : (703) 993-1521
kchang@gmu.edu

## Abstract

Distributed sensor networks seek to enable adaptive and cognitive behavior in networked information systems. These networks will exhibit truly ad hoc behavior as they adapt in situ to maintain or optimize operations under various conditions. Network topologies and membership may change in response to unpredictable variations in conditions such as spectrum availability, link conditions, power and energy constraints, latency, and routing. As a distributed system of devices, networks must support truly decentralized information exchange, and fusion.

Under the ONR Grant: #N000140711211, George Mason University has been developing a distributed fusion methodology that is both analytically tractable and can be readily implemented in a distributed and autonomous manner. The method is grounded in set-theoretic derivations of information fusion where we develop *information genealogy* to provide a global view of distributed fusion events for each agent under adverse operating conditions. The technique requires no *a priori* knowledge of network topology, or communications patterns and is applicable to both low-level and high-level fusion processes with disparate sensors comprised of traditional and non-traditional data types.

This report summarizes our research progress for the performance period from Sept. 2007 to Dec. 2010. Note that GMU received a no-cost extension of the project to June 2011.

## 1. Introduction

Current US Navy and Department of Defense (DoD) networking systems are increasing in utilization and complexity. An ongoing theme across US military operations is the time-intensive, labor-intensive, cognitive effort required to maintain situation awareness for rapid and accurate decision making. Whether the work domain is network operations management, ISR management, or battle management, the common issue is to find and fuse and continuously convert disparate data into actionable information.

In network-centric architectures such as FORCEnet in Navy's operational construct, a global information grid is proposed to be implemented through the use of mobile ad hoc systems to form sensor networks. These networks will have the capability to collect vast amounts of disparate and complementary information from geographically dispersed sources throughout the battlespace. In the architecture, there is neither a fixed central data fusion site nor a central communication facility. Instead, the data are either processed or fused at each network node and these nodes communicate on a point-to-point basis. The network topology, which may be unknown, is assumed to be changing dynamically.

Under the current effort, GMU is developing innovative mathematically rigorous methods for combining data from multiple sources to provide the best estimate of objects and events in the battlespace. Specifically, the key challenge for this research is to develop autonomous fusion algorithms designed for ad hoc wireless network operating under severe communication constraints. These algorithms must be able to scale to large numbers of entities and to combine many disparate types of data.

In particular, we have been working four components of research described below:

- A mathematical foundation for ad hoc sensor networks with arbitrary connectivity and message delays as well as random or non-synchronous local sensing and communication rates while minimizing the amount of data exchanged between agents to ensure accurate and unbiased results.
- A set of practical and robust autonomous fusion algorithms for propagating uncertainty through the integration process and a methodology for the comparison and selection of fusion rules, communication architecture, and deployment configuration of distributed sensor networks.
- Complementary multi-level dynamic Bayesian network (DBN) modeling and inference algorithms that provide the infrastructure to aggregate traditional and non-traditional data from disparate sources at each fusion level.
- A general framework for quantifying the operational characteristics of ad hoc sensor networks, a set of metrics for evaluating the operational performance of a sensor network, and evaluating the fusion performance of multiple, asynchronous sensors of varying quality.

## 2. Project Tasks

The general research tasks for this research are summarized thus:

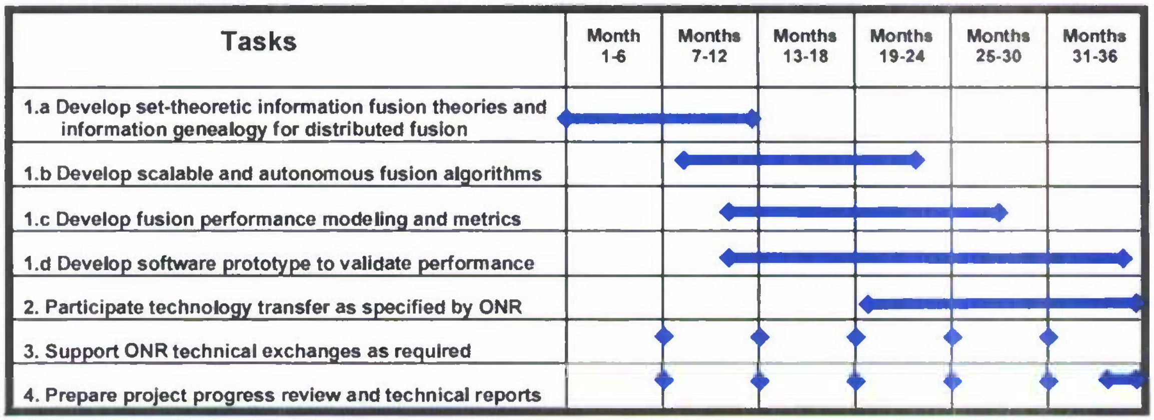
1. Conduct research to develop theories and innovative algorithms and software for distributed sensor system to enable the synergistic fusion and interpretation of data from disparate sensors (traditional and non-traditional data sources)
    - Develop set-theoretic information fusion theories based on information graph and *information genealogy* to provide a solid foundation for distributed fusion.
    - Develop scalable and autonomous fusion algorithms with dynamic communications characteristics to be implemented in distributed agents.
    - Develop fusion performance modeling and evaluation methodologies with a set of defined performance metrics.
    - Develop methodology and software prototype to validate and evaluate the performance of the fusion algorithms. Provide performance assessment for a simulated networking system to be analyzed and validated.
2. Perform model development and engineering analysis as required to support the research initiates as defined by the ONR Program Manager.
    - Perform technical development in collaboration with other performers.
    - Initiate technology transfer to industry or government as specified by ONR.
3. Support the technical exchanges and special studies as required by the ONR Program Manager.
    - Attend and participate in technical interchange meetings at the ONR-specified locations to discuss technical issues related to the research tasks.
    - Lead and participate in special studies as required.
    - Document and distribute the technical findings and the simulation results as needed.
4. Management and Reporting.
    - Prepare monthly financial reports and semi-annual technical progress reports.
    - Prepare annual progress review and comprehensive annual technical reports.

## 3. Project Schedule and Milestones

The project Work Plan Schedule is provided in Table 1. Specific milestones include:

- Preliminary Software Prototype at the end of year 1 and 2
  - Interim MATLAB prototype for initial testing (completed)
  - Test scalability and autonomy through simulation (completed)
- Final Software Prototype available at the end of year 3
  - Verified analytical performance bounds with defined metrics (completed)
  - Confirmed performance prediction through Monte Carlo simulation (completed)
  - Final MATLAB prototype for complete capability testing (completed)
- Transition Readiness Level: TRL 3 at the end of the year 3
  - Basic principles coded, experiments with synthetic data (completed)
  - Limited functionality implementations, experiments with small representative data sets (completed)

**Table 1. Work Plan Schedule**



| Tasks | Month 1-6 | Months 7-12 | Months 13-18 | Months 19-24 | Months 25-30 | Months 31-36 |
|---|---|---|---|---|---|---|
| 1.a Develop set-theoretic information fusion theories and information genealogy for distributed fusion | ◆━━ | ━━◆ | | | | |
| 1.b Develop scalable and autonomous fusion algorithms | | ◆━ | ━━ | ━◆ | | |
| 1.c Develop fusion performance modeling and metrics | | ◆ | ━━ | ━━ | ━◆ | |
| 1.d Develop software prototype to validate performance | | ◆ | ━━ | ━━ | ━━ | ━◆ |
| 2. Participate technology transfer as specified by ONR | | | | ◆━ | ━━ | ━◆ |
| 3. Support ONR technical exchanges as required | ◆ | ◆ | ◆ | ◆ | ◆ | |
| 4. Prepare project progress review and technical reports | ◆ | ◆ | ◆ | ◆ | ◆ | ◆━◆ |

## 6. Project Management

This research project is directed by Dr. KC Chang of George Mason University, who is devoting 20% of his time during the academic year and six weeks during the summer to this research. The research effort is performed by Dr. KC Chang (PI) together with several graduate students. Specifically,

- Two PhD student, Mr. Todd Martin (part-time) and Mr. Rommel Carvalho (full time), who has been working on (1) the development of a mathematical foundation and analytical methodology for distributed genealogy based fusion, (2) defining metrics to quantify the overall performance of the systems, and (3) developing analytical methods to predict fusion performance assessments given the newly developed algorithms.
- One MS student, Mr. Vikas Katori (full time), who has been working on developing (1) a modeling and simulation environment with MATLAB to support specification and performance evaluations, and (2) a set of representative scenarios and the validation of the proposed methodologies under a range of operating conditions.

Note that GMU received a no-cost extension in late 2010. The original project end date was extended from Dec. 2010 to June 2011.

## 7. Technical Progress

The prineipal issues in the design and deployment of sensor network systems include:

- An architecture that decides where and how the sensor reports are fused and the methods to avoid duplicate information
- Methods for optimizing sensor allocation and fusion rules for large-scale programmed or ad hoc networks
- Performance evaluation and trade-off analyses of different design architectures as regards to survivability, performance, data transfer and computational requirements
- Communication issues and bandwidth considerations that impact the choice of data processing and quantization approaches for sharing data amongst fusion nodes

While researchers in the field of sensor and data fusion have advanced significantly during the last decade, these algorithms have been limited for the most part to relatively well-defined network architectures.

### 7.1 Technical Accomplishments

The theoretic fundamentals of distributed information fusion are well documented and have been studied in depth. 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.

In the first two years of the project, we have been developing rigorous mathematical foundation and a set of algorithms for distributed fusion in dynamie networks. In particular, we have been focused on the following researeh:

- A mathematical foundation based on information genealogy for networked sensor fusion with arbitrary eonnectivity and message delays as well as a set of praetical autonomous information fusion and dissemination algorithms. We have documented and published several papers on this area [1-4]. The papers were well received. Specifically, the paper published in Fusion 2008 [1] was the runner-up of the best paper award (top 1% of the 300+ papers). A reprint of the journal paper [8] is attached in the report.
- Complementary multi-level dynamic Bayesian network (DBN) modeling and inferenee algorithms that provide the infrastructure to aggregate traditional and non-traditional data from disparate sourees at eaeh fusion level. We have documented and published several papers on this area [5][7]. Specifically, the paper published in Fusion 2009 [5] received one of the best student paper awards.

Our overall goal is to provide provable methodologies which follow directly from theoretieal developments and to provide quantitative actionable performanee prediction measures.

During the last year of the effort, we have been focused on the following technical areas:

- Scalable inference in distributed hybrid Bayesian network – This is an important area for research but remains a difficult task because of its potentially arbitrary distributions and possible nonlinear dependence relationships between variables. In the past year, we have conducted significant research in this area and have developed a new scalable method under a framework of message passing. We proposed a unified computing scheme of messages propagating between different types of variables. We have documented and published several papers on this area [6][9][12][14]. A reprint of the journal paper [6] is attached in the report.

- Mixture distribution representation and metrics for scalable fusion - Mixture distributions have been used in many applications for dynamic state estimation including distributed tracking, and multisensor fusion. However, the recursive processing of the mixture distributions incurs rapidly growing computational requirements. In order to keep the computational complexity tractable and to ensure scalability while trading-off performance, we developed a recursive mixture reduction algorithm with a given error bound. We have documented and published our work in [11][13]. A reprint of the paper [13] is attached in the report.

- Test real data - We have identified several data sources to test and validate our algorithms. Specifically, the first data set is for under water mine detection with acoustic sonar sensor. The data set is obtained from UC Irvine data repository. We applied and test our algorithm to combine multiple acoustic sensor data to emulate sensor fusion for mine detection. We have obtained some preliminary results and the it will be published in a paper [15]. A reprint of the paper is attached in the report. The second data set is for land mine detection with ground penetrating radar sesnor. This Ground Standoff Mine Detection System (GSTAMIDS) data set is obtained from Dr. Ken Hintz of George Mason University with the permission from Dr. Pete Howard from the Army. Since there was only one type of sensor data, we were not able to emulate and demonstrate the sensor fusion process with this data set.

- Technology transfer - We have been working with several small businesses to apply our technology to other applications. For example, we have been working with Dr. Chris Smith of Decisive Analytic Corporation to apply the scalable fusion technique we developed in this effort for missile defense application [11]. We have also worked with Dr. Craig Agate of Toyon corporation on applying our fusion techniques for ad hoc UAV sensor networks [16].

## REFERENCES

[1] KC Chang, "Distributed Autonomous and Scalable Data Fusion and Dissemination in Net-Centric World," in Proc. 11th International Conferenee on Information Fusion, Germany, July, 2008.

[2] KC Chang, CY Chong, and Shozo Mori, "On Sealablc Distributed Sensor Fusion," in Proc. 11th International Conference on Information Fusion, Germany, July, 2008.

[3] Marty Liggins and K.C. Chang, "Distributcd Fusion Architectures, Algorithms and Performance within a Network Centric Architecture," in Fusion Hand Book, Vol. I, Editcd by David Hall, Sept., 2008.

[4] KC Chang and Vikas Kotari, "An Epidemic Model for Biological Data Fusion in Ad Hoc Sensor Networks," in Proc. SPIE Defense and Security Symposium, Orlando, Florida, April, 2009.

[5] Rommel Carvalho and KC Chang, "A Performance Evaluation Tool for Multi-Sensor Classification Systems," in Proc. 12th International Conference on Information Fusion, Seattle, July, 2009.

[6] Wei Sun and KC Chang, "Message Passing for General Hybrid Bayesian Networks: Representation, Propagation and Integration", *IEEE Trans. on Aerospace and Electronic Systems*, Vol. 45, No. 4, pp. 1525-1537, Oct., 2009.

[7] Rommel Carvalho and KC Chang, "A Performance Evaluation Tool and Analysis for Multi-Sensor Classification Systems," to appear in Journal on Advanced Information Fusion, 2011.

[8] KC Chang, Chee-Yce Chong, and Shozo Mori, "Analytieal and Computational Evaluation of Scalablc Distributed Fusion Algorithms," *IEEE Trans. on Aerospace and Electronic Systems*, Vol. 46, No. 4, pp. 2022-2034, Oct., 2010.

[9] Wei Sun and KC Chang, "Direct Message Passing for Hybrid Bayesian Network and its Performance Analysis," in Proc. SPIE Defense and Security Symposium, Orlando, Florida, April, 2010.

[10] KC Chang, Ashirvad Naik, and Chist Smith, "A comparison of distance metrics between mixture distributions," in Proe. SPIE Defense and Seeurity Symposium, Orlando, Florida, April, 2010.

[11] HD Chen, KC Chang, and Christ Smith, "Constraint Optimized Weight Adaptation for Gaussian Mixture Reduction," in Proc. SPIE Defense and Security Symposium, Orlando, Florida, April, 2010.

[12] Wei Sun, KC Chang, and Kathy Laskey, "Sealable Inferenee for Hybrid Bayesian Networks with Full Density Estimations," in Proe. 13th International Conferenee on Information Fusion, Edinburgh, UK, July, 2010.

[13] KC Chang and Wei Sun, "Sealable Fusion with Mixture Distributions in Sensor Networks," in Proe. 11th International Conferenee on Control, Automation, Robotics and Vision, Singapore, Dee. 2010.

[14] Wei Sun and KC Chang, "Sealable Inferenee for Hybrid Bayesian Networks using Direet Message Passing," submitted to IEEE Trans. on AES, 2011.

[15] Vikas Katori and KC Chang, "Distributed Fusion for Underwater Mine Deteetion and Classifieation," in SPIE Defense and Seeurity Symposium, Orlando, Florida, April, 2011.

[16] Hongda Chen, KC Chang, and Craig Agate, "Traeking with UAV using Tangent-plus-Lyapunov Veetor Field Guidanee," in Proe. 12th International Conference on Information Fusion, Seattle, July, 2009.

# Message Passing for Hybrid Bayesian Networks: Representation, Propagation, and Integration

WEI SUN
K. C. CHANG
George Mason University

The traditional message passing algorithm was originally developed by Pearl in the 1980s for computing exact inference solutions for discrete polytree Bayesian networks. When a loop is present in the network, propagating messages are not exact, but the loopy algorithm usually converges and provides good approximate solutions. However, in general hybrid Bayesian networks, the message representation and manipulation for arbitrary continuous variable and message propagation between different types of variables are still open problems. The novelty of the work presented here is to propose a framework to compute, propagate, and integrate messages for hybrid models. First, we combine unscented transformation and Pearl's message passing algorithm to deal with the arbitrary functional relationships between continuous variables in the network. For the general hybrid model, we partition the network into separate network segments by introducing the concept of interface node. We then apply different algorithms for each subnetwork. Finally we integrate the information through the channel of interface nodes and then estimate the posterior distributions for all hidden variables. The numerical experiments show that the algorithm works well for nonlinear hybrid BNs.

Manuscript received October 6, 2007; revised May 19, 2008; released for publication June 16, 2008.

IEEE Log No. T-AES/45/4/935109.

Refereeing of this contribution was handled by T. Robertazzi.

Authors' current addresses: W. Sun, Dept. of Enterprise Optimization, United Airlines, 1200 E. Algonquin Rd., Elk Grove, IL 60007; K. C. Chang, Dept. of Systems Engineering and Operations Research, George Mason University, Fairfax, VA 22030-4444, E-mail: (kchang@gmu.edu).

0018-9251/09/$26.00 © 2009 IEEE

## I. INTRODUCTION

Bayesian network (BN), also known as probability belief network, causal network, [7, 23, 24] is a graphical model for knowledge representation under uncertainty and a popular tool for probabilistic inference. It models dependence relationships between random variables involved in the problem domain by conditional probability distributions (CPDs). In the network, CPD is encoded in the directed arc linking the associated random variables. The random variables that have arcs pointing to other random variables are called parent nodes and the random variables that have incoming arcs are called children nodes. The most important property of the BN is that it fully specifies the joint distribution over all random variables by a product of all CPDs. This is because each random variable is conditional independent of its nondescendant given its parents. Factoring reduces the numbers of parameters representing the joint distribution and so saves the computations for reasoning. One of the important tasks after constructing the BN model is to conduct probabilistic inference. However, this task is NP-hard in general [8]. This is true even for the seemingly easier task of finding approximate solutions [10]. Nevertheless, for some special classes such as discrete polytree or linear Gaussian polytree networks, there exists an exact inference algorithm using message passing [24] that could be done in linear time. In the past decades, researchers have proposed a great number of inference algorithms for various BNs in the literature [12]. They can be divided into two basic groups: exact and approximate algorithms. Exact inference only works for very limited types of networks with special structure and CPDs in the model. For example, the most popular exact inference algorithm—Clique tree [20, 28], also known as junction tree or clustering algorithm [13]—only works for a discrete network or the simplest hybrid model called conditional linear Gaussian (CLG) [18]. In general, the complexity of the exact inference is exponential to the size of the largest clique[1] of the triangulated moral graph in the network. For networks with many loops or general hybrid models that have mixed continuous and discrete variables, the intractability rules out the use of the exact inference algorithms.

For probabilistic inference with hybrid models, relatively little has been developed so far. The simplest hybrid model CLG is the only hybrid model for which exact inference could be done. The state-of-the-art algorithm for exact inference in CLG is Lauritzen's algorithm [17, 19]. It computes the exact answers in the sense that the first two moments of the posterior distributions are correct, while the true distribution might be a mixture of Gaussians. In general, the

[1] A fully connected subnetwork.

hybrid model may involve arbitrary distributions and arbitrary functional relationships between continuous variables. It is well known that no exact inference is possible in this case. However, approximate methods have been proposed [6, 16] to handle different hybrid models. In recent years, researchers also proposed inference algorithms using mixture of truncated exponentials (MTE) [9, 21] to approximate arbitrary distributions in order to derive the close-form solution for inference in hybrid models.

Generally, there are three main categories of approximate inference methods for BNs: model simplification, stochastic sampling, and loopy belief propagation. Model simplification methods simplify the model to make the inference algorithm applicable. Some commonly applied simplification methods include the removal of weak dependency, discretization, and linearization. Stochastic sampling is a popular framework including a number of algorithms, such as likelihood weighting (LW) [11, 27] and the state-of-the-art importance sampling algorithm called adaptive importance sampling (AIS-BN) for discrete BNs [5]. The major issue for sampling methods is to find a good sampling distribution. The sampling algorithm could be very slow to converge or in some cases with unlikely evidence, it may not converge even with a huge sample size. In recent years, applying Pearl's message passing algorithm to the network with loops, so-called "loopy belief propagation" (LBP) [22, 29], has become very popular in the literature. Although the propagating messages are not exact, researchers found that LBP usually converges, and when it converges it provides good approximate results. Due to its simplicity of implementation and good empirical performance, we propose to extend LBP for approximate inference for hybrid model. Unfortunately, because of the differences in representation and manipulations of messages with discrete and continuous variables, there is no simple and efficient way to pass messages between them. In [30], the authors use general nonparametric form to represent messages and formulate their calculation by numerical integrations for hybrid models. The method requires extensive functional estimations, samplings, and numerical integrations, and therefore is very computational intensive.

Under the framework of a message passing algorithm, first of all, we need to find a general way to represent messages. Essentially, messages are likelihoods or probabilities. In discrete case, messages are represented and manipulated by probability vectors and conditional probability tables (CPTs) which is relatively straightforward. For continuous variables, however, it is more complicated for message representation and manipulation as they may have arbitrary distributions. In this paper, we propose to use the first two moments, mean and variance, of a probability distribution to represent the continuous message regardless of its distribution. This simplification makes message calculation and propagation efficient between continuous variables while keeping the key information of the original distributions. Furthermore, to deal with the possible arbitrary functional relationship between continuous variables, a state estimation method is needed to approximate the distribution of a random variable that has gone through nonlinear transformation. Several weighted sampling algorithms such as particle filtering [1] and Bayesian bootstrapping [2] for nonlinear state estimation were proposed in the literature. However, we prefer to use unscented transformation [14, 15] due to its computational efficiency and accuracy. Unscented transformation uses a deterministic sampling scheme and can provide good estimates of the first two moments for the continuous variable undergone nonlinear transformation. For arbitrary continuous network, this approach we called unscented message passing (UMP) works very well [25]. But in the hybrid model, message propagation between discrete and continuous variables is not straightforward due to their different formats. To deal with this issue, we propose to apply conditioning. First we partition the original hybrid BNs into separate, discrete, and continuous network segments by conditioning on discrete parents of continuous variables [26]. We can then process message passing separately for each network segment before final integration.

One of the benefits of partitioning networks is to ensure that there is at least one efficient inference method applicable to each network segment. In hybrid networks, we assume that a continuous node is not allowed to have any discrete child node. Therefore, the original networks can be partitioned into separate parts by the discrete parents of continuous variables. We call these nodes the interface nodes. Each network segment separated by the interface nodes consists of purely discrete or continuous variables. By conditioning on interface nodes, the variables in different network segments are independent of each other. We then conduct loopy propagation separately in each subnetwork. Finally, messages computed in different segments are integrated through the interface nodes. We then estimate the posterior distribution of every hidden variable given evidence in all network segments.

The algorithm proposed in this paper aims to tackle nonlinear hybrid models. We believe that the proposed combination of known efficient methods and the introduction of interface nodes for hybrid network partition makes the new algorithm a good alternative for inference in nonlinear hybrid models. The remainder of this paper is organized as follows. Section II first reviews Pearl's message passing formulae. We then discuss the message representation and manipulation for continuous variable and how to propagate messages between continuous variables with nonlinear functional relationship. Section III describes the methods of network partition and message integration by introducing the concept of interface nodes. We show how message passing can be done separately and finally integrated together via the channel of interface nodes. Section IV presents the algorithm of hybrid message passing by conditioning. Several numerical experiments are presented in Section V. Finally, Section VI concludes the research we have done in this paper and suggests some potential future work.

