SAND REPORT

SAND2002-1969
Unlimited Release
Printed June/2002

Fisher Information: Its Flow, Fusion, and Coordination

Timothy Berg

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/ordering.htm

SAND2002-1969
Unlimited Release
Printed June 2002

Fisher Information: Its Flow, Fusion, and Coordination

Tim Berg
Microsystems Partnerships
Sandia National Laboratories
P. O. Box 5800
Albuquerque, NM 87185-1204

Abstract

The information form of the Kalman filter is used as a device for implementing an optimal, linear, decentralized algorithm on a decentralized topology. A systems approach utilizing design tradeoffs is required to successfully implement an effective data fusion network with minimal communication. Combining decentralized results over the past four decades with practical aspects of nodal network implementation, the final product provides an important benchmark for functionally decentralized systems designs.

Acknowledgements

The LDRD office supported this work through the Emerging Threats and Non-Proliferation and Materials Control Strategic Business Units. The author recognizes and appreciates the support of Ron Moya, Chuck Meyers, and Donna Chavez especially. Mim John, Len Napolitano, Will Bolton, Rush Robinett, Gerald Langheim, and Ann Campbell have provided encouragement and ideas. Leslie Lehoucq provided research on Fisher information while Robert Hillaire simulated early algorithms. Jason Speyer and Jonathon Wolfe of UCLA provided critical analysis uncovering logical flaws early on. Sandy Milliman provided administrative support; John Feddema and David Schoenwald proofread all but my remaining errors. All these efforts are greatly appreciated.

Introduction

This paper considers the problem of optimal, decentralized estimation— where there are no central facilities. Global sensor fusion centers or other central functions represent single points of catastrophic failure or compromise for a system. The ability to remove centralized communications, computation, and memory functions increases fault tolerance, data throughput, and system security. Central facilities continue to be used, however, for several reasons: centralized command structures are simple to understand and explain to customers, system designers are accustomed to centralized systems, and many software tools or algorithms for implementing decentralized systems do not yet exist. This paper addresses problems of the third type, specifically how system tradeoffs can be made in implementing decentralized estimation systems for a wide range of applications.

The application space for decentralized systems may be summarized by Table 1, which gives a taxonomy according to the critical system functions of communication, computation, and memory. The applications include distributed detection (e.g. distributed Chemical, Biological, and Nuclear non-proliferation), multi-sensor target tracking (e.g. monitoring of high quantities of accountable materials), and wide area estimation (e.g. updating the state of a nationwide information system).

The Kalman filter is used as a vehicle for assessing computation, communication and memory tradeoffs for decentralized estimation. This paper combines three extensions of the Kalman filter to produce a distributed implementation. The three extensions, each described below, are the decentralized filter topology, the information form of the Kalman filter, and real world tradeoffs used to implement Kalman filters. The result is an optimal, distributed algorithm that runs in parallel on systems of decentralized processors to accomplish robust, fault tolerant estimation, tracking, state update, and sensor fusion.

The Kalman Filter

The Kalman filter is an algorithm developed by R. E. Kalman which combines state space theory with recursive filter techniques to produce an optimal, vector space based tracking method [Kalman1960]. The Kalman filter operates in two stages: prediction and update, offering a convenient way to update a running computer model with new sensor data without recalculating all the data that preceded the new input. The filter recursively computes both the state and the covariance, though the covariance values become constant, a point that will be used later. The prediction stages are given by

$$\hat{\mathbf{x}}(k/k-1) = \mathbf{F}(k)\hat{\mathbf{x}}(k-1/k-1) \tag{1}$$

and

$$\mathbf{P}(k/k-1) = \mathbf{F}(k)\mathbf{P}(k-1/k-1)\mathbf{F}^T(k) + \mathbf{Q}(k-1) \tag{2}$$