## II. MESSAGE PASSING: REPRESENTATION AND PROPAGATION

Pearl's message passing algorithm [24] is the first exact inference algorithm developed originally for polytree discrete BNs. Applying Pearl's algorithm in the network with loops usually provides approximate answers, and this method is called LBP. Recall that in Pearl's message passing algorithm, $\mathbf{e}_X^+$ and $\mathbf{e}_X^-$ are defined as the evidence from the subnetwork "above" a node $X$ and the subnetwork "below" $X$, respectively. In a polytree, any node $X$ d-separates the set of evidence $\mathbf{e}$ into $\{\mathbf{e}_X^+, \mathbf{e}_X^-\}$. In the algorithm, each node in the network maintains two values called $\lambda$ value and $\pi$ value. $\lambda$ value of a node $X$, defined as

$$\lambda(X) = P(\mathbf{e}_X^- \mid X) \tag{1}$$

is the likelihood of observations $\mathbf{e}_X^-$ given $X$. $\pi$ value of a node $X$, defined as

$$\pi(X) = P(X \mid \mathbf{e}_X^+) \tag{2}$$

is the conditional probability of $X$ given $\mathbf{e}_X^+$.

The belief of a node $X$ given all evidence is the normalized product of $\pi$ value and $\lambda$ value. Each node, after updating its own belief, sends new $\lambda$ message to its parents and new $\pi$ message to its children. For a typical node $X$ with $m$ parents $\mathbf{T}(T_1, T_2, \ldots, T_m)$ and $n$ children $\mathbf{Y}(Y_1, Y_2, \ldots, Y_n)$ as illustrated in Fig. 1, the conventional propagation equations of Pearl's message passing algorithm can be expressed as the following [24]:

$$\mathrm{BEL}(X) = \alpha\pi(X)\lambda(X) \tag{3}$$

$$\lambda(X) = \prod_{j=1}^{n} \lambda_{Y_j}(X) \tag{4}$$

$$\pi(X) = \sum_{\mathbf{T}} P(X \mid \mathbf{T}) \prod_{i=1}^{m} \pi_X(T_i) \tag{5}$$

$$\lambda_X(T_i) = \sum_{X} \lambda(X) \sum_{T_k : k \neq i} P(X \mid \mathbf{T}) \prod_{k \neq i} \pi_X(T_k) \tag{6}$$

$$\pi_{Y_j}(X) = \alpha \left[ \prod_{k \neq j} \lambda_{Y_k}(X) \right] \pi(X) \tag{7}$$

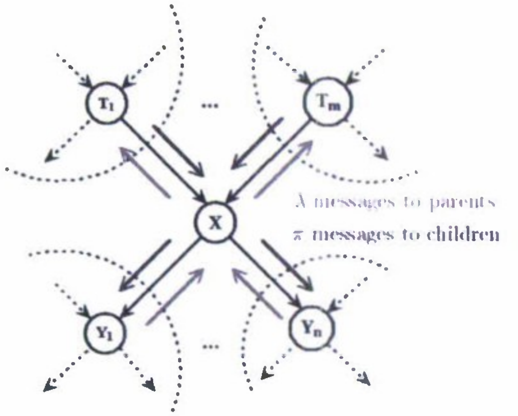


Fig. 1. Typical node $X$ with $m$ parents and $n$ children.

where $\lambda_{Y_j}(X)$ is the $\lambda$ message node $X$ receives from its child $Y_j$, $\lambda_X(T_i)$ is the $\lambda$ message $X$ sends to its parent $T_i$; $\pi_X(T_i)$ is the $\pi$ message node $X$ receives from its parent $T_i$, $\pi_{Y_j}(X)$ is the $\pi$ message $X$ sends to its child $Y_j$; and $\alpha$ is a normalizing constant.

When this algorithm is applied to a polytree network, the messages propagated are exact and so are the beliefs of all nodes after receiving all messages. For the network with loops, we can still apply this algorithm as the "loopy propagation" mentioned above. In general, loopy propagation will not provide the exact solutions. But empirical investigations on its performance have reported surprisingly good results.

For discrete variables, messages could be represented by probability vectors, and the conditional probability table of node $X$ given its parent $T$, $P(X \mid \mathbf{T})$, could be represented by a matrix. Therefore the calculations in the above formulae are the product of vectors and multiplication of vector and matrices, which can be carried out easily. However, for continuous variables, message representation and the corresponding calculations are much more complicated. First, an integral replaces summation in the above equations. Furthermore, since continuous variable could have arbitrary distribution over the continuous space, in general it is very difficult to obtain exact close-form analytical results when combining multiple continuous distributions. In order to make the computations feasible while keeping the key information, we use the first two moments, mean and variance, to represent continuous message regardless of the original distribution. Then, the product of different continuous distributions could be approximated with a Gaussian distribution. Note that for the continuous case, $P(X \mid \mathbf{T})$ is a continuous conditional distribution, and it may involve an arbitrary function between continuous variables. To integrate the product of continuous distributions as shown in (5) and (6), it has to take into account the functional transformation of continuous variables. Fortunately, unscented transformation [14, 15] provides good estimates of mean and variance for the continuous variables through nonlinear transformation. In our algorithm, unscented transformation plays a key role for computing continuous messages. Specifically, we use it to formulate and compute the $\pi$ and $\lambda$ messages since both computations involve the conditional probability distribution in which nonlinear transformation may be required.

### A. Unscented Transformation

Proposed in 1996 by Julier and Uhlmann [15], unscented transformation is a deterministic sampling method to estimate mean and variance of continuous random variable that has undergone nonlinear transformation. Consider the following problem: a continuous random variable $\mathbf{x}$ with mean $\bar{\mathbf{x}}$ and covariance matrix $\mathbf{P}_\mathbf{x}$ undergoes an arbitrary nonlinear transformation, written as $\mathbf{y} = g(\mathbf{x})$; the question is how to compute the mean and covariance of $\mathbf{y}$?

From probability theory, we have

$$p(\mathbf{y}) = \int_{\mathbf{x}} p(\mathbf{y} \mid \mathbf{x}) p(\mathbf{x}) dx.$$

However, in general the above integral may be difficult to compute analytically and may not always have a close-form solution. Therefore, instead of finding the distribution, we retreat to seek for its mean and covariance. Based on the principle that it is easier to approximate a probability distribution than an arbitrary nonlinear function, unscented transformation uses a minimal set of deterministically chosen sample points called sigma points to capture the true mean and covariance of the prior distribution. Those sigma points are propagated through the original functional transformation individually. According to its formulae, posterior mean and covariance calculated from these propagated sigma points are accurate to the 2nd order for any nonlinearity. In the special case when the transformation function is linear, the posterior mean and variance are exact.

The original unscented transformation encounters difficulties with high-dimensional variables, so the scaled unscented transformation was developed soon afterward [14]. The scaled unscented transformation is a generalization of the original unscented transformation. We will use the two terms interchangeably, but both mean scaled unscented transformation in the remainder of this paper.

Now let us describe the formulae of unscented transformation. Assume $\mathbf{x}$ is $L$-dimensional multivariate random variable. First, a set of $2L + 1$ sigma points are specified by the following formulae:

$$\lambda = \alpha^2 (L + \kappa) - L$$

$$\mathcal{X} = \begin{cases} \mathcal{X}_0 = \bar{\mathbf{x}} & i = 0 \\ \mathcal{X}_i = \bar{\mathbf{x}} + \left(\sqrt{(L+\lambda)\mathbf{P}_\mathbf{x}}\right)_i & i = 1,\ldots,L \\ \mathcal{X}_i = \bar{\mathbf{x}} - \left(\sqrt{(L+\lambda)\mathbf{P}_\mathbf{x}}\right)_i & i = L+1,\ldots,2L \end{cases} \tag{8}$$

and the associated weights for these $2L + 1$ sigma points are

$$\begin{aligned} w_0^{(m)} &= \frac{\lambda}{L+\lambda} & i &= 0 \\ w_0^{(c)} &= \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) & i &= 0 \\ w_0^{(m)} = w_0^{(c)} &= \frac{1}{2(L+\lambda)} & i &= 1,\ldots,2L \end{aligned} \tag{9}$$

where $\alpha$, $\beta$, $\kappa$ are scaling parameters and the superscripts "$(m)$," "$(c)$" indicate the weights for computing posterior mean and covariance, respectively. The values of scaling parameters could be chosen by $0 \le \alpha \le 1$, $\beta \ge 0$, and $\kappa \ge 0$. It has been shown empirically that the specific values chosen for the parameters are not critical because unscented transformation is not sensitive to those parameters. We choose $\alpha = 0.8$, $\beta = 2$ (optimal for Gaussian prior [14]), and $\kappa = 0$ in all of our experiments.

After the sigma points are selected, they are propagated through the functional transformation:

$$\mathcal{Y}_i = \mathbf{g}(\mathcal{X}_i) \qquad i = 0,\ldots,2L. \tag{10}$$

Finally, the posterior mean and covariance are estimated by combining the propagated sigma points as follows:

$$\bar{\mathbf{y}} \approx \sum_{i=0}^{2L} w_i^{(m)} \mathcal{Y}_i \tag{11}$$

$$\mathbf{P}_\mathbf{y} \approx \sum_{i=0}^{2L} w_i^{(c)} (\mathcal{Y}_i - \bar{\mathbf{y}})(\mathcal{Y}_i - \bar{\mathbf{y}})^\mathrm{T} \tag{12}$$

$$\mathbf{P}_{\mathbf{xy}} \approx \sum_{i=0}^{2L} w_i^{(c)} (\mathcal{X}_i - \bar{\mathbf{x}})(\mathcal{Y}_i - \bar{\mathbf{y}})^\mathrm{T}. \tag{13}$$

In short, we denote the unscented transformation for $X$ undergoing a functional transformation $Y = f(X)$ as the following:

$$(Y.\mathrm{mu}, Y.\mathrm{cov}) = UT\left(X \xrightarrow{f(X)} Y\right). \tag{14}$$

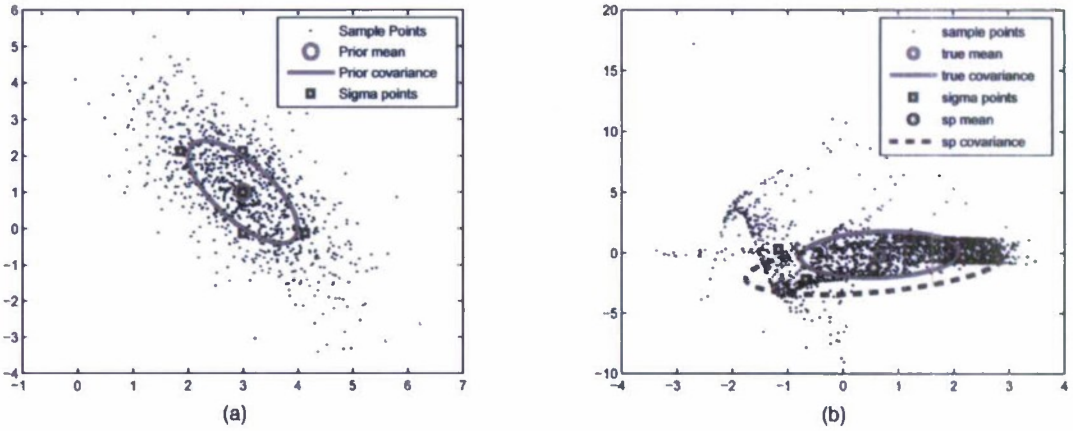
We demonstrate the unscented transformation by a simple two-dimension Gaussian example. Let $\mathbf{x} = [x_1 \ x_2]$ with mean and covariance matrix given as

$$\bar{\mathbf{x}} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \qquad \mathbf{P}_\mathbf{x} = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}.$$

In order to show the robustness of unscented transformation, we choose a set of functions with severe nonlinearity shown below:

$$y_1 = \log(x_1^2)\cos(x_2), \qquad y_2 = \sqrt{\exp(x_2)}\sin(x_1 x_2).$$

The true posterior statistics are approximated very closely by brute force Monte Carlo simulation



(a)

(b)

Fig. 2. Demonstration of unscented transformation. (a) Prior distribution. (b) After nonlinear transformation.

using 100,000 sample points drawn from the prior distribution and then propagated through the nonlinear mapping. We compare them with the estimates calculated by unscented transformation using only 5 sigma points. Fig. 2 shows that the mean calculated by transformed sigma points is very close to the true mean and that the posterior covariance seems consistent and efficient because the sigma-point covariance ellipse is larger but still tight around the true posterior covariance ellipse.

### B. Unscented Message Passing

Now let us take a closer look at Pearl's general message propagation formulae shown in (3)–(7). In recursive Bayesian inference, $\pi$ message represents prior information and $\lambda$ message represents evidential support in the form of a likelihood function. Equations (3), (4), and (7) are essentially the combination of different messages by multiplication. They are similar to the data fusion concept where estimates received from multiple sources are combined.

Under the assumption of Gaussian distribution, the fusion formula is relatively straightforward [3]. Specifically, (3), (4), and (7) can be rewritten in terms of the first two moments of the probability distributions as the following:

$$\text{BEL}(X)\begin{cases} \text{cov} = \left(\dfrac{1}{\pi(X).\text{cov}} + \dfrac{1}{\lambda(X).\text{cov}}\right)^{-1} \\ \text{mu} = \text{cov}\left[\dfrac{\pi(X).\text{mu}}{\pi(X).\text{cov}} + \dfrac{\lambda(X).\text{mu}}{\lambda(X).\text{cov}}\right] \end{cases} \qquad (15)$$

$$\lambda(X)\begin{cases} \text{cov} = \left(\displaystyle\sum_{j=1}^{n}\dfrac{1}{\lambda_{Y_j}(X).\text{cov}}\right)^{-1} \\ \text{mu} = \text{cov}\left[\displaystyle\sum_{j=1}^{n}\dfrac{\lambda_{Y_j}(X).\text{mu}}{\lambda_{Y_j}(X).\text{cov}}\right] \end{cases} \qquad (16)$$

$$\pi_{Y_j}(X)\begin{cases} \text{cov} = \left(\dfrac{1}{\pi(X).\text{cov}} + \displaystyle\sum_{k\neq j}\dfrac{1}{\lambda_{Y_k}(X).\text{cov}}\right)^{-1} \\ \text{mu} = \text{cov}\left[\dfrac{\pi(X).\text{mu}}{\pi(X).\text{cov}} + \displaystyle\sum_{k\neq j}\dfrac{\lambda_{Y_k}(X).\text{mu}}{\lambda_{Y_k}(X).\text{cov}}\right] \end{cases} \qquad (17)$$

where mu, cov stand for corresponding mean and covariance, respectively.

Equation (5) computes the $\pi$ value for node $X$. Analytically, this is equivalent to treating $X$ as a functional transformation of $\mathbf{T}$ and the function is the one defined in CPD of $X$ denoted as $h(X)$. Technically, we take $\mathbf{T}$ as a multivariate random variable with a mean vector and a covariance matrix; then by using unscented transformation, we obtain an estimate of mean and variance of $X$ to serve as the $\pi$ value for node $X$. In (5), $\pi_X(T_i)$ is the $\pi$ messages sending to $X$ from its parent $T_i$, which is also represented by mean and covariance. By combining all the incoming $\pi_X(T_i)$ messages, we can estimate the mean vector and covariance matrix of $\mathbf{T}$. Obviously, the simplest way is to view all parents as independent variables; then combine their means into a mean vector, and place their variances at the diagonal positions to form a diagonal covariance matrix.[2] With that, we can compute the $\pi$ value of node $X$ by

$$(\pi(X).\text{mu}, \pi(X).\text{cov}) = UT\left(\mathbf{T}\xrightarrow{h(X)}X\right). \qquad (18)$$

Similarly but a bit more complicated, (6) computes the $\lambda$ message sending to its parent $(T_i)$ from node $X$. Note here that we integrate out $X$ and all of its parents except the one $(T_i)$ we are sending $\lambda$ message to. Theoretically, this is equivalent to regarding $T_i$ as the functional transformation of $X$ and $\mathbf{T}\backslash T_i$. It

[2] This is actually how the original loopy algorithm works and why it is not exact. To improve the algorithm, we can estimate the correlations between all parents and include them in the covariance matrix of $\mathbf{T}$.

is necessary to mention that the function used for transformation is the inverse function of the original one specified in $P(X \mid \mathbf{T})$ with $T_i$ as the independent variable. We denote this inverse function as $v(X, \mathbf{T}\backslash T_i)$. Note that in practical problems, the original function may not be invertible, or its inverse function may not be unique. In such a case, we need additional steps to apply the method. In this paper, we assume the inverse function is unique and always available. To compute the message, we first augment $X$ with $\mathbf{T}\backslash T_i$ to obtain a new multivariate random variable called **TX**; then the mean vector and covariance matrix of **TX** are estimated by combining $\lambda(X)$ and $\pi_X(T_k)(k \neq i)$. After applying unscented transformation to **TX** with the new inverse function $v(X, \mathbf{T}\backslash T_i)$, we obtain an estimate of the mean and variance for $T_i$ serving as the $\lambda_X(T_i)$ message as below:

$$(\lambda_X(T_i).\mathrm{mu}, \lambda_X(T_i).\mathrm{cov}) = UT\left(\mathbf{TX} \xrightarrow{v(X,\mathbf{T}\backslash T_i)} T_i\right). \quad (19)$$

With (15)–(19), we can now compute all messages for continuous variables. As one may notice, unscented transformation plays a key role here. This is why we call it UMP for continuous BNs.

So far, we have summarized message representation and propagation for discrete and continuous variables, respectively. However, for the hybrid model, we have to deal with the messages passing between both types of variables. Since they are in different formats, messages cannot be integrated directly. As mentioned in Section I, our approach is to partition the original network before propagating messages between them.

## III. NETWORK PARTITION AND MESSAGE INTEGRATION FOR HYBRID MODEL

First of all, as mentioned earlier, we assume that a discrete node can only have discrete parents in the hybrid models, which implies continuous variable cannot have any discrete child node.

DEFINITION 1 In a hybrid BN, a discrete variable is called a discrete parent if and only if it has at least one continuous child node.

It is well known that BN has an important property that every node is independent of its nondescendant nodes given its parents. Therefore the following theorem follows.

THEOREM 1 *All discrete parents in the hybrid BN model can partition the network into independent network segments, each having either purely discrete or purely continuous variables. We call the set of all discrete parents in the hybrid network the interface nodes. In other words, the interface nodes "d-separate" the network into different network segments.*

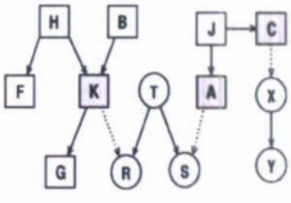


Fig. 3. Demonstration of interface nodes and network partition.

It is obvious that the variables in different segments of the network are independent of each other given the interface nodes. An example is shown in Fig. 3 where a 13-node hybrid model is presented. Following the convention, we use a square or rectangle to depict the discrete variable and a circle or ellipse to depict the continuous variable. As can be seen, $K$, $A$, and $C$ are the interface nodes in this example. By representing the arcs between discrete parents and their continuous children as dot lines, four independent network segments are formulated—two discrete parts ($H$, $B$, $F$, $K$, $G$ and $J$, $A$, $C$) and two continuous parts ($T$, $R$, $S$ and $X$, $Y$).

After partitioning the network with the interface nodes, we choose the most appropriate inference algorithm for each network segment. In fact, we can also combine some segments together if the same algorithm works for all of them. The purpose of introducing the interface nodes is to facilitate the network partition so that at least one algorithm could be applicable to each segment. In general, separate message passing in either discrete or continuous network segment is always doable. Typically, the continuous network segment with nonlinear and/or non-Gaussian CPDs is the most difficult one to deal with. In such case, we apply UMP presented in Section IIB for approximate solutions.

Finally, we need to summarize the prior and evidence information for each network segment and encode it as messages to be passed between network segments through the interface nodes. This is similar to general message passing but requires message integrations between different network segments.

### A. Message Integration for Hybrid Model

For a hybrid model, without loss of generality, let us assume that the network is partitioned into two parts denoted as $\mathcal{D}$ and $\mathcal{C}$. Part $\mathcal{D}$ is a discrete network and it is solvable by appropriate algorithms such as junction tree or discrete loopy propagation. Part $\mathcal{C}$ is an arbitrary continuous network. Let us denote the observable evidence in part $\mathcal{D}$ as $E_d$, and the evidence from $\mathcal{C}$ as $E_c$. Therefore the entire evidence set $\mathbf{E}$ consists of $E_d$ and $E_c$. As mentioned before, given interface nodes, variables from the two network segments are conditional independent of each other. The evidence from part $\mathcal{D}$ affects the posterior

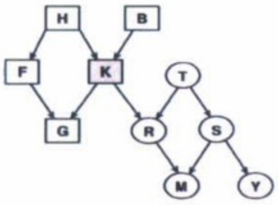


Fig. 4. Synthetic hybrid Bayesian networks-1.

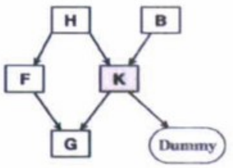


Fig. 5. Transformed model with dummy node.

probability of hidden nodes in part $C$ and vice versa only through the channel of the interface nodes.

We therefore summarize the prior and evidence information of each network segment and encode them as either $\pi$ or $\lambda$ value at the interface nodes. Assuming that the set of interface nodes between two network segments is $\mathbf{I}$, then the two messages are: $\lambda(\mathbf{I}) = P(E_c \mid \mathbf{I})$ and $\pi(\mathbf{I}) = P(\mathbf{I} \mid E_d)$. These values are to be passed between network segments to facilitate information integration. As in Pearl's algorithm, this approach can be easily integrated with the UMP-BN loopy algorithm mentioned above in a unified manner.

We use the following concrete example to illustrate how to integrate messages from different network segments. As can be seen in Fig. 4, synthetic hybrid model-1 has $K$ as the interface node dividing the network into a discrete part consisting of $H$, $B$, $F$, $K$, $G$ and a continuous part consisting of $T$, $R$, $S$, $M$, $Y$. For the purpose of illustration, let us assume all discrete nodes are binary and all continuous nodes are scalar Gaussian variables.

Suppose the leaf nodes $G$, $M$, $Y$ are observable evidence. We first focus on the continuous segment. In this step, we compute the $\lambda$ message sending to the interface node $K$ from continuous evidence. And conditioning on each possible state of $K$, we estimate the posterior distributions for all hidden continuous variables given continuous evidence. Under Gaussian assumption, these posterior distributions are represented by means and variances and they are intermediate results that will be combined after we obtain the a posterior probability distribution of the interface node $K$ given all evidence. Probabilities of all possible states of $K$ are served as the mixing weights, similar to computing the mean and variance of a Gaussian mixture.

Given $K$, it is straightforward to compute the likelihood of continuous evidence $M = m$, $Y = y$ because we can easily estimate the conditional probability distribution of evidence node given interface nodes and other observations. For example, let

$$P(M = m, Y = y \mid K = 1) = a$$

$$P(M = m, Y = y \mid K = 2) = b.$$

Then to incorporate the evidence likelihood is equivalent to adding a binary discrete dummy node as the child of the interface node $K$ with the conditional probability table shown as the following:

|  | Dummy |  |
|---|---|---|
| $K$ | 1 | 2 |
| 1 | $\alpha a$ | 1-$\alpha a$ |
| 2 | $\alpha b$ | 1-$\alpha b$ |

where $\alpha$ is a normalizing constant.

By setting "Dummy" to be observed as state 1, the entire continuous segment could be replaced by the node Dummy. Then the original hybrid BN can be transformed into a purely discrete model shown in Fig. 5 in which Dummy integrates all of the continuous evidence information.

The second step is to compute the posterior distributions for all hidden discrete nodes given $G = g$, Dummy = 1. We have several algorithms to choose for inference depending on the complexity of the transformed model. In general, we can always apply discrete loopy propagation algorithm to obtain approximate results regardless of network topology. Note that the posterior distributions of the discrete nodes have taken into account all evidence including the ones from continuous segment via the Dummy node. However, we need to send the updated information back to the continuous subnetwork via the set of interface nodes. This is done by computing the joint posterior probability distribution of the interface nodes denoted as $P(\mathbf{I} \mid \mathrm{E})$. Essentially, it is the $\pi$ messages to be sent to the continuous network segment.

With the messages encoded in the interface nodes, the last step is to go back to the continuous segment to compute the a posterior probability distributions for all hidden continuous variables. Recall that in the first step, for any hidden continuous variable $X$, we already have $P(X \mid \mathbf{I}, E_c)$ computed and saved. The following derivation shows how to compute $P(X \mid \mathrm{E})$:

$$\begin{aligned} P(X \mid \mathrm{E}) &= P(X \mid E_c, E_d) \\ &= \sum_{\mathbf{I}} P(X, \mathbf{I} \mid E_c, E_d) \\ &= \sum_{\mathbf{I}} P(X \mid \mathbf{I}, E_c, E_d) P(\mathbf{I} \mid E_c, E_d) \\ &= \sum_{\mathbf{I}} P(X \mid \mathbf{I}, E_c) P(\mathbf{I} \mid \mathrm{E}). \end{aligned} \tag{20}$$

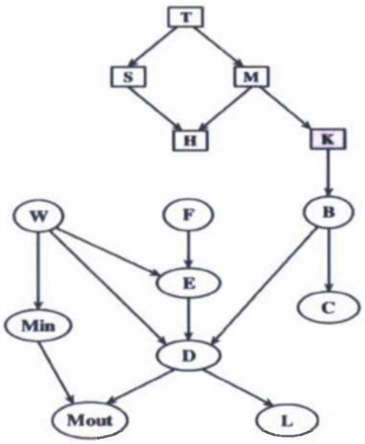


Fig. 6. GHM-2.

The fourth equality is due to the fact that the set of interface node d-separate the node $X$ with $E_d$.

Assuming given an instantiation of the set of interface nodes $\mathbf{I} = i$, $P(X \mid \mathbf{I} = i, E_c)$ is a Gaussian distribution with mean $\bar{x}_i$ and variance $\sigma_i^2$. Then (20) is equivalent to computing the probability density function of a Gaussian mixture with $P(\mathbf{I} = i \mid \mathbf{E})$ as the weighting factors. Denoting $P(\mathbf{I} = i \mid \mathbf{E})$ as $p_i$, the mean $\bar{x}$ and the variance $\sigma_x^2$ of $P(X \mid \mathbf{E})$ can be computed as the following [3, p. 56]:

$$\bar{x} = \sum_i p_i \bar{x}_i \tag{21}$$

$$\sigma_x^2 = \sum_i p_i \sigma_i^2 + \sum_i p_i \bar{x}_i^2 - \bar{x}^2. \tag{22}$$

Through the above three steps, we successfully integrate messages from different subnetworks to obtain the approximate posterior marginal distribution for both continuous and discrete hidden variables given all evidence. There are two approximations in the algorithm. One is from loopy propagation method itself. Another one is that we approximate continuous variable as Gaussian distributed as we only use the first two moments to represent continuous messages. However, it provides promising performance as seen in the numerical experiment results.

## IV. HYBRID MESSAGE PASSING ALGORITHM

We have presented separate message passing in either discrete or continuous network segment and message integration in hybrid model via interface nodes. In this section, we summarize the general algorithm of message passing for hybrid BNs as shown in Table I.

TABLE I
Hybrid Message Passing Algorithm for General Mixed BN

**Algorithm:** Hybrid Message Passing for General Mixed BN (HMP-BN).

**Input:** General hybrid BN given a set of evidence.

**Output:** Posterior marginal distributions of all hidden nodes.

1. Determine the interface nodes and partition the network into independent segments with interface nodes. Choose the appropriate inference algorithm for each network segment.
2. Continuous network segment: compute the $\lambda$ message sending to the interface nodes and the intermediate posterior distribution of the hidden continuous variables given the interface nodes and the local evidence.
3. Transform the original network into an equivalent discrete model with a dummy node added as a child of the interface nodes. This dummy discrete node carries the $\lambda$ message from continuous evidence to the interface nodes.
4. Compute the posterior distribution for every hidden discrete variable using the transformed discrete model. The joint posterior probability table of the interface nodes is saved as the $\pi$ message to be sent back to the continuous network segment.
5. Compute the posterior distribution for every hidden continuous variable given all evidence by integrating the $\pi$ message using (20).

In order to incorporate evidence information, we allow a node to send a $\lambda$ message to itself. For a discrete network, we initialize the messages by letting all evidence nodes send to themselves a vector of a "1" for observed state and 0s for other states. All other messages are initialized as vectors of 1s. For continuous network, a message is represented by mean and variance. We initialize the messages for all continuous evidence nodes, sending themselves as the one with the mean equal to the observed value and the variance equal to zero. All other messages in continuous network are initialized as uniform, specifically, zero-mean and infinity variance (the so-called "diffusion prior"). Then in each iteration, every node computes its own belief and outgoing messages based on the incoming messages from its neighbors. We assess the convergence by checking if any belief change is less than a prespecified threshold (for example, $10^{-4}$). We use parallel updating for each node until the messages are converged.

## V. NUMERICAL EVALUATION

### A. Experiment Method

We use two synthetic hybrid models for experiments. One is shown in Fig. 4 as mentioned in Section IIIA called GHM-1. GHM-1 has one loop in each network segment, respectively, (partitioned by the interface node $K$). Another experiment model is shown in Fig. 6 called GHM-2. GHM-2 has multiple loops in the continuous segment.

For GHM-1, we assume that the leaf nodes $G, M, Y$ are observable evidence. We model its continuous segment as a linear Gaussian network given the interface node $K$. Therefore the original network is a

CLG so that the exact inference algorithm (junction tree) can be used to provide the true answer as a golden standard for performance comparison. The CPTs and CPDs for nodes in GHM-1 are randomly specified.

Note that our algorithm can handle general arbitrary hybrid model, not just CLG. GHM-2 is designed specifically to test the algorithm under the situation where nonlinear CPDs are involved in the model. The structure of the continuous segment in GHM-2 is borrowed from [17] in which the author proposed junction tree algorithm for CLG. The discrete nodes in the GHM-2 are binary, and we randomly specify the CPTs for them similar to the one in GHM-1. But the CPDs for the continuous nodes are deliberately specified using severe nonlinear functions shown below to test the robustness of the algorithm:

$$\mathbf{F} \sim \mathcal{N}(-10,3)$$

$$\mathbf{W} \sim \mathcal{N}(100,10)$$

$$\mathbf{B} \mid \mathbf{K} = \mathbf{1} \sim \mathcal{N}(50,5)$$

$$\mathbf{B} \mid \mathbf{K} = \mathbf{2} \sim \mathcal{N}(60,5)$$

$$\mathbf{E} \sim \mathcal{N}(W + 2F,1)$$

$$\mathbf{C} \sim \mathcal{N}(e^{\sqrt[3]{B}},3)$$

$$\mathbf{D} \sim \mathcal{N}(\sqrt{W} \times \log(E) - B,5)$$

$$\mathbf{Min} \sim \mathcal{N}(\sqrt{W} + 6,3)$$

$$\mathbf{Mout} \sim \mathcal{N}(0.5 \times D \times \mathrm{Min},5)$$

$$\mathbf{L} \sim \mathcal{N}(-5 \times D,5).$$

We assume that the evidence set in the GHM-2 is $\{H,C,\mathrm{Mout},L\}$. Since no exact algorithm is available for such model, for comparison purposes, we use the brute force sampling method, likelihood weighting, to obtain an approximate true solution with a large number of samples (20 million samples).

In our experiments, we first randomly sample the network and clamp the evidence nodes by their sampled value. Then we run HMP-BN to compute the posterior distributions for the hidden nodes. It is important to mention that in both discrete and continuous network segments, we implement HMP-BN using loopy algorithms to make it general, although junction tree could be used in network segment whenever it is applicable. In addition, we run LW using as many samples as it can generate within roughly the same amount of time HMP-BN consumes. There are 10 random runs for GHM-1 and 5 random runs for GHM-2. We compare the average Kullback-Leibler (KL) divergences of the posterior distributions obtained by different algorithms.

Given unlikely evidence, it is well known that the sampling methods converge very slowly even with a large sample size. We use GHM-1 to test the robustness of our algorithm in this case because junction tree can provide the ground true for GHM-1 regardless of the evidence likelihood. We generate 10 random cases with evidence likelihood between $10^{-5} \sim 10^{-15}$ and run both HMP-BN and LW to compare the performances.

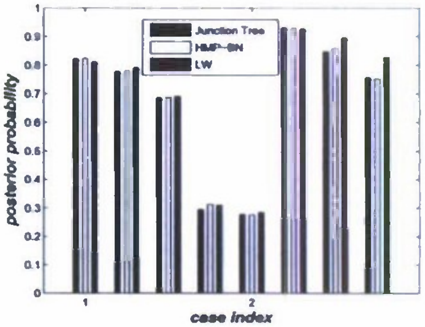


Fig. 7. Posterior probability of hidden discrete variables in two typical runs.

### B. Experiment Results

For model GHM-1, there are 4 hidden discrete nodes and 3 hidden continuous nodes. Fig. 7 illustrates the posterior probabilities of hidden discrete nodes computed by junction tree, HMP-BN, and LW in two typical runs. Since GHM-1 is a simple model and we did not use unlikely evidence, both HMP-BN and LW perform well.

For continuous variables in GHM-1, Fig. 8 shows the performance comparisons in means and variances of the posterior distributions for the hidden continuous nodes in all of the 10 runs. The normalized error is defined as the ratio of the absolute error over the corresponding true value. From the figure, it is evident that HMP-BN provides accurate estimates of means, while the estimated variances deviate from the true somewhat but HMP-BN is still better than LW in most cases.

We then demonstrate the robustness of HMP-BN by testing its performance given unlikely evidence shown in Fig. 9. In this experiment, 10 random sets of evidence are chosen with likelihood between $10^{-5}$ and $10^{-15}$. As can be seen, HMP-BN performs significantly better than LW in this case. The average KL divergences are consistently small with the maximum value less than 0.05. This is not surprising because LW uses the prior to generate samples so that it hardly hits the area close to the observations.

We summarize the performance results with GHM-1 in Table II. Note that given unlikely evidence, the average KL divergence by HMP-BN is more than one order of magnitude better than LW.

In GHM-2, due to the nonlinear nature of the model, no exact method exists to provide the benchmark. We use LW with 20 million samples to obtain an approximation of the true value. We implemented five simulation runs with randomly

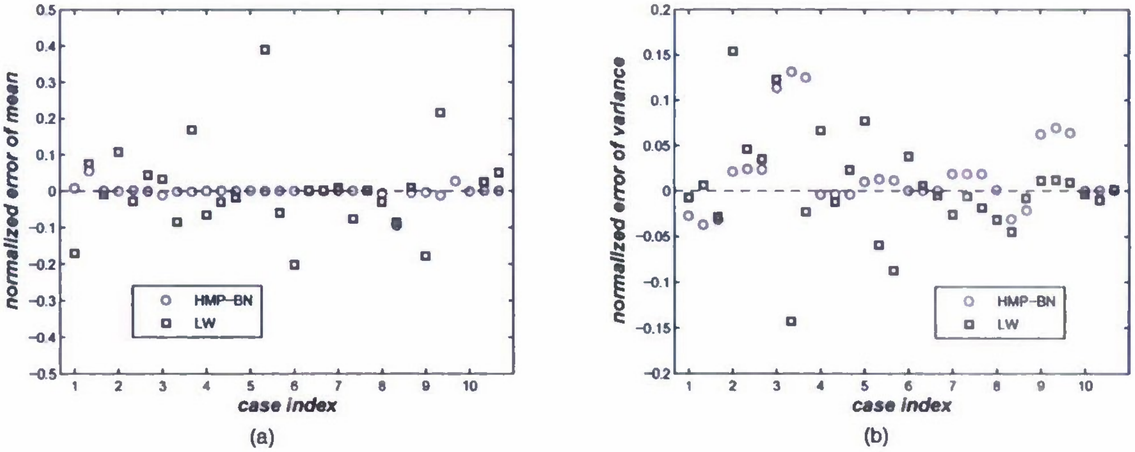


Fig. 8. GHM-1 Performance comparison for 10 random runs (ground true is provided by junction tree). (a) Mean comparison. (b) Variance comparison.

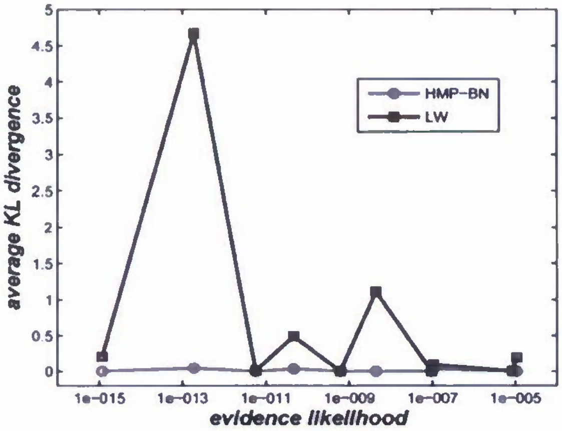


Fig. 9. GHM-1: Performance comparison given unlikely evidence.

sampled evidence. In this experiment, we adopt our newly developed algorithm UMP-BN for inference in continuous network segment [25]. Fig. 10 shows the performance comparison in means and variances of the posterior distribution for the hidden continuous variables. Also, Table III summarizes the average KL divergences in testing GHM-2. From the data, we see that HMP-BN combining with UMP-BN applied in the continuous subnetwork produces very good results. In this nonlinear model with the normal evidence, the new algorithm performs much better than LW despite its advantages of being a model-free algorithm. However, since there is only one interface node in these models, implementing HMP-BN is relatively simple.

### C. Complexity of HMP-BN

In general, when there are multiple interface nodes, HMP-BN computes the posterior distributions of hidden continuous variables given continuous evidence, conditioned on every combination of

TABLE II
Average KL-Divergence Comparison in Testing GHM-1

| Average<br>KL divergence | Normal Evidence<br>$> 10^{-5}$ | Unlikely Evidence<br>$10^{-5}$–$10^{-15}$ |
|---|---|---|
| HMP-BN | 0.0011 | 0.0108 |
| LW | 0.0052 | 0.67 |

TABLE III
Average KL-Divergence Comparison in Testing GHM-2

| | Average KL Divergence |
|---|---|
| HMP-BN | 0.0056 |
| LW | 0.0639 |

instantiations of all interface nodes. So the complexity of the algorithm is highly dependent on the size of interface nodes. To assess the complexity of HMP-BN, we conducted a random experiment using network structure borrowed from the ALARM model [4] as shown in Fig. 11 in which there are 37 nodes. We randomly selected each node to be discrete or continuous with only a requirement that continuous variable cannot have any discrete child node. In this experiment, the average number of interface nodes was about 12. HMP-BN still provided good estimates of the posterior distributions but it took a much longer time than the one with only one interface node. If we have $n$ interface nodes $K_1, K_2, \ldots, K_n$ with number of states $n_1, n_2, \ldots, n_k$, respectively, the computational complexity of HMP-BN is proportional to $\mathcal{O}(n_1 \times n_2 \times n_3 \cdots \times n_k)$. This implies that our algorithm is not scalable for a large number of interface nodes. However, our goal is not to propose an algorithm for all models (NP-hard in general) and we suspect that it is rare to have a large number of interface nodes in most practical models. Even with the considerable size of interface nodes, HMP-BN provides good results within a reasonable time while the stochastic sampling

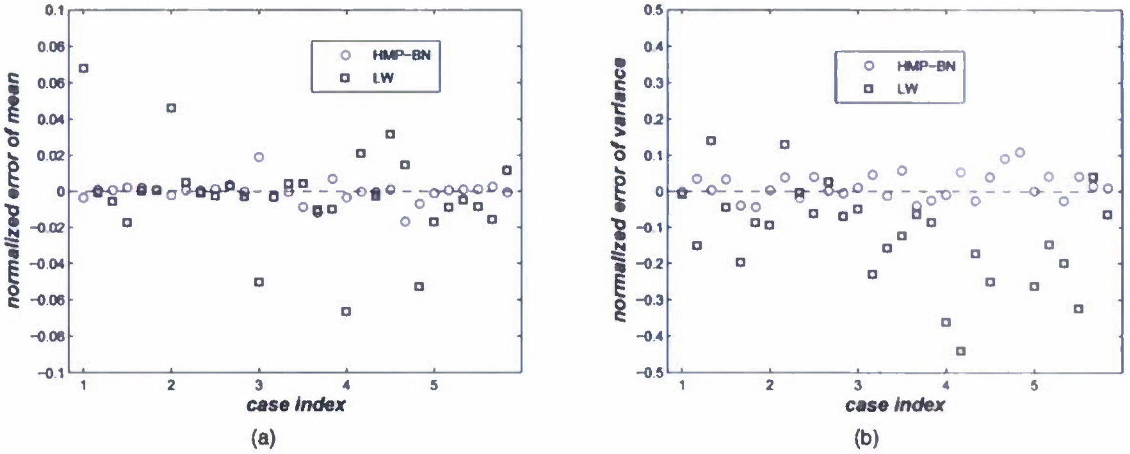


Fig. 10. GHM-2: Performance comparison for 5 random runs (the reference base is provided by LW with 20 millions samples). (a) Mean comparison. (b) Variance comparison.

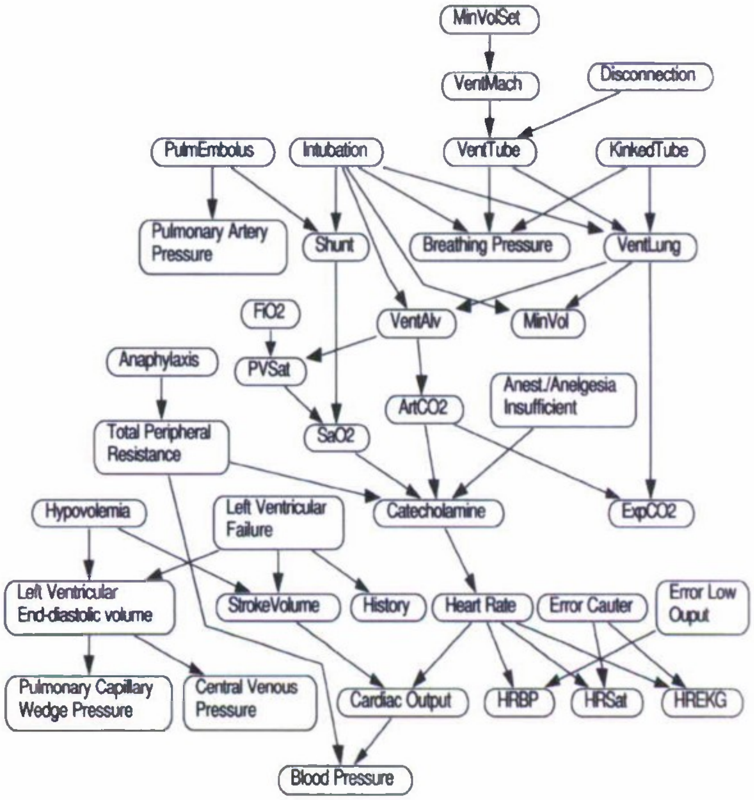


Fig. 11. ALARM: network constructed by medical expert for monitoring patients in intensive care.

methods can perform very poorly using the same amount of time. In addition, there are several ways to reduce the computational burden such as assuming that some interface nodes with small correlations are independent of each other. Nevertheless, this is beyond the scope of the paper and could be an interesting topic for future research.

## VI. CONCLUSION

In this paper, we develop a hybrid propagation algorithm for general BNs with mixed discrete and continuous variables. In the algorithm, we first partition the network into discrete and continuous segments by introducing the interface nodes. We then apply message passing for each network segment and encode the updated information as messages to be exchanged between segments through the set of interface nodes. Finally we integrate the separate messages from different network segments and compute the a posterior distributions for all hidden nodes. The preliminary simulation results show that the algorithm works well for hybrid BN models.

The main contribution of this paper is to provide a general framework for inference in hybrid model. Based on the principle of decomposition and conditioning, we introduce the set of interface nodes to partition the network. Therefore it is possible to apply exact inference algorithms such as junction tree to some applicable network segments which enables the integration of different efficient algorithms from multiple subnetworks. For complicated network segment such as the one with nonlinear and/or non-Gaussian variables, we provide options to use a loopy-type message passing algorithm.

Although the bottleneck of our algorithm is the size of interface nodes, we believe that HMP-BN is a good alternative for nonlinear and/or non-Gaussian hybrid models since no efficient algorithm exists for this case (as far as we know from the literature), especially given unlikely evidence. We are currently exploring another idea of propagating messages directly between different types of nodes without network partition or interface nodes. However, it is beyond the scope of the current paper.

Note that the focus of this paper is on developing a unified message passing algorithm for general hybrid networks. While the algorithm works well to estimate the means and variances for the hidden continuous variables, the true posterior distributions may have multiple modes. In practice, it might be more important to know where the probability mass is than just knowing mean and variance. One idea for future research is to utilize the messages computed in HMP-BN to obtain a good importance function and apply importance sampling to estimate the probability distributions. Another future research direction is to extend the hybrid algorithm to the general BN models without restriction of node ordering, such as to allow continuous parents for discrete variables. If successful, it would be a significant step forward.

## REFERENCES

[1] Arulampalam, S., Maskell, S., Gordon, N., and Clapp, T.
A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking.
*IEEE Transoctions on Signal Processing*, **50**, 2 (Feb. 2002), 174–188.

[2] Beadle, E. R., and Djuric, P. M.
A fast-weighted Bayesian bootstrap filter for nonlinear model state estimation.
*IEEE Transactions on Aerospace and Electronic Systems*, **33**, 1 (Jan. 1997), 338–343.

[3] Bar-Shalom, Y., Li, X. R., and Kirubarajan, T.
*Estimation with Applications to Tracking and Navigation*.
New York: Wiley, 2001.

[4] Beinlich, I., Suermondt, G., Chavez, R., and Cooper, G.
The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks.
In *Proceedings of 2nd European Conference on AI and Medicine*, 1989.

[5] Cheng, J., and Druzdzel, M. J.
AIS-BN: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks.
*Journal of Artificiol Intelligence Research* (JAIR), **13** (2000), 155–188.

[6] Chang, K. C.
Almost instant time inference for hybrid partially dynamic Bayesian networks.
*IEEE Transactions on Aerospace and Electronic Systems*, **43**, 1 (Jan. 2007), 13–22.

[7] Charniak, E.
Bayesian networks without tears: Making Bayesian networks more accessible to the probabilistically unsophisticated.
*AI Magazine*, **12**, 4 (1991), 50–63.

[8] Cooper, G. F.
The computational complexity of probabilistic inference using Bayesian belief networks.
*Artificial Intelligence*, **42** (1990), 393–405.

[9] Cobb, B. R., and Shenoy, P. P.
Inference in hybrid Bayesian networks with mixtures of truncated exponentials.
*International Journal of Approximate Reasoning*, **41**, 3 (Apr. 2006), 257–286.

[10] Dagum, P., and Luby, M.
Approximating probabilistic inference in Bayesian belief networks is NP-hard.
*Artificial Intelligence*, **60** (1993), 141–153.

[11] Fung, R., and Chang, K. C.
Weighting and integrating evidence for stochastic simulation in Bayesian networks.
In *Uncertainty in Artificial Intelligence 5*, New York: Elsevier, 1989, 209–219.

[12] Guo, H., and Hsu, W.
A Survey of algorithms for real-time Bayesian network inference.
Presented at AAAI/KDD/UAI—2002 Joint Workshop on Real-Time Decision Support and Diagnosis Systems, Edmonton, Alberta, Canada, 2002.

[13] Jensen, F. V.
*An Introduction to Bayesian Networks*.
New York: Springer-Verlag, 1996.

[14] Julier, S. J.
The scaled unscented transformation.
In *Proceedings of the American Control Conference*, vol. 6, May 2002, 4555–4559.

[15] Julier, S. J., and Uhlmann, J. K.
A general method for approximating non-linear transformations of probability distribution.
Dept. of Engineering Science, University of Oxford, Technical Report, RRG, Nov. 1996.

[16] Koller, D., Lerner, U., and Angelov, D.
A general algorithm for approximate inference and its application to hybrid Bayes nets.
In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence* (UAI-99), Stockholm, Sweden, July 30–Aug. 1, 1999, 324–333.

[17] Lauritzen, S. L.
Propagation of probabilities, means, and variances in mixed graphical association models.
*JASA*, **87**, 420 (1992), 1089–1108.

[18] Lerner, U. N.
Hybrid Bayesian Networks for Reasoning about Complex Systems.
Ph.D. dissertation, Stanford University, Stanford, CA, Oct. 2002.

[19] Lauritzen, S. L., and Jensen, F.
Stable local computations with conditional Gaussian distributions.
*Statistics and Computing*, **11**, 2 (Apr. 2001), 191–203.

[20] Lauritzen, S. L., and Spiegelhalter, D. J.
Local computations with probabilities on graphical structures and their applications to expert systems.
In *Proceedings of the Royal Statistical Society*, Series B, 50 (1988), 157–224.

[21] Moral, S., Rumi, R., and Salmeron, A.
Mixtures of truncated exponentials in hybrid Bayesian networks.
In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, vol. 2143, Berlin: Springer-Verlag, 156–167.

[22] Murphy, K., Weiss, Y., and Jordan, M.
Loopy belief propagation for approximate inference: An empirical study.
In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence* (UAI-99), San Francisco, CA: Morgan Kaufmann Publishers, 1999, 467–475.

[23] Neapolitan, R. E.
*Probabilistic Reasoning in Expert Systems.*
New York: Wiley, 1990.

[24] Pearl, J.
*Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.*
San Mateo, CA: Morgan Kauffinan, 1988.

[25] Sun, W., and Chang, K. C.
Unscented message passing for arbitrary continuous Bayesian networks.
In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, Vancouver, Canada, July 2007.

[26] Sun, W., and Chang, K. C.
Hybrid message passing for mixed Bayesian networks.
In *Proceedings of the 10th International Conference on Information Fusion*, Quebec, Canada, July 2007.

[27] Shachter, R. D., and Peot, M. A.
Simulation approaches to general probabilistic inference on belief networks.
In *Proceedings of the Conference on Uncertainty in AI*, vol. 5, 1990.

[28] Shenoy, P. P., and Shafer, G. R.
Axioms for probability and belief-function propagation.
In *Proceedings Uncertainty of Artificial Intelligence*, vol. 4, 1990, 169–198.

[29] Weiss, Y., and Freeman, W. T.
Correctness of belief propagation in Gaussian graphical models of arbitrary topology.
University of California at Berkeley, Computer Science Dept., Technical Report, UCB.CSD-99-1046, 1999.

[30] Yuan, C., and Druzdzel, M. J.
Hybrid loopy belief propagation.
In M. Studeny and J. Vomlel (Eds.), *Proceedings of the Third European Workshop on Probabilistic Graphical Models* (PGM-06), Prague, 2006, 317–324.

**Wei Sun** received his B.S. in electrical engineering in 1991 from Zhejiang University, Hangzhou, China. He received the M.S. and Ph.D. in operations research from George Mason University, Fairfax, VA, in 2003 and 2007, respectively.

He worked with Guizhou No.1 Power Plant Construction Company in China before he came to the United States for his graduate studies. He is currently a senior analyst in the Department of Enterprise Optimization at United Airlines. His research interests include artificial intelligence, Bayesian networks, simulation, and optimization. In the last few years, he has been focusing on inference algorithm development for hybrid Bayesian networks.

Dr. Sun is active in presenting and publishing papers in several international conferences such as AAAI, International Conference on Information Fusion, and SPIE.

**Kuo-Chu Chang** received the M.S. and Ph.D. degrees in electrical engineering from the University of Connecticut, Storrs, in 1983 and 1986, respectively.

From 1983 to 1992, he was a senior research scientist in Advanced Decision Systems (ADS) division, Booz-Allen & Hamilton, Mountain View, CA. In 1992, he joined the Systems Engineering and Operations Research Department, George Mason University where he is currently a professor. His research interests include estimation theory, optimization, signal processing, and multisensor data fusion. He is particularly interested in applying unconventional techniques in the conventional decision and control systems.

He has more than 25 years of industrial and academic experience and has published more than 150 papers in the areas of multitarget tracking, distributed sensor fusion, and Bayesian networks technologies. He was an associate editor on Tracking/Navigation Systems from 1993 to 1996 and on Large Scale Systems from 1996 to 2006 for *IEEE Transactions on Aerospace and Electronic Systems.* He was also an associate editor of *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, from 2002 to 2007.

Dr. Chang is a member of Etta Kappa Nu and Tau Beta Pi.

# Analytical and Computational Evaluation of Scalable Distributed Fusion Algorithms

**KC CHANG**
George Mason University

**CHEE-YEE CHONG**
**SHOZO MORI**
BAE Systems

**The theoretical fundamentals of distributed information fusion have been developed over the past two decades and are now fairly well established. 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 because of 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, naïve fusion, Chernoff fusion, Shannon fusion, and Bhattacharyya fusion algorithms. We also compare their performance to "optimal" centralized fusion under a specific communication pattern. The results show that the channel filter fusion, representing a first order approximation to the information graph fusion, is the only "consistent" fusion algorithm.**

Manuscript received February 15, 2009; revised June 26, 2009; released for publication August 8, 2009.

IEEE Log No. T-AES/46/4/938814.

Refereeing of this contribution was handled by Wolfgang Koch.

Authors' addresses: KC Chang, Dept. of Systems Engineering and Operations Research, George Mason University, Fairfax, VA 22030, E-mail (kchang@gmu.edu); Chee-Yee Chong and Shozo Mori, BAE Systems, Advanced Information Technologies, Los Altos, CA 94022.

## I. INTRODUCTION

A distributed data fusion system consists of a network of sensors and processors that may be colocated with the sensors. Sensors generate data by observing the environment. Processors process local sensor data and fuse data from other sensors or processors. The performance of a distributed fusion system over a network depends on three factors: the network architecture, the reliability of communication links within the network, and applicable fusion algorithms. Even though the network architecture may be fixed and known, adaptive communication strategies and possible communication link failures will result in a dynamically changing communication structure among the fusion nodes. Thus, a distributed fusion algorithm is not really practical unless it can handle a dynamic communication structure.

There has been a great deal of work in developing distributed fusion algorithms applicable to a network centric architecture [1–5]. However, most of these algorithms have been designed for fixed communication structures and may not be practical for distributed systems such as ad hoc sensor networks where the communication architecture changes dynamically [6]. In particular, the distributed fusion algorithm based on the information graph approach [7] was developed to optimally combine information from multiple nodes by maintaining information pedigree and using it to avoid any double counting of information. However, when the communication structure changes in real time, this algorithm may not scale because of its requirements to carry long pedigree information for decorrelation.

In this paper, we 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 Fig. 1. We conduct extensive simulations to validate the theoretical predictions. We have chosen this network structure because of its complexity due to multiple paths for 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.

Channel filter
Naïve fusion
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

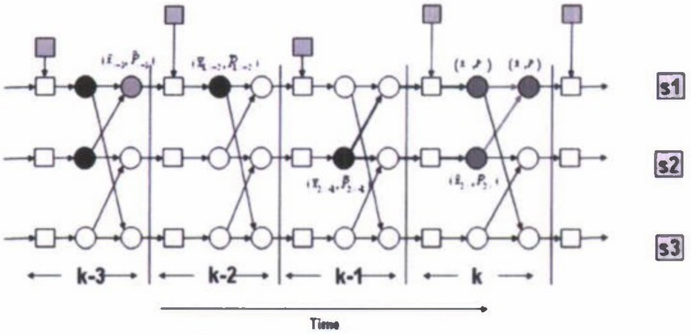


Fig. 1. Three sensor cyclic communication scenario.

part of a wider objective to understand the system trades involved in a general decentralized ad hoc sensor network. The rest of this paper is organized as follows. Section II briefly describes the set of scalable distributed fusion algorithms to be considered in this paper. Section III derives the analytical fusion performance evaluation in terms of steady-state mean square error. Section IV summarizes the technical findings of the study, and Section V presents some future research directions.

## II. SCALABLE FUSION ALGORITHMS

The theoretic fundamentals of distributed information fusion are well documented and have been studied in depth [7–11]. It is noted, however, that practical applications of these theoretical results to nondeterministic information flow have remained a challenge. The main difficulty is the need to identify and remove common information from the data sets to be fused, while minimizing the amount of data exchanged between agents.

The basic fusion process, as described in [7], follows from set theory, where the combination of $n$ event probabilities $\Phi(\cdot \mid I_i)$ given the information $I_i$ can be represented as

$$\Phi\left(\cdot \,\middle|\, \bigcup_{i=1}^{n} I_i\right) = \frac{1}{C}\prod_{i=1}^{n} S_i^{(-1)^{i-1}} \tag{1}$$

where $S_i$ represents the combination of $i$ event probabilities such that, $S_1 = \prod_{i=1}^{n}\Phi(\cdot \mid I_i)$, $S_2 = \prod_{i=1,j\in\{i+1,\ldots,n\}}^{n-1}\Phi(\cdot \mid I_i \cap I_j),\ldots,S_n = \Phi(\cdot \mid I_1 \cap I_2 \cap \cdots I_n)$. 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 [7–9] 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 IG is shown in Fig. 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 IG 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 Fig. 1, the resulting formula for the fusion between the first two sensors at time $k$ is [7]

$$p(x) = \frac{1}{c}\frac{p_{1,k}(x)p_{2,k}(x)p_{1,k-3}(x)}{p_{1,k-2}(x)p_{2,k-1}(x)} \tag{2}$$

where $c$ is the normalization constant, $p(x)$ is the conditional probability at node $s1$ after fusion, and $p_{i,k}(x)$ 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 Fig. 1)

$$\begin{aligned} P_k^{-1} &= P_{1,k}^{-1} + P_{2,k}^{-1} - \bar{P}_{1,k-2}^{-1} - \bar{P}_{2,k-1}^{-1} + P_{k-3}^{-1} \\ P_k^{-1}\hat{x}_k &= P_{1,k}^{-1}\hat{x}_{1,k} + P_{2,k}^{-1}\hat{x}_{2,k} - \bar{P}_{1,k-2}^{-1}\bar{x}_{1,k-2} \\ &\quad - \bar{P}_{2,k-1}^{-1}\bar{x}_{2,k-1} + P_{k-3}^{-1}\hat{x}_{k-3}. \end{aligned} \tag{3}$$

In general, to construct the "optimal"[1] fusion formula may require carrying long pedigree information[2] that

[1] The IG approach is optimal when the underlying system is deterministic.

[2] Information includes communication and fusion events history as well as past fusion data.

might not be practical in an environment with limited communication bandwidth [12].

To address the scalability issue, we have developed each of the fusion algorithms described in the following sections for autonomous sensors in arbitrary network conditions. All of these approaches are suboptimal in general but provide adequate performance when basic assumptions are met.

### A. Channel Filter

The channel filter approach [13–16] is simpler than IG fusion in that only the first order redundant information is considered. Each channel is defined by a pair of agents—a transmitting agent and a 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 because of 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 optimal IG approach.

Specifically, the channel filter fusion equation is given as

$$p(x) = \frac{p_1(x)p_2(x)/\bar{p}(x)}{\int [p_1(x)p_2(x)/\bar{p}(x)]dx} \tag{4}$$

where $p_1(x)$ and $p_2(x)$ are the two probability density functions to be fused (one local and the other received from a particular channel) and $\bar{p}(x)$ 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 estimate and corresponding covariance error can be written as

$$\begin{aligned} P^{-1} &= P_1^{-1} + P_2^{-1} - \bar{P}^{-1} \\ P^{-1}\hat{x} &= P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2 - \bar{P}^{-1}\bar{x}. \end{aligned} \tag{5}$$

While simpler, it is obvious that dependent information is more likely to be lost in the channel filter when compared with the IG approach. On the other hand, if the time between when that redundancy occurred and the current processing time is relatively long, the impact could be minimal.

### B. Naïve Fusion

Naïve fusion is the simplest fusion approach, where it is assumed that the dependency between the density functions is negligible. This fusion approach is the simplest type, but it can be unreliable. The naïve fusion formula can be written as

$$p(x) = \frac{p_1(x)p_2(x)}{\int p_1(x)p_2(x)dx}. \tag{6}$$

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

$$\begin{aligned} P^{-1} &= P_1^{-1} + P_2^{-1} \\ P^{-1}\hat{x} &= P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2. \end{aligned} \tag{7}$$

Note that the fused track covariance is the inverse of the sum of the inverses of the local track covariance matrices. Thus, because of 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).

### C. Chernoff Fusion

When the dependency between two distributions is unknown, one idea is to use the Chernoff information [17]. The fusion formula is based on the following:

$$p(x) = \frac{p_1^w(x)p_2^{1-w}(x)}{\int p_1^w(x)p_2^{1-w}(x)dx} \tag{8}$$

where $w \in [0 \;\; 1]$ is an appropriate parameter that 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 [17, 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

$$\begin{aligned} P^{-1} &= wP_1^{-1} + (1-w)P_2^{-1} \\ P^{-1}\hat{x} &= wP_1^{-1}\hat{x}_1 + (1-w)P_2^{-1}\hat{x}_2. \end{aligned} \tag{9}$$

This formula is identical to the covariance intersection (CI) fusion technique [14–15]. 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 [18].

### D. Shannon Fusion

A special case of (8) is when the parameter $w$ is chosen to minimize the determinant of the fused covariance [18, 19]. In the Gaussian case, it is equivalent to minimizing the Shannon information of the fused density. This is because the Shannon information defined as $I_s = -\int_x p(x) \ln p(x) dx$ can be shown to be equal to $I_s = \frac{1}{2}\ln((2\pi)^n |P|^{1/2}) + n/2$ when $p(x)$ is Gaussian with covariance $P$ [18]. 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 [18].

### E. 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, $p(x) = (1/B)\sqrt{p_1(x)p_2(x)}$, the Bhattacharyya fusion. When both $p_1(x)$ and $p_2(x)$ are Gaussian, the fusion equation can be written as

$$\begin{aligned}
P^{-1} &= \tfrac{1}{2}(P_1^{-1} + P_2^{-1}) \\
P^{-1}\hat{x} &= \tfrac{1}{2}(P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2) \qquad (10) \\
&\Rightarrow \hat{x} = (P_1^{-1} + P_2^{-1})^{-1}(P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2).
\end{aligned}$$

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

$$\begin{aligned}
P^{-1} &= \tfrac{1}{2}(P_1^{-1} + P_2^{-1}) = (P_1^{-1} + P_2^{-1}) - \tfrac{1}{2}(P_1^{-1} + P_2^{-1}) \\
P^{-1}\hat{x} &= \tfrac{1}{2}(P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2) \qquad (11) \\
&= (P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2) - \tfrac{1}{2}(P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2).
\end{aligned}$$

This formula replaces the common prior information of (5) for the channel filter by the average of the two sets of information to be fused. Namely, $\bar{P}^{-1} \leftarrow \frac{1}{2}(P_1^{-1} + P_2^{-1})$ and $\bar{P}^{-1}\bar{x} \leftarrow \frac{1}{2}(P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2)$. 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 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 Fig. 1. We will also conduct extensive simulation to evaluate other alternative fusion algorithms.

## III. ANALYTICAL PERFORMANCE PREDICTION

As shown in Fig. 1, wherc at timc k we define 1) $\hat{x} \equiv \hat{x}_{k|k}; P_0 \equiv P_{k|k}$ and $\bar{x} \equiv \hat{x}_{k-1|k-1}; \bar{P} \equiv P_{k-1|k-1}$ as the fused state estimates and the associated filter covariances at time $k$ and $k-1$; 2) $\hat{x}_i \equiv \hat{x}_{i,k|k}; P_i \equiv P_{i,k|k}$ as the local updated state estimates and thc associated filter covariances; and 3) $\bar{x}_i \equiv \hat{x}_{i,k-1|k-1}; \bar{P}_i \equiv P_{i,k-1|k-1}$ 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, $\Omega = \lim_{k\to\infty} E[(\hat{x}_{k|k} - x_k)(\hat{x}_{k|k} - x_k)'] = E[(\hat{x} - x)(\hat{x} - x)']$. 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., $z_{i,k} = x_k + w_{i,k}$ where $w_{i,k}$ is a zero-mean Gaussian measurement noise with variance $R_i$ for sensor $i$. In the following, we assume that the sensors have the same quality, i.e., $R_1 = R_2 = R_3 = R$. Therefore, in steady state, let $P_i = P$, then $\bar{P}_i = P_{i,k-1|k-1} = P_{i,k|k} = P_i = P$.

### A. Channel Filter

With a channel filter, as shown in (5), the fusion equations are written as

$$P_0^{-1} = P_1^{-1} + P_2^{-1} - \bar{P}_{2,k|k-1}^{-1} \qquad (12)$$

$$P_0^{-1}\hat{x} = P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2 - \bar{P}_{2,k|k-1}^{-1}\bar{x}_{2,k|k-1}. \qquad (13)$$

Equation (13) can be rewritten as,

$$\begin{aligned}
A \equiv P_0^{-1}(\hat{x} - x) &= P_1^{-1}(\hat{x}_1 - x) + P_2^{-1}(\hat{x}_2 - x) \\
&\quad - \bar{P}_{2,k|k-1}^{-1}(\bar{x}_{2,k|k-1} - x) \\
&= P^{-1}(\hat{x}_1 - x) + P^{-1}(\hat{x}_2 - x) - (P + Q)^{-1}(\bar{x}_2 - x) \\
&\Rightarrow (\hat{x} - x) = P_0 A \\
&= P_0 P^{-1}(\hat{x}_1 - x) + P_0 P^{-1}(\hat{x}_2 - x) \\
&\quad - P_0(P + Q)^{-1}(\bar{x}_2 - x). \qquad (14)
\end{aligned}$$

Therefore,

$$\Omega \equiv E[(\hat{x} - x)^2] = P_0 E(AA')P_0'. \qquad (15)$$

In the scalar case,

$$\begin{aligned}
(\hat{x} - x)^2 &= \frac{P_0^2}{P^2}(\hat{x}_1 - x)^2 + \frac{P_0^2}{P^2}(\hat{x}_2 - x)^2 + \frac{P_0^2}{(P + Q)^2}(\bar{x}_2 - x)^2 \\
&\quad + \frac{2P_0^2(\hat{x}_1 - x)(\hat{x}_2 - x)}{P^2} - \frac{2P_0^2(\hat{x}_1 - x)(\bar{x}_2 - x)}{P(P + Q)} \\
&\quad - \frac{2P_0^2(\hat{x}_2 - x)(\bar{x}_2 - x)}{P(P + Q)} \qquad (16)
\end{aligned}$$

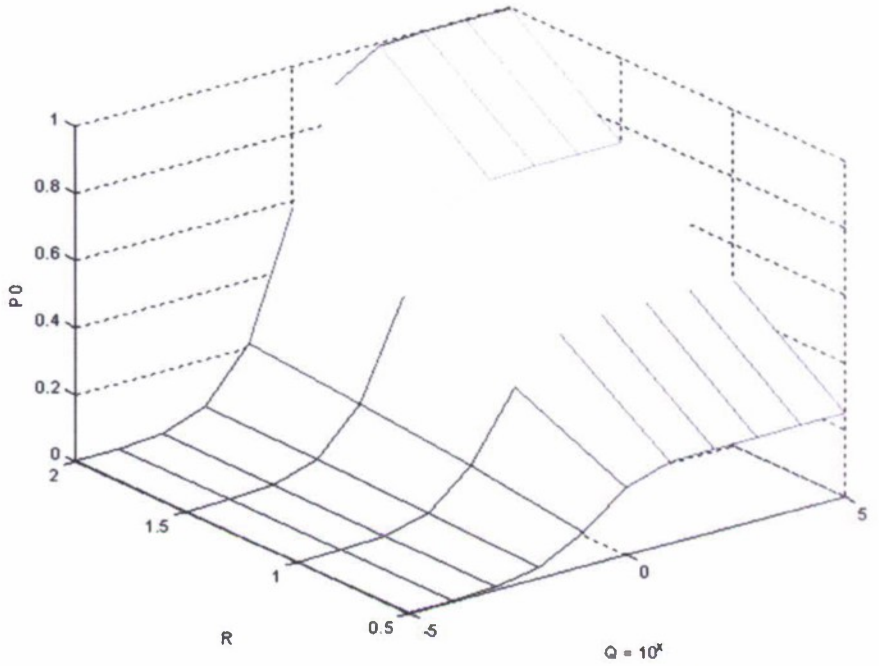


Fig. 2. Steady-state filter variances $P$ and $P_0$ for various $Q$ and $R$.

$$\Rightarrow \Omega = \frac{2P_0^2}{P^2}(B+E') + \frac{P_0^2}{(P+Q)^2}(B+Q) - \frac{2P_0^2}{P(P+Q)}(C_1+C_2) \tag{17}$$

where

$$\begin{aligned} E[(\bar{x}_2 - x)^2] &= E[(\bar{x}_{2,k|k-1} - x_k)^2] \\ &= E[(\bar{x}_{2,k-1|k-1} - x_{k-1} - v_{k-1})^2] \\ &= B + Q \end{aligned} \tag{18}$$

$$B \equiv E[(\hat{x}_1 - x)^2] = E[(\hat{x}_2 - x)^2] \tag{19}$$

$$E' \equiv E[(\hat{x}_1 - x)(\hat{x}_2 - x)] \tag{20}$$

$$\begin{aligned} C_1 &\equiv E[(\hat{x}_2 - x)(\bar{x}_2 - x)] \qquad \text{and} \\ C_2 &\equiv E[(\hat{x}_1 - x)(\bar{x}_2 - x)]. \end{aligned} \tag{21}$$

Note that in (17), $P_0$ and $P$ are the steady-state "filter" variances. They can be obtained by solving the following two equations:

$$\begin{aligned} P_0^{-1} &= P_1^{-1} + P_2^{-1} - \bar{P}_{2,k|k-1}^{-1} = 2P^{-1} - (P+Q)^{-1} \\ &\Rightarrow P_0 = P(P+Q)/(P+2Q) \end{aligned} \tag{22}$$

$$\begin{aligned} P &= (P_0 + Q) - KSK' = (P_0+Q) - \frac{(P_0+Q)^2}{(P_0+Q+R)} \\ &= \frac{(P_0+Q)R}{(P_0+Q+R)} \end{aligned} \tag{23}$$

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

$$P^3 + (2Q)P^2 + (2Q^2)P - (2Q^2)R = 0. \tag{24}$$

A closed form real solution of the preceding cubic polynomial can be solved and the resulting $P_0$ as a function of various $Q$ and $R$ is shown in Fig. 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),

$$B = (1-K)^2\Omega + (1-K)^2 Q + K^2 R \equiv \lambda\Omega + \alpha \tag{25}$$

$$\begin{aligned} E' &= (1-K)^2\{E[(\bar{x}_1' - x_{k-1})(\bar{x}_2' - x_{k-1})] + Q\} \\ &\equiv (1-K)^2[E_p + Q] \end{aligned} \tag{26}$$

$$E_p = \left(\frac{P_0}{P}\right)^2 (3E' + B) + \left(\frac{P_0}{P+Q}\right)^2 (E' + Q) - \left(\frac{P_0^2}{P(P+Q)}\right)(C_1 + C_2 + 2D) \tag{27}$$

$$\begin{aligned} C_1 = C_2 &= \frac{(1-K)\left(\frac{P_0}{P}B + \frac{P_0}{P}E' + Q\right)}{1 + (1-K)\frac{P_0}{P+Q}} \\ &= \frac{(1-K)(P+Q)}{2(P+Q) - KP}(E' + B) + \frac{(1-K)(P+2Q)}{2(P+Q) - KP}Q \\ &\equiv \eta(E' + B) + \eta' \end{aligned} \tag{28}$$

and

$$D = \eta(2E') + \eta'. \tag{29}$$

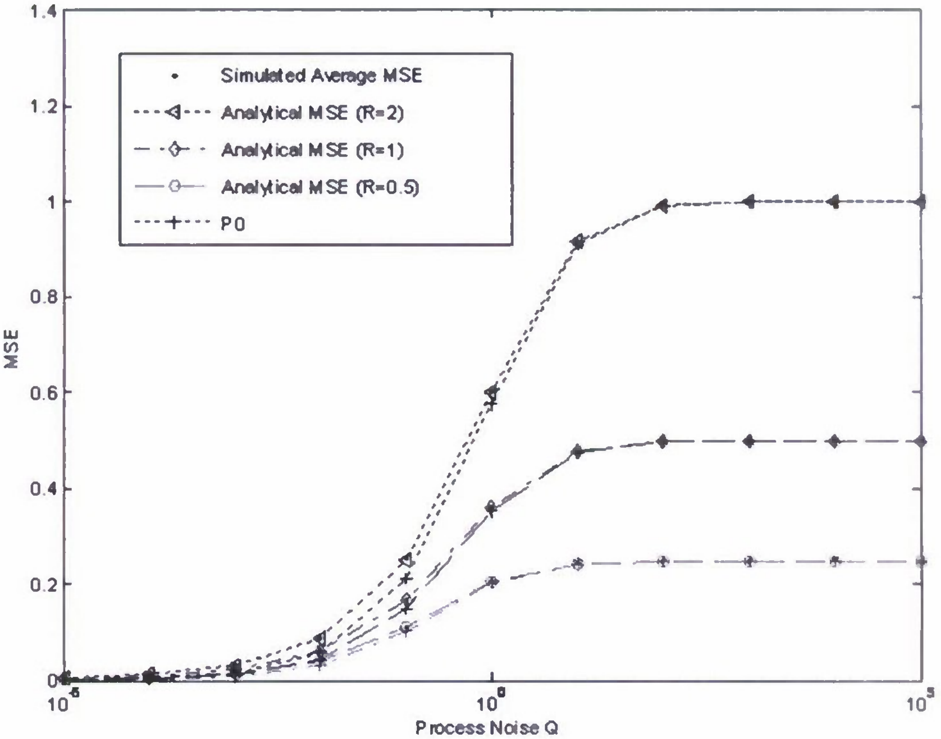


Fig. 3. Comparison of channel filter analytical MSE with simulated MSE (1000 MC trials).

Using relations (25)–(29), (17) can be rewritten as

$$\begin{aligned}\Omega &= \frac{2P_0^2}{P^2}(B+E') + \frac{P_0^2}{(P+Q)^2}(B+Q) - \frac{2P_0^2}{P(P+Q)}(2C)\\ &= \left(\frac{2P_0^2}{P^2} + \frac{P_0^2}{(P+Q)^2} - \frac{4P_0^2}{P(P+Q)}\eta\right)B\\ &\quad + \left(\frac{2P_0^2}{P^2} - \frac{4P_0^2}{P(P+Q)}\eta\right)E'\\ &\quad + \frac{P_0^2}{(P+Q)^2}Q - \frac{4P_0^2}{P(P+Q)}\eta'\\ &\equiv l_1 B + l_2 E' + l_3 = (l_1 + l_2 f_1)B + l_2 f_2 + l_3\\ &\equiv m_1 B + m_2 = m_1(\lambda\Omega + \alpha) + m_2\\ &\Rightarrow \Omega = \frac{m_1\alpha + m_2}{1 - m_1\lambda}. \end{aligned} \tag{30}$$

Fig. 3 compares the analytical mean square errors (MSE) based on (30) with the average MSE based on 1000 Monte Carlo simulation trials. It is clear that they are in perfect agreement. Fig. 3 also shows that the filter variance $P_0$ is very close to the true MSE, which indicates that the algorithm behaves well and is reasonably consistent [9].

### B. Naïve Fusion

With the notations defined earlier, the naïve fusion equations can be written as

$$P_0 = (P_1^{-1} + P_2^{-1})^{-1} = P/2 \tag{31}$$

$$\hat{x} = P_0(P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2) = (\hat{x}_1 + \hat{x}_2)/2. \tag{32}$$

From (31), $(\hat{x} - x) = [(\hat{x}_1 - x) + (\hat{x}_2 - x)]/2$; therefore,

$$\begin{aligned}\Omega &\equiv E[(\hat{x} - x)^2]\\ &= \tfrac{1}{4}[E[(\hat{x}_1 - x)^2] + E[(\hat{x}_2 - x)^2] + 2E[(\hat{x}_1 - x)(\hat{x}_2 - x)]]\\ &= \tfrac{1}{2}(B + E')\end{aligned} \tag{33}$$

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

From (25), $B = \lambda\Omega + \alpha$, and from (26),

$$\begin{aligned}E' &= E[(\hat{x}_1 - x_k)(\hat{x}_2 - x_k)]\\ &= (1-K)^2\{E[(\bar{x}'_1 - x_{k-1})(\bar{x}'_2 - x_{k-1})] + Q\}\\ &= \tfrac{1}{4}(1-K)^2(B + 3E' + 4Q)\\ &\Rightarrow E' = \frac{(1-K)^2}{4 - 3(1-K)^2}(B + 4Q) \equiv \mu(B + 4Q).\end{aligned} \tag{34}$$

Therefore, from (25), (33), and (34), we have,

$$\begin{aligned}\Omega &= \tfrac{1}{2}(B + E') = \tfrac{1}{2}(1+\mu)B + 2\mu Q\\ &= \tfrac{1}{2}(1+\mu)(\lambda\Omega + \alpha) + 2\mu Q \Rightarrow \Omega = \frac{(1+\mu)\alpha/2 + 2\mu Q}{1 - (1+\mu)\lambda/2}.\end{aligned} \tag{35}$$

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

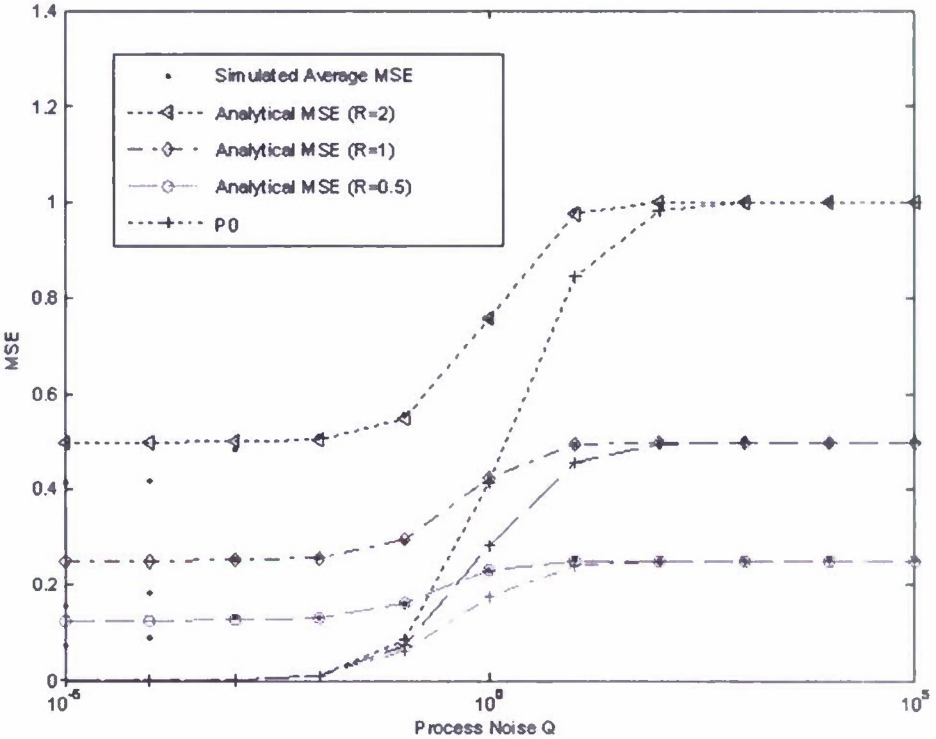


Fig. 4. Comparison of naïve fusion analytical MSE with simulated MSE (1000 MC trials).

equations:

$$P_0^{-1} = P_1^{-1} + P_2^{-1} = 2P^{-1} \Rightarrow P_0 = P/2 \tag{36}$$

$$\begin{aligned} P &= (P_0 + Q) - KSK' = (P_0 + Q) - \frac{(P_0 + Q)^2}{(P_0 + Q + R)} \\ &= \frac{(P/2 + Q)R}{(P/2 + Q + R)}. \end{aligned} \tag{37}$$

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

$$\begin{aligned} & P^2 + (2Q + R)P - 2QR = 0 \\ & \Rightarrow P = \frac{\sqrt{(2Q + R)^2 + 8QR} - (2Q + R)}{2}. \end{aligned} \tag{38}$$

Fig. 4 compares the analytical MSEs based on (35) with the average MSE based on 1000 Monte Carlo simulation trials. It is clear that they are very close to each other when the 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 numerical round off error caused by the small magnitude of the noise. Fig. 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 [11].

### C. Bhattacharyya Fusion

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

$$P_0 = 2(P_1^{-1} + P_2^{-1})^{-1} = P \tag{39}$$

$$\hat{x} = \tfrac{1}{2}P_0(P_1^{-1}\hat{x}_1 + P_2^{-1}\hat{x}_2) = (\hat{x}_1 + \hat{x}_2)/2. \tag{40}$$

As in (33)

$$\begin{aligned} \Omega &\equiv E[(\hat{x} - x)^2] \\ &= \tfrac{1}{4}\{E[(\hat{x}_1 - x)^2] + E[(\hat{x}_2 - x)^2] + 2E[(\hat{x}_1 - x)(\hat{x}_2 - x)]\} \\ &= \tfrac{1}{2}(B + E') \end{aligned} \tag{41}$$

where, as defined before, $B = \lambda\Omega + \alpha$ and $E' = [(1 - K)^2/4 - 3(1 - K)^2](B + 4Q) \equiv \mu(B + 4Q)$. Therefore, as in (35)

$$\Omega = \frac{(1 + \mu)\alpha/2 + 2\mu Q}{1 - (1 + \mu)\lambda/2}. \tag{42}$$

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:

$$\begin{aligned} P &= (P_0 + Q) - KSK' = (P_0 + Q) - \frac{(P_0 + Q)^2}{(P_0 + Q + R)} \\ &= \frac{(P + Q)R}{(P + Q + R)}. \end{aligned} \tag{43}$$

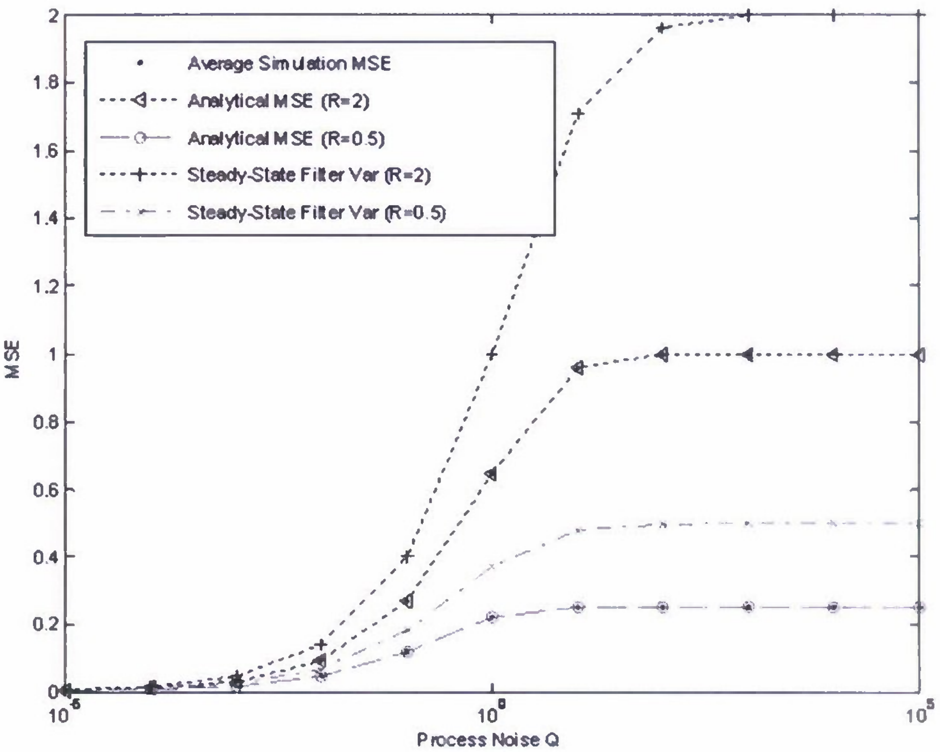


Fig. 5. Comparison of Bhattacharyya fusion analytical and simulated MSE (1000 MC trials).

From (43), it can be easily shown that

$$P^2 + PQ - QR = 0 \Rightarrow P = \frac{\sqrt{Q^2 + 4QR} - Q}{2}. \tag{44}$$

Fig. 5 compares the analytical MSEs based on (41) with the average MSE based on 1000 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.

## IV. 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 optimal centralized fusion. The results are shown in Fig. 6. As can be seen, in addition to naïve fusion, Shannon fusion also performs poorly. This is because in the 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.

As shown in Fig. 6, the remaining three algorithms have very similar performance. A closer look (Fig. 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 Fig. 8, which compares the performance of channel filter, Chernoff fusion, and Bhattacharyya fusion versus 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 the 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 reliabilities. Because all algorithms under consideration are scalable and autonomous, no additional changes are necessary in the algorithms for the test. The results in Fig. 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.

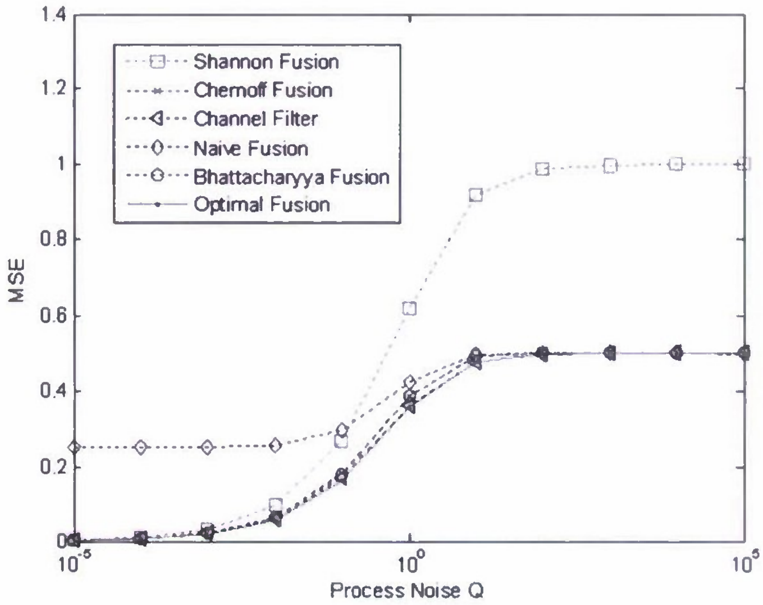


Fig. 6. Comparison of alternative fusion algorithms with optimal fusion algorithm ($R_1 = R_2 = R_3 = 1$).

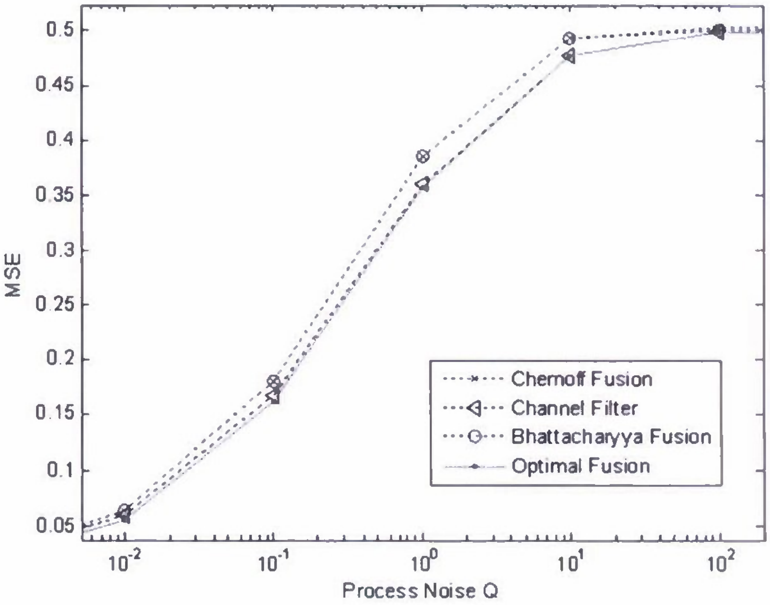


Fig. 7. Comparison of channel filter, Chernoff fusion, and Bhattacharyya fusion ($R_1 = R_2 = R_3 = 1$).

It should be noted that channel filter, while requiring a one-step memory 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 channel filter is at least one order of magnitude faster than the Chernoff fusion. Further investigation is needed to compare the trade-offs between these promising algorithms in a more reliable manner.

## V. 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, naïve fusion, Chernoff fusion, Shannon fusion, and Bhattacharyya fusion algorithms. We also compare their performance to "optimal" centralized fusion algorithms under a specific communication pattern.

The results show that naïve fusion and Shannon fusion perform poorly while several other scalable

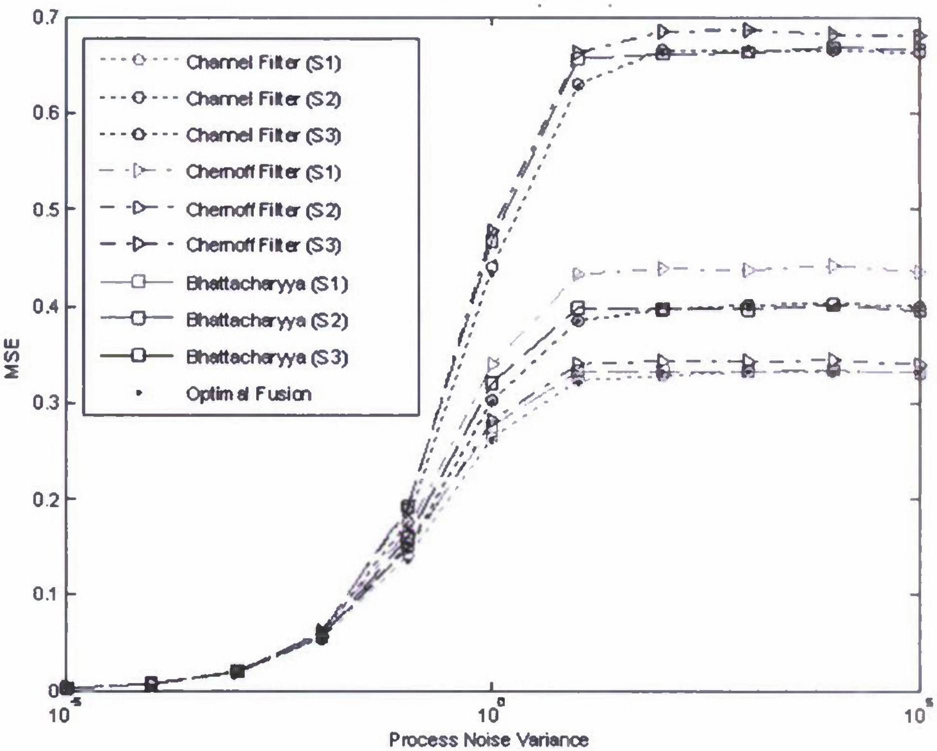


Fig. 8. Channel filter, Chernoff fusion, and Bhatacharyya fusion versus optimal fusion with sensors of different qualities ($R_1 = 0.5$, $R_2 = 1.0$, $R_3 = 2.0$).

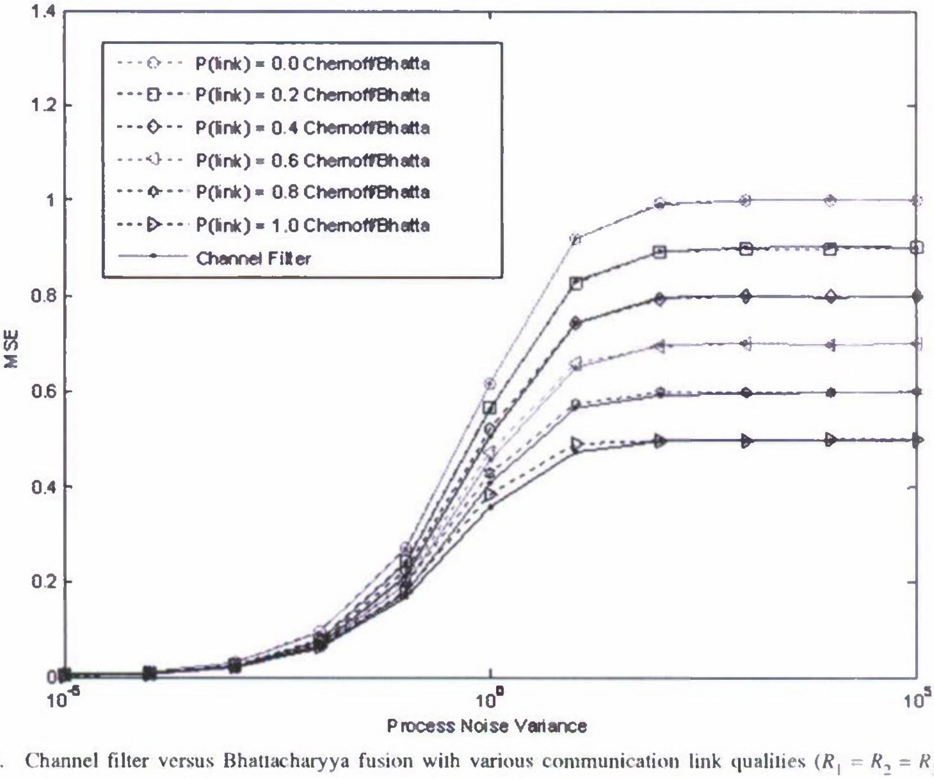


Fig. 9. Channel filter versus Bhattacharyya fusion with various communication link qualities ($R_1 = R_2 = R_3 = 1$).

algorithms including channel filter, Chernoff fusion, and Bhattacharyya fusion, require minimum communication and perform fairly well. Their performance is comparable with that of the optimal fusion algorithm. In particular the channel filter fusion, representing a first-order approximation to IG fusion, works surprisingly well and has been shown to be the only "consistent" fusion algorithm.

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. It is important to consider heterogeneous sensors with different sampling interval and error characteristics under dynamic communication topology and constraints. It is also useful to develop theoretical analysis for specific algorithms whenever possible for a given network scenario.

## REFERENCES

[1] U.S. Department of Defense, Office of Force Transformation
*The Implementation of Network-Centric Warfare.*
Washington, DC: U.S. Department of Defense, Jan. 2005.

[2] Alberts, D. S., Garstka, J. J., and Stein, F. P.
*Network Centric Warfare: Developing and Leveraging Information Superiority* (2nd ed.).
Washington, D.C.: DoD C4ISR Cooperative Research Program, 2000.

[3] Waltz, E. and Llinas, J.
*Multisensor Data Fusion.*
Norwood, MA: Artech House, 1990.

[4] Hall, D. and McMullen, S.
*Mathematical Techniques in Multisensor Data Fusion.*
Norwood, MA: Artech House, 2004.

[5] Network Centric Warfare: Department of Defense Report to Congress
http://cio-nii.defense.gov/docs/pt2_ncw_main.pdf,
July 2001.

[6] Sukkarieh, S., Nettleton, E., Kim, J. H., Ridley, M., Goktogan, A., and Durrant-Whyte, H.
The ANSER project: Data fusion across multiple uninhabited air vehicles.
*International Journal of Robotics Research*, **22**, 7 (July 2003), 505–539.

[7] Chong, C. Y., Mori, S., and Chang, KC
Distributed multitarget multisensor tracking.
In Y. Bar-Shalom (Ed.), *Multitarget-Multisensor Tracking: Advanced Applications*, Norwood, MA: Artech House, 1990, ch. 8.

[8] Martin, T. and Chang, KC
A distributed data fusion approach for mobile ad-hoc networks.
In *Proceedings of the 8th International Conference on Information Fusion*, Philadelphia, PA, July 2005.

[9] Martin, T. and Chang, KC
A data fusion formulation for decentralized estimation predictions under communications uncertainty.
In *Proceedings of the 9th International Conference on Information Fusion*, Florence, Italy, July 2006.

[10] Bar-Shalom, Y. and Blair, D.
*Multitarget Multisensor Tracking: Applications and Advances.*
vol. III, Norwood, MA: Artech House, 2000.

[11] Bar-Shalom, Y., Li, X-R., and Thiagalingam, K.
*Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software.*
John Wiley & Sons, 2001.

[12] Julier, S. J., Uhlmann, J. K., Walters, J., Mittu, R., and Palaniappan, K.
The challenge of scalable and distributed fusion of disparate sources of information.
In *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications* (SPIE), vol. 6242, Apr. 2006.

[13] Grime, S. and Durrant-Whyte, H.
Communication in decentralized systems.
In *IFAC Control Engineering Practice*, vol. 2, no. 5, Pergamon Press, 1994.

[14] Nicholson, D., Julier, S. J., and Uhlmann, J. K.
DDF: An evaluation of covariance intersection.
In *Proceedings of the 4th International Conference on Infarmation Fusion*, Montreal, Canada, vol. I, July 2001.

[15] Nicholson, D., Lloyd, C. M., Julier, S. J., and Uhlmann J. K.
Scalable distributed data fusion.
In *Proceedings of the 5th International Conference on Information Fusion*, Annapolis, MD, July 2002, 630–635.

[16] Bourgault, F. and Durrant-Whyte, H. F.
Communication in general decentralized filter and the coordinated search strategy.
In *Proceedings of the 7th International Conference on Information Fusion*, Stockholm, Sweden, July 2004.

[17] Cover, T. M. and Thomas, J. A.
*Elements of Information Theory.*
New York: Wiley, 1991.

[18] Hurley, M.
An information-theoretic justification for covariance intersection and its generalization.
In *Proceedings of the 5th International Conference on Information Fusion*, Annapolis, MD, July 2002.

[19] Julier, S. J.
An empirical study into the use of Chernoff information for robust, distributed fusion of Gaussian mixture models.
In *Proceedings of the 9th International Conference on Informatian Fusion*, Florence, Italy, July 2006.

**Kuo-Chu Chang** (F'10) received his M.S. and Ph.D. degrees in electrical engineering from the University of Connecticut in 1983 and 1986, respectively.

From 1983 to 1992, he was a senior research scientist in Advanced Decision Systems (ADS) Division, Booz-Allen & Hamilton, Mountain View, CA. In 1992, he joined the Systems Engineering and Operations Research Department, George Mason University where he is currently a professor. His research interests include estimation theory, optimization, signal processing, and multisensor data fusion. He is particularly interested in applying unconventional techniques in conventional decision and control systems. He has more than 25 years of industrial and academic experience and published more than 150 papers in the areas of multitarget tracking, distributed sensor fusion, and Bayesian network technologies. He was an associate editor on Tracking/Navigation Systems from 1993 to 1996 and on Large Scale Systems from 1996 to 2006 for *IEEE Transactions on Aerospace and Electronic Systems*. He was also an associate editor of *IEEE Transactions on Systems, Man, and Cybernetics*, Part A, from 2002 to 2007. He was the technical cochair for the 12th International Conference on Information Fusion (Fusion 2009) in Seattle.

Dr. Chang is a member of Etta Kappa Nu and Tau Beta Pi.

**Chee-Yee Chong** received his S.B., S.M., and Ph.D. degrees in electrical engineering from MIT.

He is chief scientist in the information fusion area for BAE Systems Advanced Information Technologies, formerly ALPHATECH. Prior to joining ALPHATECH, he was with Booz Allen Hamilton, which acquired Advanced Decision Systems (ADS), a small advanced research and development company in California. Before ADS, he was a professor at the Georgia Institute of Technology in Atlanta, GA. He has been involved in distributed fusion and sensor networks research for over 25 years. He developed one of the first fusion equations to remove double counting in networks and distributed multiple hypothesis tracking algorithms. His research interests include sensors, data and information fusion, and sensor resource management, both centralized and distributed versions; Bayesian networks, machine learning, and intelligence; and application of the above to real systems.

Dr. Chong was a cofounder the International Society of Information Fusion (ISIF), served as its president in 2004, and was the general cochair for the 12th International Conference on Information Fusion (Fusion 2009) in Seattle. He was an associate editor of the *IEEE Transactions on Automatic Control* and served on the editorial board of the *International Journal on Information Fusion*. He is currently an area editor for the *Journal of Advances in Information Fusion* (JAIF), the flagship journal for ISIF. He received the J. Mignogna Data Fusion Award from the U.S. JDL Data Fusion Group in 2005.

**Shozo Mori** obtained his B.S. degree in electric engineering from Waseda University, Tokyo, Japan; the M.S. degree in control engineering from Tokyo Institute of Technology, Tokyo, Japan; and the Ph.D. degree in engineering-economic systems from Stanford University, Stanford, CA.

He is a principal engineer at BAE Systems, Advanced Information Technologies Division, formerly ALPHATECH, in Los Altos, CA, office. Before joining ALPHATECH in 2005, he was with Information Extraction & Transport, Arlington, VA, Raytheon Company (Tiburon Systems), San Jose, CA, and Advanced Decision Systems (Advanced Information & Decision Systems), Mountain View, CA.

He is currently an area editor for *ISIF Journal of Advances in Information Fusion*, and an assistant editor for *IEEE Transactions on Systems, Man, and Cybernetics*, Part A. His current research interests include multiple-object tracking, particularly, as applications of theories of random finite sets, finite point processes, Bayesian networks, and distributed Bayesian detection, inference, and state estimation.

# *Scalable Fusion with Mixture Distributions in Sensor Networks*

KC Chang and Wei Sun
Dept. of Systems Engineering and Operations Research
George Mason University
Fairfax, VA 22030, USA

*Abstract* - **Mixture distributions such as Gaussian mixture model (GMM) have been used in many applications for dynamic state estimation. These applications include robotics, image and acoustic processing, distributed tracking, and multisensor data fusion. However, the recursive processing of the mixture distributions incurs rapidly growing computational requirements. In particular, the number of components in the mixture distribution grows exponentially when multiple of them are combined. In order to keep the computational complexity tractable, it is necessary to approximate a mixture distribution by a reduced one with fewer components. Mixture reduction is traditionally done by iteratively removing insignificantly components or merging similar ones. However, a systematic procedure is needed in order to ensure scalability while trading-off performance. In this paper, we propose a recursive mixture reduction algorithm for Gaussian mixture distribution with a given error bound. To meet the error bound, we applied a constraint optimized weight adaptation to minimize the integrated squared error (ISE) between the reduced distribution and the original one. With extensive simulations, we showed that the proposed algorithm provides an efficient and effective mixture reduction performance in distributed fusion applications.**

*Keywords* - **Gaussian mixture reduction, Constraint optimization, Integral squared error, distributed fusion, sensor networks.**

## I. Introduction

A mixture distribution is a combination of different probability density functions (pdfs). For example, Gaussian Mixture Model (GMM) is a special case of mixture distribution where a set of Gaussian pdfs are linearly combined. It is well known that GMM can be used to represent arbitrary probability densities to any desired accuracy. Due to this universal approximation property, GMM has been employed in many applications such as robotics [1], image processing [2], acoustic and speech recognition [3], multitarget tracking [4], distributed fusion [5], and Bayesian inference [6-7].

For instance, in content-based image retrieval (CBIR) systems the search could be based on criteria such as color, shape, texture or any such information. In such systems each semantic class can be represented by a Gaussian mixture model. When the query is made a template GMM is provided with the required characteristics. The distance between the reference and images in the database is then calculated to find the degree of similarity for retrieval [2][8]. Also in audio classification, the pdf of acoustic signal frequency spectrum is typically modeled by a Gaussian mixture model. A measure of similarity between a reference and a given sample is calculated by using a pre-defined distance metric in order to classify music [9]. Similarly, in distributed nonlinear tracking, a proper distance metric is defined to compare/correlate two tracks with mixture distributions [10].

However, most of these applications have to deal with the recursive processing of the mixture densities. For example, in multitarget tracking and fusion with distributed sensor networks, the "fusion" process is usually performed by multiplication of these densities [11]. While the product of Gaussian mixtures can be computed exactly, the number of components in the resulting mixture increases exponentially. In order to keep the computational and memory requirements bounded, it is essential to control this growth by approximating the resulting mixture with fewer components.

Several methods were developed recently to manage mixture reduction. Typically, the reduction is achieved by successively combining similar components or pruning away insignificant ones. For example, Salmond [12] proposed a joining and clustering algorithm for target tracking in clutter and West [13] proposed to collapse mixture components by replacing nearest neighboring components with merged component. Instead of repeatedly removing mixture components, another approach builds up the Gaussian mixture successively to approximate the original mixture [14-15]. Starting with a single Gaussian density, the algorithm proposed in [15] adds new Gaussian components to the approximate mixture by splitting existing components to provide better approximation.

In order to control and measure the performance of the mixture reduction, various similarity measures were proposed and employed in different algorithms. For instance, an Integral Square Error (ISE) based cost-function approach was developed to hypothesis control problem for multiple model tracking algorithms [16-17], and a Kullback-Leibler (KL) discrimination measure was used for the GM reduction [18].

In this paper, we first describe the distributed fusion problem with GMMs. We then examine several existing GMM reduction algorithms and develop a new approach by taking advantages of the state-of-the-art algorithms. The paper is organized as the follows. Section 2 describes the application of GMM in the distributed fusion problem, which is the one we are primarily interested in. Section 3 presents general Gaussian mixture reduction algorithms and our proposed approach. The simulation results are presented in Section 4 followed by some concluding remarks.

## II. DISTRIBUTED FUSION WITH GAUSSIAN MIXTURE

In a mixture model, a probability distribution is represented as a linear combination of basis funetions. Specifically, a Gaussian mixture model (GMM) can be expressed as,

$$f(x)=\sum_{i=1}^{N}\alpha_i N\left(x;\hat{x}_i,P_i\right) \tag{1}$$

wherc $\sum_{i=1}^{N}\alpha_i=1$ and $N\left(x;\hat{x}_i,P_i\right)$ is a Gaussian distribution eomponent with mean vector $\hat{x}_i$ and covariance matrix $P_i$.

In a distributed fusion problem, assuming two GMMs, $f_1(x)=\sum_{i=1}^{N_1}\alpha_{1i}N\left(x;\hat{x}_{1i},P_{1i}\right)$ and $f_2(x)=\sum_{i=1}^{N_2}\alpha_{2i}N\left(x;\hat{x}_{2i},P_{2i}\right)$ are to be fused with a common prior distribution, $f_3(x)=\sum_{i=1}^{N_3}\alpha_{3i}N\left(x;\hat{x}_{3i},P_{3i}\right)$.

With a standard fusion formula [11], the fused pdf can be obtained as,

$$f_F(x)=\frac{1}{c}\frac{f_1(x)f_2(x)}{f_3(x)} \tag{2}$$

where $c=\int\frac{f_1(x)f_2(x)}{f_3(x)}dx$ is a normalization constant. From (1) and (2), we have,

$$f_F(x)=\frac{1}{c}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\alpha_{1i}\alpha_{2j}\left(\frac{N\left(x;\hat{x}_{1i},P_{1i}\right)N\left(x;\hat{x}_{2j},P_{2j}\right)}{f_3(x)}\right) \tag{3}$$

where

$$c=\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\alpha_{1i}\alpha_{2j}\int\frac{N\left(x;\hat{x}_{1i},P_{1i}\right)N\left(x;\hat{x}_{2j},P_{2j}\right)}{f_3(x)}dx\equiv\sum_{i=1}^{N_2}\sum_{j=1}^{N_2}\alpha_{1i}\alpha_{2j}c_{ij} \tag{4}$$

is a normalization constant. In general, the integration in Equation (4) can not be obtained in elosed-form due to the mixture term in the denominator. To avoid the potential eomplexity using numerieal integration, one idea is to approximate the denominator, $f_3(x)$, with a single Gaussian pdf. Namely,

$$f_3(x)=\sum_{i=1}^{N_1}\alpha_{3i}N\left(x;\hat{x}_{3i},P_{3i}\right)\approx N\left(x;\hat{x}_3,P_3\right) \tag{5}$$

whcre $\hat{x}_3=\sum_{i=1}^{N_3}\alpha_{3i}\hat{x}_{3i}$ and $P_3=\sum_{i=1}^{N_3}\alpha_{3i}\left[P_{3i}+\left(\hat{x}_{3i}-\hat{x}_3\right)\left(\hat{x}_{3i}-\hat{x}_3\right)'\right]$.

With this approximation, the integration in (4) can be earried out analytieally and equation (3) can be rewritten as,

$$f_F(x)\approx\frac{1}{c}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\alpha_{1i}\alpha_{2j}c_{ij}\cdot N\left(x;\hat{x}_{ij},P_{ij}\right) \tag{6}$$

where $P_{ij}=\left[P_{1i}^{-1}+P_{2j}^{-1}-P_3^{-1}\right]^{-1}$ and $\hat{x}_{ij}=P_{ij}\left[P_{1i}^{-1}\hat{x}_{1i}+P_{2j}^{-1}\hat{x}_{2j}-P_3^{-1}\hat{x}_3\right]$.

Note that in the case when no common prior information was shared by the two distributions, equation (2) becomes,

$$\begin{aligned}f_F(x)&=\frac{1}{c}f_1(x)f_2(x)=\frac{1}{c}\sum_{i=1}^{N_1}\alpha_{1i}N\left(x;\hat{x}_{1i},P_{1i}\right)\sum_{j=1}^{N_2}\alpha_{2j}N\left(x;\hat{x}_{2j},P_{2j}\right)\\&=\frac{1}{c}\sum_{i=1}^{N_1}\sum_{j=1}^{N_1}\alpha_{1i}\alpha_{2j}N\left(x;\hat{x}_{1i},P_{1i}\right)N\left(x;\hat{x}_{2j},P_{2j}\right)\end{aligned} \tag{7}$$

where

$$c=\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\alpha_{1i}\alpha_{2j}\int N\left(x;\hat{x}_{1i},P_{1i}\right)N\left(x;\hat{x}_{2j},P_{2j}\right)d x\equiv\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\alpha_{1i}\alpha_{2j}c_{ij}$$

and $P_{ij}=\left[P_{1i}^{-1}+P_{2j}^{-1}\right]^{-1}$ and $\hat{x}_{ij}=P_{ij}\left[P_{1i}^{-1}\hat{x}_{1i}+P_{2j}^{-1}\hat{x}_{2j}\right]$.

With that, equation (7) can be rewritten as,

$$f_F(x)=\frac{1}{c}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\alpha_{1i}\alpha_{2j}c_{ij}\cdot N\left(x;\hat{x}_{ij},P_{ij}\right) \tag{8}$$

As one can see from both equations (6) and (8), thc fused probability density function has exponentially growing number of eomponents as morc GMs arc multiplied. To ensure sealability, it is necessary to managc the growth with a systematic and effective proeedure.

## III. GAUSSIAN MIXTURE REDUCTION

Given a Gaussian mixture distribution with $N$ eomponents, we wish to approximate it by a redueed one with $M$ components, where $M < N$. Traditionally, a mixture reduetion algorithm is reeursively eonducted sueh that the number of components is reduced by repeatedly choosing two components that appear to be most similar to eaeh other and merging them. For example, $K$-means algorithms and some variations can be applied to cluster Gaussian mixture components in groups, use a eenter component to represent all componcnts in each group, and then refine the parameters in the center eomponents basød on their members aecordingly.

West [13] proposed to eollapse mixture eomponents by simply replaeing nearest neighboring eomponents with a single merged eomponent. Thc basie routine proceeds as follows: First, loeate the eomponent with smallest weight. Then find another eomponent, whieh is the nearest neighbor of the selected one. Finally, merge the two components sueh that the resulting eomponent is the weighted average of thc two. The procedure is repeated until the desirable reduction of components is achieved.

### *A. Mixture Distance Metrics*

In general, there is no single bcst way to measure the distanec between two mixture distributions. There are a few distance metrics proposed in the literaturc. Williams [16] used integral squared error (ISE) as the similarity measure. Runnalls [18] used the Kullback-Leibler (KL) discrimination measure. In [10], we eompared several distanee metries for mixtures distributions. Speeifically, we foeus on the Integral Square Error (ISE) distance, the Bhattaeharyya distance, and the Kullback-Leibler distanee together with a general mixture distance (GMD) [19]. Among them, ISE is the most popular one due to its simplieity and closed forms expressions in Gaussian ease.

Specifically, the ISE distance between two probability distributions is defined as,

$$D_{ISE} = \int\left[f(\mathbf{x}) - \tilde{f}(\mathbf{x})\right]^2 d\mathbf{x} \tag{9}$$

When both $f(x)$ and $\tilde{f}(x)$ are Gaussian mixtures, equation (9) can be carried out in closed form [20]. Note that the ISE distancc we used in the simulation is the square root of the normalized version of the ISE distance, namely,

$$\tilde{D}_{ISE} = \sqrt{\frac{\int\left[f(\mathbf{x}) - \tilde{f}(\mathbf{x})\right]^2 d\mathbf{x}}{\int f(\mathbf{x})^2 d\mathbf{x} + \sqrt{\int \tilde{f}(\mathbf{x})^2}}} \tag{10}$$

The normalized distance varies from 0 to 1. $\tilde{D}_{ISE} = 0$ indicates a perfect match and $\tilde{D}_{ISE} = 1$ is the maximum possible distance. It can be shown that for two one-dimcnsional unit-variance Gaussian distributions with mean $\bar{x}_1$ and $\bar{x}_2$ respectively, the normalized ISE distance is $\tilde{D}_{ISE} = \sqrt{1 - e^{-\Delta x^2/4}}$, where $\Delta x = \left|\bar{x}_1 - \bar{x}_2\right|$. For example, a one STD merging distance threshold described in the next section would be $\gamma = \sqrt{1 - e^{-1/4}} \approx 0.47$.

### *B. GMM Reduction Algorithm*

In this section, we propose a GMM reduction algorithm based on a combination of the enhanced West/K-mean algorithm and a constraint optimized weight adaptation (COWA) algorithm. Specifically, with a pre-specified error bound $\varepsilon$, a minimum number of components $\kappa$, and a distance threshold $\gamma$, the algorithm consists of the following steps:

(1) Select a component to be merged

In the first step, a normalized (by the determinant of the covariance) weight of each component is obtained such that

$$w_i' = \frac{w_i}{\sqrt{\det(\mathbf{P}_i)}} \tag{11}$$

These normalized weights are used to order the GMM components such that the component with the smallest weight is selected as a candidate to be merged with another one that is closest to it in the ISE sense. However, the candidate component will be merged only when the closest neighbor is within a pre-defined distance threshold $\gamma$. This is to avoid a potential elimination of a "unique" isolated feature from the mixture distribution. If no qualified neighbor can be found for the current candidate, select the next one from the list with a larger weight until a qualified nearest neighbor is found.

(2) Merge the two selected components

The chosen candidate and its qualified closest component are merged based on the following linear combination rule,

$$w_{ij} = w_i + w_j$$

$$\mathbf{u}_{ij} = \lambda_i\mathbf{u}_i + \lambda_j\mathbf{u}_j$$

$$\mathbf{P}_{ij} = \lambda_i\mathbf{P}_i + \lambda_j\mathbf{P}_j + \lambda_i\lambda_j\left(\mathbf{u}_i - \mathbf{u}_j\right)\left(\mathbf{u}_i - \mathbf{u}_j\right)^{\mathrm{T}} \tag{12}$$

where $\lambda_i = w_i / w_{ij}$ and $\lambda_j = w_j / w_{ij}$.

(3) Apply constraint optimized weight adaptation

After each reduction step, apply the constraint optimized weight adaptation algorithm (COWA) [20] to adjust the weights of the reduced GMM components such that the ISE distance between the reduced GMM and the original one is minimized. The details of the adaptation algorithm arc presented in the next section.

(4) Repcat the above steps until either no more candidates can be found or the pre-specified stopping criterion is met. The stopping criterion states that either thc number of components reaches the goal $\kappa$ or the ISE distance meets the pre-determined threshold $\varepsilon$.

### *C. Constraint Optimized Weight Adaptation*

Suppose that we have initially located a Gaussian mixture of $K$ components to approximate an original Gaussian mixturc of $N$ components, where $K<N$. We now use constraint optimization method to adjust the $K$-componcnt weights to minimize the ISE from the original GMM. The optimization problem can be formulated as

$$\min \int_{\Omega}\left[\sum_{i=1}^{N}\alpha_i\mathbf{N}(\mathbf{x}\mid\mathbf{u}_i,\mathbf{P}_i) - \sum_{j=1}^{K}\beta_j\mathbf{N}(\mathbf{x}\mid\mathbf{u}_i',\mathbf{P}_i')\right]^2 d\mathbf{x} \tag{13}$$

$$s.t. \quad \sum_{j=1}^{K}\beta_j = 1$$

where $\mathbf{N}(\mathbf{x}\mid\mathbf{P}_i,\ _i)$ denotes the $i$-th component of the original multivariate Gaussian density with mean, $\mathbf{u}_i$, and covariance, $\mathbf{P}_i$. The weights satisfy $\sum_{i=1}^{N}\alpha_i = 1$. With a GMM rcduction process, the reduced GMM can be rcpresented ad,

$$\tilde{f}(\mathbf{x}) = \mathbf{P}\sum_{i=1}^{K}\beta_i \cdot \mathbf{N}(\mathbf{x}\mid\mathbf{u}_i',\ _i) \tag{14}$$

where $K<N$ and $\sum_{i=1}^{K}\beta_i = 1$. Our objectivc is to find the best set of weights $\{\beta_i\}$ to minimize ISE as shown in (13). Using the Lagrange formulation, we have

$$\left[\sum_{i=1}^{N}\alpha_i\mathbf{N}(\mathbf{x}\mid\mathbf{u}_i,\mathbf{P}_i) - \sum_{j=1}^{K}\beta_j\mathbf{N}(\mathbf{x}\mid\mathbf{u}_i',\mathbf{P}_i')\right]^2 + \lambda\left(\sum_{j=1}^{K}\beta_j - 1\right) \tag{15}$$

It has been shown that the optimal solution for $\{\beta_i\}$ can be derived in closed forms [20], specifically,

$$\mathbf{b}^* = \mathbf{H}^{-1}\mathbf{a} - \mathbf{H}^{-1}\mathbf{c}\left(\mathbf{c}^{\mathrm{T}}\mathbf{H}^{-1}\mathbf{a} - 1\right)\left(\mathbf{c}^{\mathrm{T}}\mathbf{H}^{-1}\mathbf{c}\right)^{-1} \tag{16}$$

where $\mathbf{c}^{\mathrm{T}} \equiv [1 \quad 1 \quad \cdots \quad 1]$, $\mathbf{b}^{\mathrm{T}} \equiv [\beta_1 \quad \beta_2 \quad \cdots \quad \beta_k]$

$$\mathbf{a}=\left[\sum_{i=1}^{N}\alpha_i\mathbf{N}\left(\mathbf{u}_1^{'}\,|\,\mathbf{u}_i,\mathbf{P}_1^{'}+\mathbf{P}_i\right)\ \cdots\ \sum_{i=1}^{N}\alpha_i\mathbf{N}\left(\mathbf{u}_k^{'}\,|\,\mathbf{u}_i,\mathbf{P}_k^{'}+\mathbf{P}_i\right)\right]^T \quad (17)$$

and

$$\mathbf{H}=\begin{bmatrix}\mathbf{N}\left(\mathbf{u}_1^{'}\,|\,\mathbf{u}_1^{'}.2\mathbf{P}_1^{'}\right) & \cdots & \mathbf{N}\left(\mathbf{u}_1^{'}\,|\,\mathbf{u}_k^{'},\mathbf{P}_1^{'}+\mathbf{P}_k^{'}\right)\\ \vdots & \ddots & \vdots\\ \mathbf{N}\left(\mathbf{u}_k^{'}\,|\,\mathbf{u}_1^{'},\mathbf{P}_1^{'}+\mathbf{P}_k^{'}\right) & \cdots & \mathbf{N}\left(\mathbf{u}_k^{'}\,|\,\mathbf{u}_k^{'},2\mathbf{P}_k^{'}\right)\end{bmatrix} \quad (18)$$

## IV. Test and Evaluation

To test the algorithm, we simulate a network with a number of cooperating sensors. Each sensor is assumed to observe an object in a one-dimensional (1D) or two-dimensional (2D) space and produce a mixture distribution representing its estimate of the object state. The sensors communicate their estimates with each other in a sequential manner where each sensor node is responsible to "fuse" the incoming estimates with its own estimate and pass the resulting fused estimate to the next node. For example, for a network with $n$ nodes, suppose each sensor has a local state estimate represented by a GMM of $m$ components. After a sequence of communication and fusion (sensor 1 sends its estimate to sensor 2, sensor 2 combines the estimates and sends the fused results to sensor 3, etc.), the total number of components of the resulting GMM at the end of the process will be $m^n$ which is clearly not desirable.

To ensure scalability, we apply the algorithm described in Section 3 to "compress" the combined mixture distribution before forwarding it to the next node. In order to meet the accuracy requirement, a pre-defined error bound in terms of ISE distance is given so that the reduced GMM is guaranteed to be within the specified distance to the original GMM at the end of each fusion step.

### *A. Scenario I*

In the first scenario, we simulate a network of eight sensors each estimating a one-dimensional target state with a two-component GMM. As mentioned before, the communications are taken place in a sequential manner where each sensor participates exactly once at a particular order. Without the reduction process, at the $n^{\text{th}}$ stage of the communication chain, the resulting GMM will have $2^n$ components. It is the result of the product of $n$ GMMs each with 2 components and will serve as the ground truth to compare the reduced GMMs.

For a sample trial, Figure 1 shows the local estimates (GMM and its components) from the eight sensors before fusion. After the sequential fusion, the resulting true fused GMMs and the corresponding reduced GMMs together with their components are shown in Figure 2. In the trial, the simulation parameters were set to be $\varepsilon=0.01$, $\kappa=1$, and $\gamma=0.47$. At the end of the chain (sensor node 8, bottom-right of Figure 2), the reduced (fused) GMM requires only 5 components and it is less than 1% away (in the ISE sense) from the true GMM consisting of $2^8=256$ components.

We test the scenario with 100 Monte Carlo trials with similar parameters and the results are shown in Figures 3-4. As can be seen, with $\varepsilon=0.01$ or $\varepsilon=0.001$, the errors are well within the bounds and the computations are relatively scalable. When $\varepsilon=0.0001$, the complexity increases slightly more than linearly to the network size while the accuracy is still well within the bound.

### *B. Scenario II*

In this scenario, we simulate a network of eight sensors each estimating a 2D target state with a two-component GMM. As in scenario 1, the communications are taken place in a sequential manner. For a sample trial, Figure 5 shows the local estimates (GMM and its components) from the eight sensors. The resulting true fused GMMs and the corresponding reduced GMMs together with their components are shown in Figures 6 and 7 respectively.

The results with 100 Monte Carlo trials shown in Figures 8-9 are very similar to the ones in Figures 3-4. It also shows that a trade-off between performance and complexity could be achieved by selecting a proper operating point (error bound) at each local reduction step.

## V. Summary

This paper presents a method to approximate a Gaussian mixture by a smaller one with fewer components. The method ensures that the ISE error between the original GM and its approximation is smaller than a predefined threshold with a minimum number of components. We also show empirically that the cumulated error, after compressing and fusion, is somewhat bounded. This is important for controlling the trade-off between system performance and scalability particularly for distributed estimation in a large sensor networks. We conducted extensive tests with a distributed fusion scenario. The simulation results demonstrate the validity and scalability of the algorithm. The results also suggest a simple approach to control the trade-off between the performance and the complexity.

To ensure scalability and understand the theoretical performance bounds, one important future research direction is to analyze the propagation of the local error bounds over multiple fusion steps and to conduct the convergence analysis of the algorithm. In addition to distributed fusion, another potential application of the algorithm is for probabilistic inference in hybrid Bayesian Networks as described in [6-7][21]. In these networks, messages in terms of mixture distributions are propagated between discrete and continuous nodes. The inference process involves multiplication of multiple mixture densities. Further research along this direction is critical in order to manage the complexity of probabilistic inference in hybrid dynamic Bayesian networks.

## VI. Acknowledgement

The research was partially supported by the National Science Council of the Republic of China (NSC#99-2811-E-002-097).

## VII. References

[1] Marco Huber and Uwe Hanebeck, "Hybrid Transition Density Approximation for Efficient Recursive Prediction of Nonlinear Dynamic Systems," *IPSN'07*, April 2007, Cambridge, MA, USA.

[2] Yossi Ruhner , Carlo Tomasi , Leonidas J. Guibas, A Metric for Distributions with Applications to Image Databases, Proceedings of the 6th International Conference on Computer Vision, January 04-07, 1998.

[3] Yue Pan and Alex Waibel, "Minimum Kullback-Leibler Distance based Multivariate Gaussian Feature Adaptation for Distant-Talking Speech Recognition," in Proc. Acoustics, Speech, and Signal Processing, 2004.

[4] Y. Bar-Shalom and T. E. Fortmann, Tracking and Data Association, Academic Press, 1988.

[5] Simon J. Julier, "An Empirical Study into the Use of Chernoff Information for Rohust, Distributed Fusion of Gaussian Mixture Models," in proc. of the 9th international conf. on Inf. Fusion, 2006.

[6] Prakash Shenoy, "Inference in Hybrid Bayesian Networks using Mixtures of Gaussians," Proc. UAI 2006.

[7] Wei Sun and KC Chang, "Direct Message Passing for Hybrid Bayesian Networks and Performance Analysis," in Proc. SPIE Defense and Security Symposium, Orlando, Florida, April, 2010.

[8] H. Greenspan, A. T. Pinhas, "Medical image categorization and retrieval for PACS using the GMM-KL framework." in IEEE Trans Inf Technol Biomed, Vol. 11, No. 2. (March 2007), pp. 190-202.

[9] B. Logan and A. Salomon, "A music similarity function hased on signal analysis," in Proc. IEEE Int. Conf. Multimedia Expo, 2001, pp.745–748.

[10] Ashirvad Naik and KC chang, "A Comparison of Distance Metrics between Mixture Distributions," in Proc. SPIE Defense and Security Symposium, Orlando, Florida, April, 2010.

[11] KC Chang, Chee-Yee Chong, and Shozo Mori, "Analytical and Computational Evaluation of Scalahle Distributed Fusion Algorithms," to appear in *IEEE Trans. on Aerospace and Electronic Systems*, 2010.

[12] D. J. Salmond, "Mixture reduction algorithms for target tracking," in IEE Colloquium on State Estimation in Aerospace and Tracking Applications, London, UK, Dec. 1989, pp. 7/1–7/4.

[13] W. West, "Approximating Posterior Distrihutions hy Mixture", Journal of the Royal Stat Society, Series B, Vol. 55, No. 2, pp. 409-422, 1993.

[14] O. C. Schrempf, O. Feiermann, and U. D. Hanebeck, "Optimal Mixture Reduction of the Product of Mixtures," in Proc. of the 8th International Conference on Information Fusion, vol. 1, Jul. 2005, pp. 85–92.

[15] Marco Huber and Uwe Hanebeck, "Progressive Gaussian Mixture Reduction," in proc/ of the 11th international conf on Inf. Fusion, 2008.

[16] J. L. Williams and P. S. Maybeck, "Cost-Function-Based Hypothesis Control Techniques for Multiple Hypothesis Tracking", in proc. of the 6th international conf on Information Fusion, Vol.2, pp.1047-1054, 2003.

[17] P. S. Maybeck and Brian Smith, "Multiple Model Tracker based on Gaussian Mixture Reduction for Maneuvering Targets in Clutter," in proc. of the 8th international conference on Information Fusion, 2005.

[18] A. R. Runnalls, "Kullback-Leibler Approach to Gaussian Mixture Reduction", IEEE Transactions on AES, Vol.43, No.3, July 2007.

[19] Z. Liu, Q. Huang, "A new distance measure for probability distribution function of mixture type," ICASSP, vol. 1, pp.616-619, 2000.

[20] HD Chen, KC Chang, and Christ Smith, "Constraint Optimized Weight Adaptation for Gaussian Mixture Reduction," in Proc. SPIE Defense and Security Symposium, Orlando, Florida, April, 2010.

[21] O. C. Schrempf and U. D. Hanebeck, "A New Approach for Hyhrid Bayesian Networks Using Full Densities," in Proc. of 6th Workshop on Computer Science and Information Tech, Budapest, Hungary, 2004.

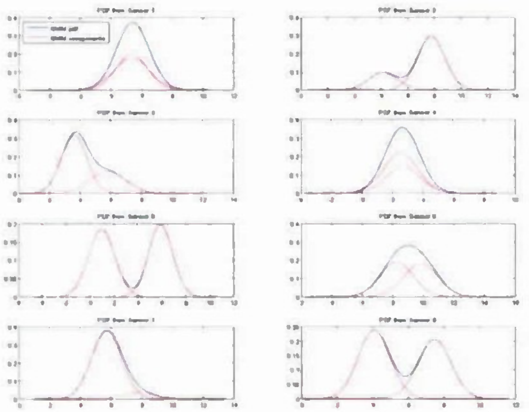


Figure 1. 1D Local Sensor Estimates

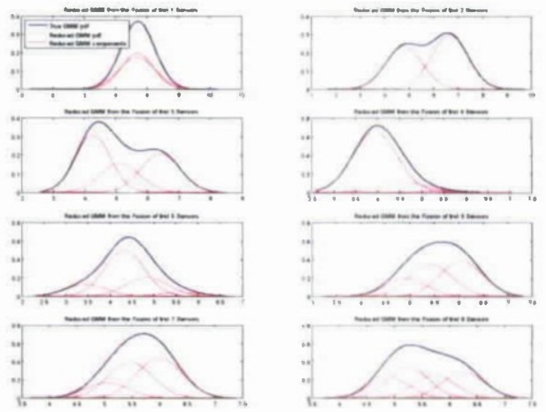


Figure 2. 1D Sequential Fused Estimates

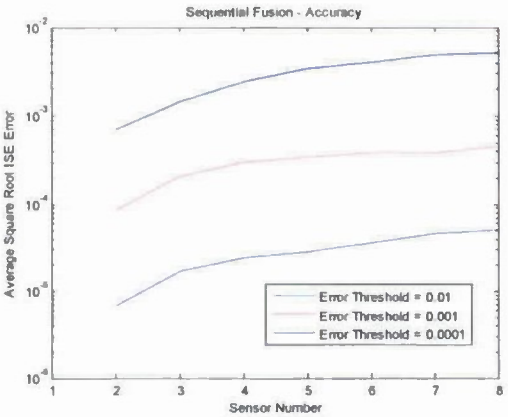


Figure 3. 1D Average ISE Error with Centralized Fusion

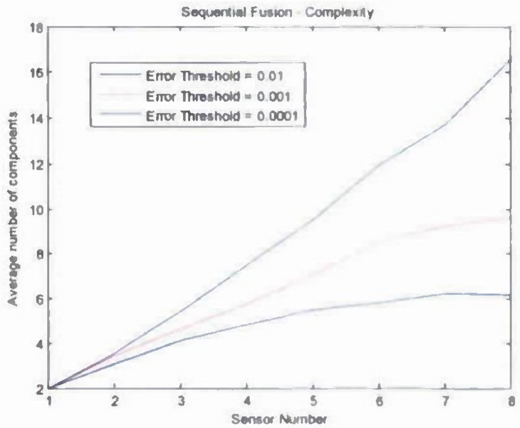


Figure 4. 1D Avg. # of Components with Centralized Fusion

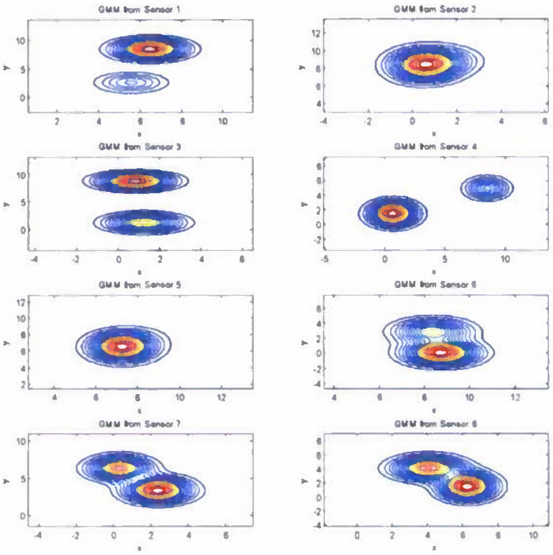


Figure 5. Local Sensor GMMS

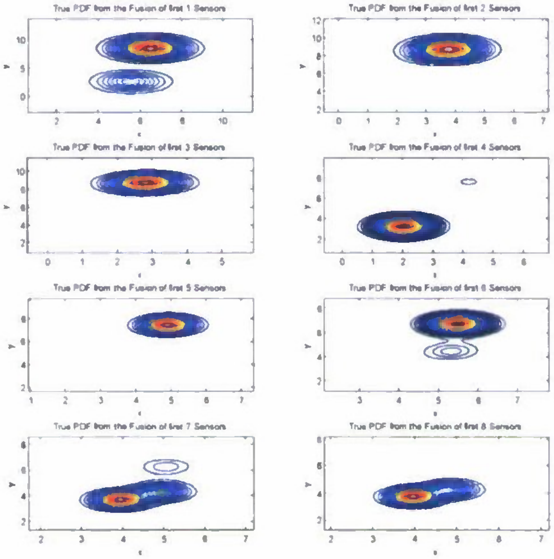


Figure 6. Sequential Fused Estimates – True GMMs

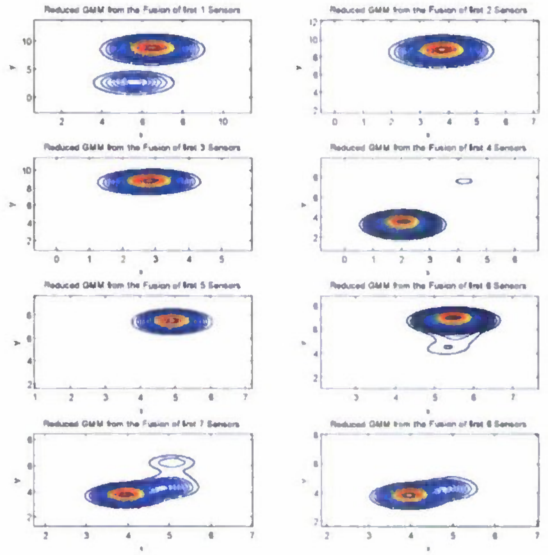


Figure 7. Sequential Fused Estimates – Reduced GMMs

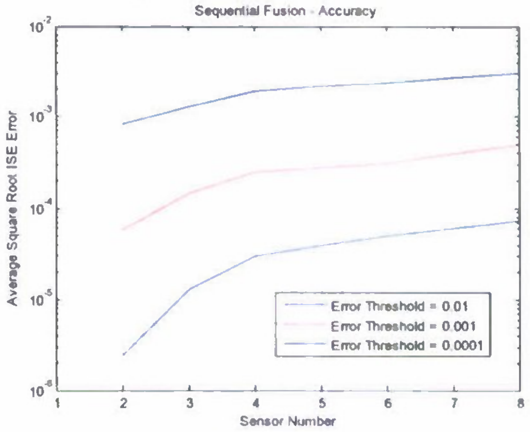


Figure 8. 2D Average ISE Error with Centralized Fusion

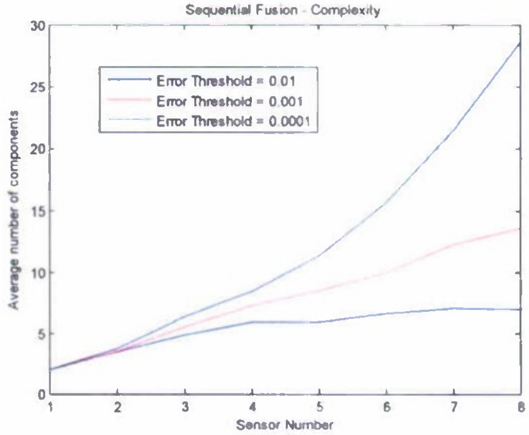


Figure 9. 2D Avg. # of Components with Centralized Fusion

# Fusion and Gaussian Mixture Based Classifiers for SONAR Data

Vikas Kotari [a] and KC Chang[a]

[a]*Dept. of SEOR, George Mason University, 4400 University Dr., Fairfax, VA 22030*

## ABSTRACT

Underwater mines are inexpensive and highly effective weapons. They are difficult to detect and classify. Hence detection and classification of underwater mines is essential for the safety of naval vessels. This necessitates a formulation of highly efficient classifiers and detection techniques. Current techniques primarily focus on signals from one source. Data fusion is known to increase the accuracy of detection and classification. In this paper, we formulated a fusion-based classifier and a Gaussian mixture model (GMM) based classifier for classification of underwater mines. The emphasis has been on sound navigation and ranging (SONAR) signals due to their extensive use in current naval operations. The classifiers have been tested on real SONAR data obtained from University of California Irvine (UCI) repository. The performance of both GMM based classifier and fusion based classifier clearly demonstrate their superior classification accuracy over conventional single source cases and validate our approach.

**Keywords:** Data Fusion, Gaussian Mixture model, SONAR, detection and classification

## 1. INTRODUCTION

Protecting a nation's ocean border is very important for its defense. A nation's oceans can be attacked in a variety of ways of which naval mines are the easiest. Underwater mines can easily flood oceans. Since 1950, naval mines have been responsible for more ship causalities on US fleet than all other threats combined [1]. They have also been accounted for damage to local economies, marine life, and sailor life. These underwater mines are very inexpensive to acquire and deploy yet highly destructive. Underwater mines come in variety of types such as, bottom mines, shallow mines, and magnetic mines. Irrespective of the type their lethality is high. This combined with the difficulty in detecting and classifying them makes them highly effective. This necessitates a formulation of efficient classifiers and detection techniques. Most prevalent methods of classification focus on signals from single source, such as learned classification using massively parallel networks [2], MML inference of oblique decision trees [3], and second order cone programming approach [4]. Although some data fusion based methods, such as algorithm fusion [5] and computer aided detection and fusion [6] have been proposed, the application of these methods is limited due to their complexity. Each of the aforementioned methods has limitations in terms of accuracy and applicability.

Data fusion is traditionally applied for command and control operations. However recently data fusion techniques are being employed for classification purposes [7]. Similarly, Gaussian mixtures are well established methods that have been extensively used in speech recognition and other classification applications [8]. But their use in underwater mine classification is limited.

In this paper, we propose a fusion based classifier and a Gaussian mixture based classifier. These classifiers have been applied to SONAR data due to its extensive use in practice. The goal is to investigate the proposed classifiers and to evaluate its performance. The remaining of the paper is organized as follows. Section 2 describes briefly about the SONAR data we will be testing and describes some initial classification analysis. Section 3 presents the data fusion classifiers and Section 4 explains the Gaussian mixture classifiers. Section 5 presents the performance results in terms of receiver operator characteristic (ROC). Section 6 summarizes our study and proposes some future research directions.

## 2. INITIAL DATA ANALYSIS

Almost all naval vessels use SONAR systems extensively. This is attributed to their cost effectiveness, detection accuracy and ease of use. Therefore our focus will be on classification with these SONAR signals. To this end we obtained data from UCI repository. This data consisted of 208 returns from a SONAR system. Of the 208 returns 111 were returns bouncing off from a mine at different depths and angles and 97 were from a rock. The aspect angles of the signals varied from 90 degrees to180 degrees. The signal strength in the data represents the energy at a particular frequency band. Each data set consists of a vector of 60 elements [9]. The data are represented in figure 1 and figure 2 below. Figure 1 presents the signal received from mine detection Figure 2 presents the signal from rock detection. Figure 3 shows the high similarity between the rock and mine templates created from the average of the signals. To understand the data quality, a simple but efficient nearest neighbor method using Euclidean distance was used to determine the Bayesian bound. Euclidean distance is a special case of $L_k$ norm, where k =2. For a d-dimensional space, the $L_k$ norm is defined as,

$$L_k(x,y) = \sum_{i=1}^{d}(\|x^i - y^i\|^k)^{1/k} \tag{1}$$

It was observed from the SONAR data set that the classification accuracy was 82.69% based on nearest neighbor. Therefore, the Bayesian performance bound would be $B_e < 1 - \frac{1}{2}(1 - 0.8269) = 0.9134$ = 91.34% [10].

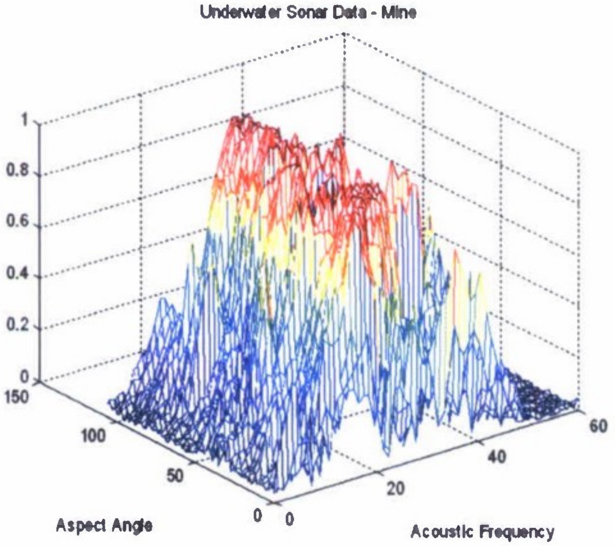


Figure 1. Signal for Mine Detection

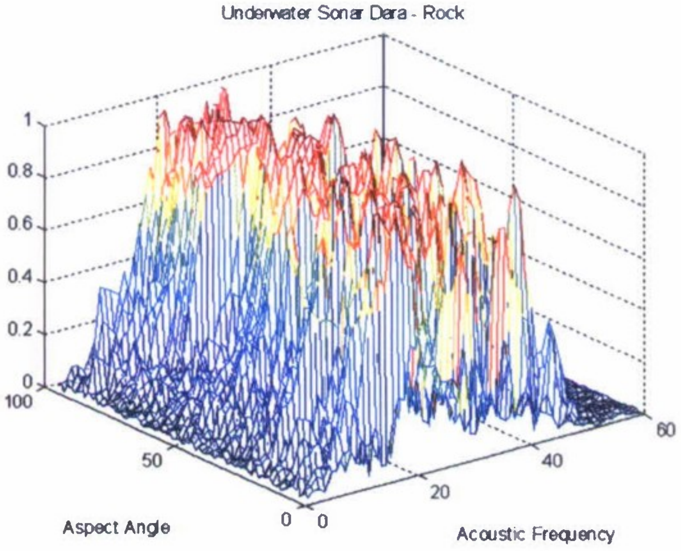


Figure 2. Signal for Rock Detection

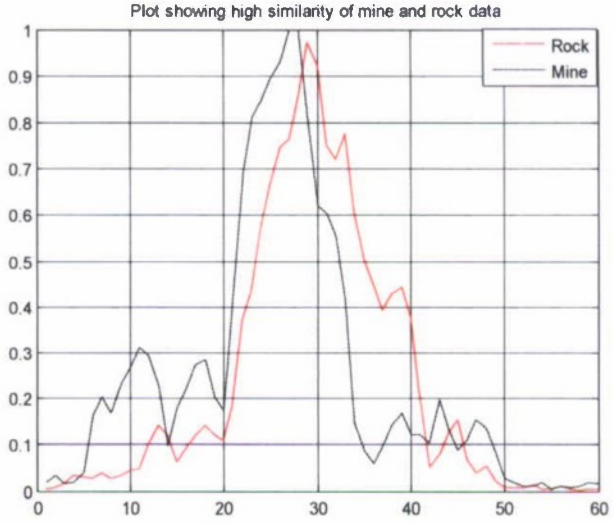


Figure 3. Mine and Rock Signal Templates

After the Bayesian bound analysis, initial classification was performed on the entire dataset by dividing the data into training and testing datasets. Among the 208 data samples all the even samples were treated as training datasets and odd samples as testing datasets. After which every single data point from the testing case was compared to every other data point in the training data set. The Euclidean (L2) distance was initially used to determine the k-nearest neighbor (kNN). Based on the similarity of a sample with the training data set, it was determined if the neighbor was either a rock or a mine. This process of classification was repeated for different values of k (i.e., number of neighbors). Due to limited number of data samples, this approach resulted in relatively poor performance. We then repeated the analysis by converting the data to GMMs, which is detailed in section 4. For GMM, the Euclidean distance was replaced by integral square error (ISE) distance. The Integral Square Error (ISE) distance is defined as [11]:

$$Js = \int (g(x)-h(x))^2 dx = \int\{g^2(x)-2g(x)h(x)+h^2(x)\, dx\} \qquad (2)$$

where $g(x)$ and $h(x)$ represent two density functions. For GMM, the ISE distance can be obtained in close form [11]. The GMM method resulted in an increased accuracy of 85.09%. It was observed that the GMM method performs superior to the conventional kNN classifier based on Euclidean distance.

## 3. DATA FUSION BASED CLASSIFIERS

Data fusion systems combine data from multiple sources/sensors to improve situation assessment. This is done to increase accuracies and achieve better inferences than achieved by a single sensor alone. Historically, developed for command control communication and intelligence applications, they are currently finding a plethora of possibilities in areas ranging from manufacturing to medicine. One such application is in classification. Fusion based classifiers were developed because of their advantages over single source based classifiers. Some of the advantages of data fusion are robust operational performance, increased spatial and temporal coverage, increased confidence, and reduced ambiguity [12]. The data fusion approach in this work was applied in a two-fold manner, wherein the first approach was to combine data and the other was to combine decision.

In the first approach, we form an augmented data vector by combining each data with every other set of data from the same source to emulate data received from a 2-sensor scenario. The combination process resulted in a significant number of *synthetic* data set. In each data set, the number of elements increases from 60 to 120. Each of these 120 element vector represents either a rock or a mine, detected by 2 SONAR sensors. This process is to simulate a centralized 2 sensor fusion scenario. The combined data was tested and the nearest neighbor was found over the entire data to determine the accuracy. Because of the enhanced performance due to sensor fusion, the accuracy had substantially increased to 94.34% and the corresponding Bayesian bound also increased to 97.17%. Employing the kNN based approach described in section 2 for the data fusion case, the performance of the classifier also increased significantly. This can evidently be attributed to the advantages of the sensor fusion approach.

Although the accuracy was substantially superior for data fusion approach, the communication requirements for the centralized data fusion case were high. Another approach is to use decision fusion based classifiers. In decision fusion, instead of relaying all the 120 bits (each sensor contributes 60 bits of data) of the data, only one bit indicating (decision) whether the data is rock or mine is communicated and fused. This reduces the bandwidth by more than 99%. The XOR, the OR and, the majority vote fusion rules were tested. For the OR rule, as long as one of the two sensors classifies the object as a mine, the object was classified as a mine. The accuracy in this case was about 77.59 with L2 distance. As expected the XOR rule performed inferior to the OR rule. This is because XOR classified a return as a rock even if one sensor called the object the rock. The accuracy with the same L2 case reduced to 64.6%. The majority fusion rule considered 3 sensors and used 2/3-majority vote, which produced best overall accuracy of 88.03% as expected. It is clear that the decision fusion approach would perform worse than the centralized fusion case due to significantly reduced data quantity. However, when communication bandwidth is paramount to the system, then decision fusion could be a good alternative.

## 4. GAUSSIAN MIXTURE BASED CLASSIFIERS

Gaussian Mixture Model (GMM) is a special case of mixture distributions where a set of Gaussian densities is linearly combined. Mixture distributions subsist in many applications, such as speech recognition, image retrieval, nonlinear filtering, and target tracking [8][11][13][14]. Gaussian mixture model (GMM) is typically used in classification applications to model the probability density function (PDF) of a signal's frequency spectrum. A similarity measure is calculated with respect to a reference sample to classify data. It is therefore natural to formulate a GMM based classifier for SONAR data.

The acquired UCI data was first converted to GMMs using expectation maximization (EM) method. Once the data was converted to Gaussian mixtures the classifier was trained. The integral square distance (ISE) was used to measure the similarity between the data sample.

It was observed that the classification performance based on ISE distance is superior to other distance metrics in a high signal to noise ratio cases [15]. For the case of 60 (full data size) terms GMM, the accuracy was 85.09% with the Bayesian bound increased to 92.51%. To test the tradeoff between complexity and performance, we reduced the number of terms in GMM from 60 to 20 and lower. It was observed that with 20 terms, the accuracy reduced to 83.65%, which is still better than the performance based on the original data using L2 norm distance. We applied the similar fusion method described in the previous section to the GMM data. It was observed that the accuracy improved to 92.59%.

We also tested the decision fusion performance based on the GMM converted data. ISE distance metric was used with nearest neighbor approach analogous to the previous case. The accuracy values for OR, XOR and majority vote fusion rule were 79.22%, 67.16%, 89.88% respectively. Again, they perform slightly better than the case with the original data described in Section 3.

## 5. ROC CURVE ANALYSIS

Designed initially for RADAR systems, the receiver operation characteristic curve is a standard metric to measure a classifier's performance. These curves were obtained by varying the detection threshold to observe the tradeoff between probability of detection and probability of false alarm. With the original data, the resulting ROC curves for the single sensor and two-sensor centralized fusion cases are shown in figure 4. It is evident that the two-sensor case performs significantly better than the single sensor case.

To test the trade-off between communication requirements and performance for the centralized fusion, we lowered the communication requirements by transmitting only partial data. The results are presented in Figure 5. It can be observed that with 1/3$^{rd}$ of the data transmitted (every third data point, a total of 20 data points for each sensor observation), the classification performance was only slightly worse than the one with full data rate. Similarly it can be seen that when only 1/12$^{th}$ of the data points from each sensor were transmitted for fusion, the performance was much poorer but was comparable to the single sensor case.

Similar analysis was performed based on GMM approach with the ISE distance. The ROC curves were generated for two-sensor case for both GMM and the original data. The plot showing the ROC curves for both ISE and L2 distance case are presented in Figure 6. It can be seen from the figure that the performance of both approaches are comparable although the GMM approach works marginally better at lower detection thresholds (higher false alarm rates).

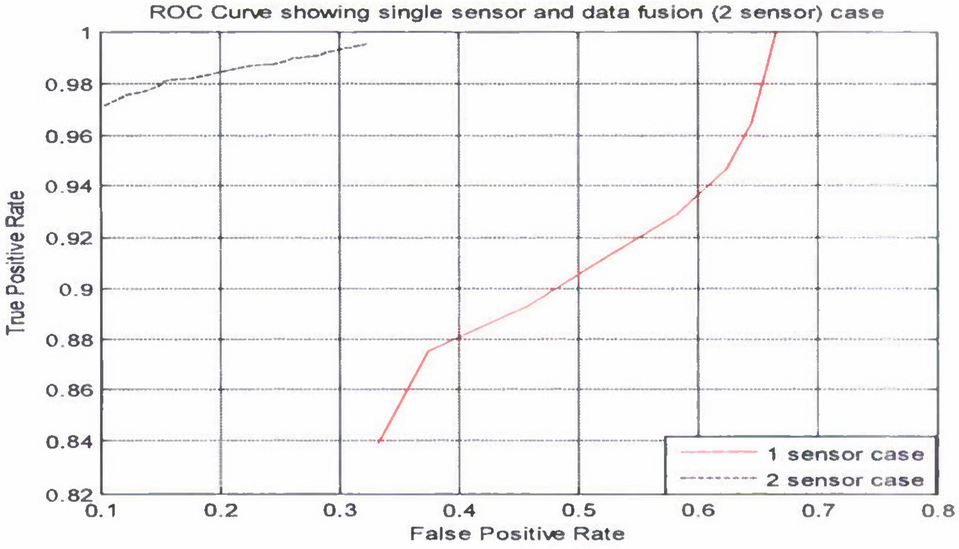


Figure 4. ROC curve for two-sensor case and single-sensor case

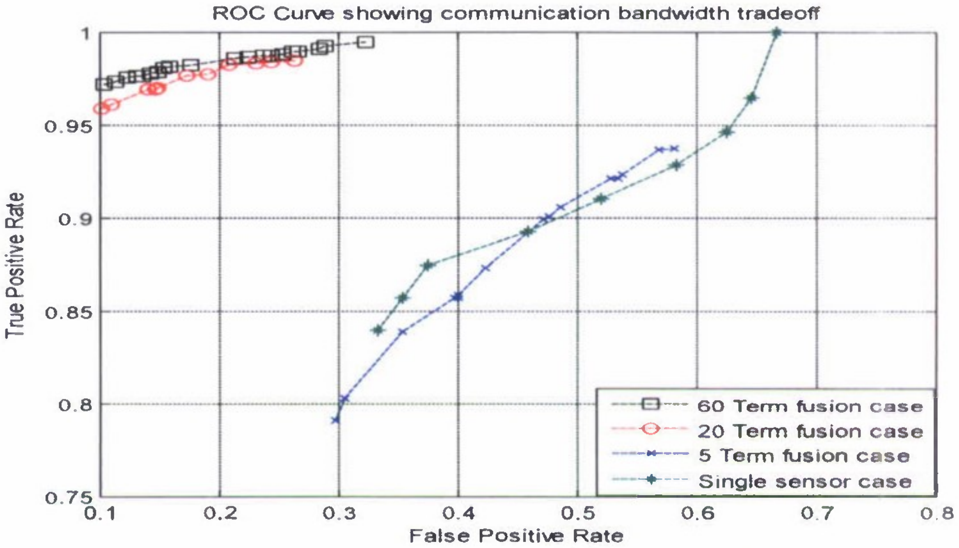


Figure 5. ROC curve showing tradeoff between communication bandwidth and performance.

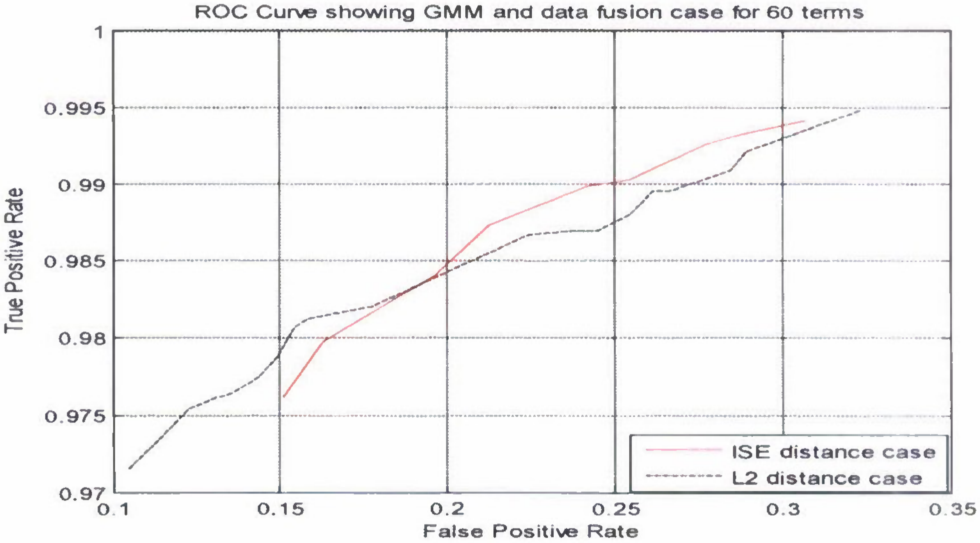


Figure 6. ROC curve showing ISE distance and L2 distance methods

The majority of this study has focused on using Euclidean distance, which uses the L2 distance. It has been shown that for high dimensional data, the performance improves with reduced order of the distance (L) [16][17]. We tested the effectiveness of the classifiers by varying the distance parameter of L distance. The results are presented in Figure 7. It can be seen from the figure that while the classification performance does not change too much between L1-norm to L10-norm distance, a significant jump in performance was observed when the distance measure goes from L1 to fractional distance. This is consistent with the observation in [18] where the fractional distance provides significant performance improvements for high dimensional data over Manhattan distance (L1) and Euclidean distance (L2). We then varied the values of L-norm distance metric to obtain the ROC curves. Figure 8 presents the ROC curves with L values varying from 0.5 to 5 for single sensor case and Figure 9 presents similar ROC curves for two sensor centralized fusion case. It is interesting to observe that as the values of L decreased from 5 to 0.5 the performance of the classifier improved greatly.

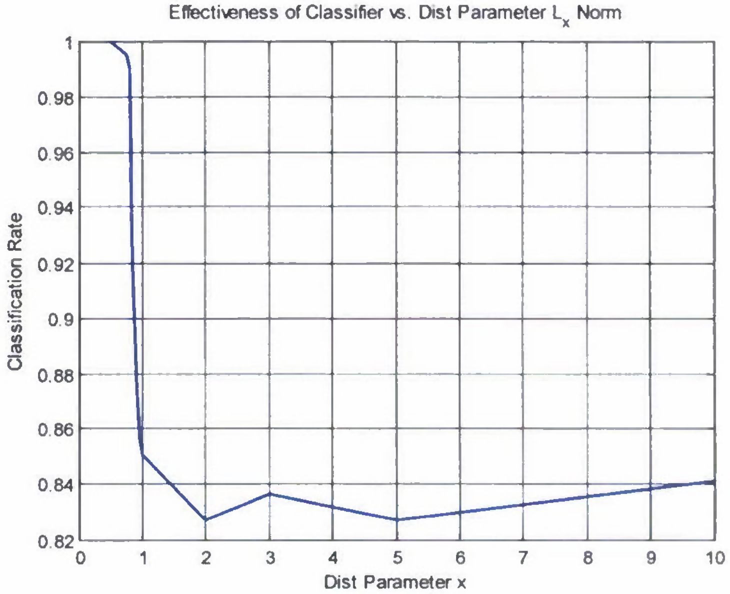


Figure 7. The effectiveness of classifier for varying L values

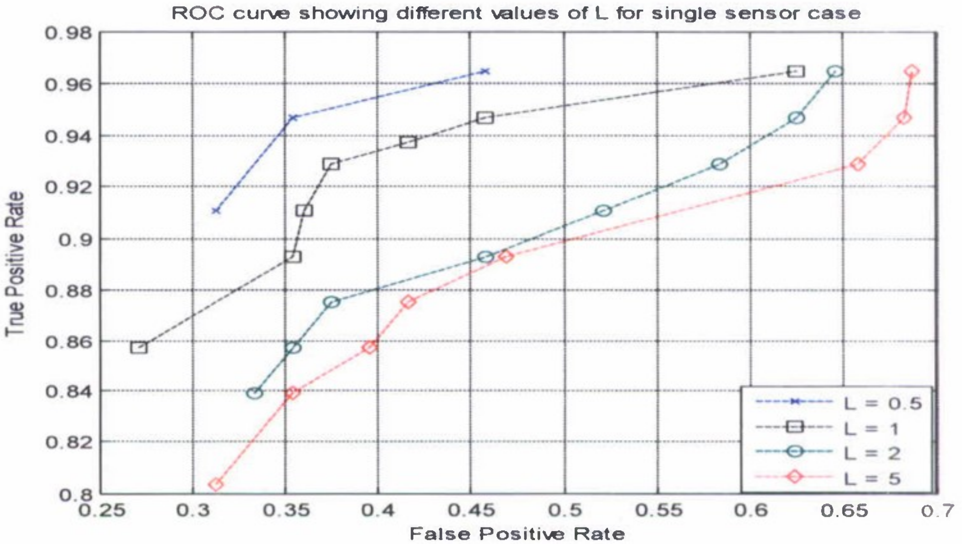


Figure 8. ROC curve showing varying L values for single sensor casc.

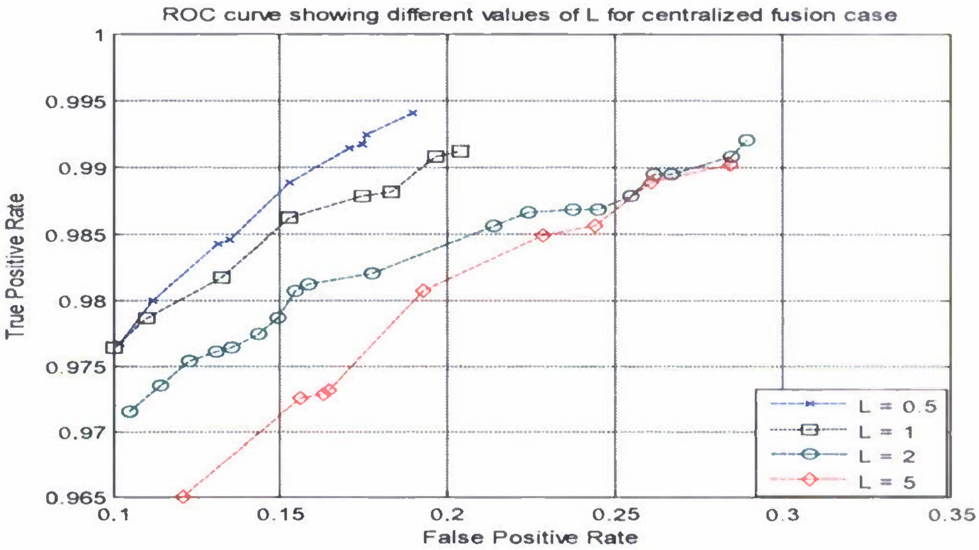


Figure 9. ROC curve showing varying L values for centralized fusion case.

## 6. SUMMARY

We focus on formulation of fusion based and GMM based classifiers for application on SONAR signals. We perform extensive simulations to test the validity of our approach. In our analysis we have observed that GMM based nearest neighbor classifiers using ISE distance metric perform analogous to conventional Euclidean distance metric nearest neighbor classifiers. We have also observed that the performance of multi-sensor based centralized fusion classifiers is superior to single source methods. Since the communication bandwidth requirements for the centralized fusion based classifier is very high, compressed versions of the data with or without GMM can be communicated and classification can be performed with reduced bandwidth. We have observed that this data compression approach limits the communication bandwidth usage while performs superior to the conventional nearest neighbor methods with the single sensor data. When the communication bandwidth is extremely limited, we propose to fuse decisions in place of data. The simulation results showed that the decision fusion approach could be highly efficient with some performance degradation. We have also tested the influence of various L-norm distances on this high dimensional data and observed that fractional L-norm distance perform superior to higher order L-norm distance.

One of the future research directions is to test the applicability of the classifiers with data from various other types of sensor system. They include SONAR systems, video systems, and other imaging techniques. Another research direction is to perform communication bandwidth tradeoff studies on fusion performance with multi-modality data. We also intend to investigate the possibility of identifying the optimal L-norm distance for various types of high dimensional data.

## REFERENCES

[1] U.S Navy Marine Mammal Program, U.S. Navy Marine Mammal Mine Hunting Systems. http://www.spawar.navy.mil/sandiego/technology/mammals/mine_hunting.html .

[2] R. Paul Gorman and Terrence J. Sejnowski, "*Learned Classification of Sonar Targets Using a Massively Parallel Network*" IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. 36, NO. 7,JULY 1988 .

[3] Jianbin Tan and David L. Dowe, "*MML Inference of Oblique Decision Trees*". Australian Conference on Artificial Intelligence. 2004.

[4] Chiranjib Bhattacharyya, "*Robust Classification of noisy data using Second Order Cone Programming approach*". Proceedings of International Conference on Intelligent Sensing and Information Processing, 2004.

[5] Gerald J. Dobeck, "*Algorithm Fusion for Automated Sea Mine Detection and Classification*". OCEANS, 2001. MTS/IEEE Conference and Exhibition.

[6] Charles M. Ciany and, Jim Huang, "*Computer Aided Detection/Computer Aided Classification and Data Fusion Algorithms for Automated Detection and Classification of Underwater Mines*". OCEANS 2000 MTS/IEEE Conference and Exhibition.

[7] Dong, J.; Zhuang, D.F.; Huang, Y.H.; Fu, J.Y. "*Advances in multi-sensor data fusion: Algorithms and applications.*" *Sensors* 2009, *9*, 7771-7784

[8] B. Logan and A. Salomon, "*A music similarity function based on signal analysis*," in Proc. IEEE Int. Conf. Multimedia Expo, 2001, pp. 745 – 748.

[9] Blake, C.L., Merz, C.J, UCI Repository of Machine Learning Databases Irvine, CA, University of California, Department of Information and Computer Science (1998) http://www.ics.uci.edu/~mlearn/MLRepository.html.

[10] Richard Duda, Peter Hard, and David Stork, "*Pattern Classification*", Wiley, 2001 (p. 180).

[11] Peter S. Maybeck and Brian D. Smith, "*Multiple Model Tracker Based on Gaussian Mixture Reduction for Maneuvering Targets in Clutter*" in International Conference on Information Fusion (FUSION), 2005.

[12] E. Waltz, "*Data fusion for $C^3I$: A tutorial*," in Command and Control, Communications Intelligence ($C^3I$) Handbook. Palo Alto, CA: EW Communications, 1986, pp.217-226.

[13] H. Greenspan, A. T. Pinhas, "*Medical image categorization and retrieval for PACS using the GMM-KL framework.*" in IEEE Trans Inf Technol Biomed, Vol. 11, No. 2. (March 2007), pp. 190-202.

[14] Yossi Rubner , Carlo Tomasi , Leonidas J. Guibas, "*A Metric for Distributions with Applications to Image Databases, Proceedings of the Sixth International Conference on Computer Vision*", p.59, January 04-07, 1998.

[15] Ashirvad rameshwar Naik, K.C Chang, "*A comparison of distance metrics between mixture distributions.*" Proceedings of SPIE 2010.

[16] CC Aggarwal, A. Hinneburg, and DA keim, "*On the surprising behavior of distance metrics in high dimensional space*," in Proc. of the international conference on Database theory (ICDT 2001).

[17] Ping Li, "*Very sparse stable random projections for dimension reduction in lα ($0 < \alpha \leq 2$) norm*" Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (2007)

[18] Peter Howarth and Stefan Ruger, "*Fractional Distance Measures for Content-Based Image Retrieval*," D.E Losadsa and J.M Fernandez-Luna (Eds,): ECIR 2005, LNCS 3408, pp. 447-456, 2005