Fault Type	System Type		
	Communication Sensor Net Internet Telecom/Cellular Robot Swarms Software Agents	Computation Sensor Fusion Image Processing Intelligent Routing Robot Reasoning Parallel Computing	Memory Diskless Computing Database Model Building Web Search Disk Farms
Communication Link Loss/Jamming Bandwidth Drop Sporadic/Intermittent Link Compromise	Denial of Service Man in the Middle Cell Switching Quality of Service Stealth Sensors	MPI node mapping Router table update Agent coordination Bus Errors Stranded Robot	Session Timeout Process Timeout Synchronization Data Loss Backup
Computation Node drop Floating Point Error Thread Competition Adversary Process	Encryption failure Router Failure Node Loss DNS etc. Loss	Node Saturation Code Paralellization Grid Loss Algorithm Design	Synchronization Server Load Caching Buffer Overflow
Memory Corruption Bit Flipping Table Corruption Compromise	Router Poisoning DNS Nonresolve Sensor Spoofing	Buffer Underflow Data Mining Algorithm Design Navigation	Data verity Lost Disks Hashing Privacy

Table 1. Taxonomy of Application Problems

where $\hat{\mathbf{x}}(k/k-1)$ represents the state estimate $\hat{\mathbf{x}}$ of the true state $\mathbf{x}$ at time k based on information up to time $k-1$. Similarly, $\hat{\mathbf{x}}(k-1/k-1)$ is the previous time step update. The prediction model matrix $\mathbf{F}(k)$ transforms the previous time update to a prediction of the states at the next, future time. In equation (2), $\mathbf{P}(k/k-1)$ is the covariance prediction and $\mathbf{Q}(k-1)$ is a covariance of a zero mean, Gaussian process noise $\mathbf{w}(k) \sim N(\mathbf{0}, \mathbf{Q}(k))$ of the true state dynamic equation

$$\mathbf{x}(k) = \mathbf{F}(k-1)\mathbf{x}(k-1) + \mathbf{w}(k) \tag{3}$$

which is unknown but estimated. Uppercase bold variables represent matrices while lowercase bold letters represent vectors. The Kalman filter update equations are given in this notation as

$$\begin{aligned} \hat{\mathbf{x}}(k/k) &= \hat{\mathbf{x}}(k/k-1) + \mathbf{K}(k)\{\mathbf{z}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k/k-1)\} \\ &= \{\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)\}\hat{\mathbf{x}}(k/k-1) + \mathbf{K}(k)\mathbf{z}(k) \end{aligned} \tag{4}$$

$$\begin{aligned} \mathbf{P}(k/k) &= \{\mathbf{P}^{-1}(k/k-1) + \mathbf{H}^T(k)\mathbf{R}^{-1}(k)\mathbf{H}(k)\}^{-1} \\ &= \{\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)\}\mathbf{P}(k/k-1) \end{aligned} \tag{5}$$

where the Kalman Gain is given by

$$\begin{aligned}\mathbf{K}(k) &= \mathbf{P}(k/k-1)\mathbf{H}^T(k)\{\mathbf{H}(k)\mathbf{P}(k/k-1)\mathbf{H}^T(k)+\mathbf{R}(k)\}^{-1} \\ &= \mathbf{P}(k/k)\mathbf{H}^T(k)\mathbf{R}^{-1}(k)\end{aligned} \tag{6}$$

The new terms $\mathbf{z}(k)$ and $\mathbf{R}(k)$ come from the observation equation representing a noise corrupted measurement process

$$\mathbf{z}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{v}(k) \tag{7}$$

where $\mathbf{z}(k)$ is an observation vector, $\mathbf{H}(k)$ is the observation model, and $\mathbf{v}(k)$ is measurement noise modeled as a zero mean, uncorrelated sequence:

$$E[\mathbf{v}(k)\mathbf{v}^T(k')] = \delta_{kk'}\mathbf{R}(k) \tag{8}$$

where $E[\cdot]$ is the expectation operator and $\mathbf{R}(k)$ is the observation noise matrix such that:

$$\mathbf{v}(k) \sim N(\mathbf{0}, \mathbf{R}(k)) . \tag{9}$$

The main assumptions of the Kalman filter are that the dynamics not represented by the system model and observation model may be treated as zero mean, Gaussian noise. The models can be derived from first principles of physics but in practice they are also derived by removing all structure from system signals until all that is left is zero mean and Gaussian noise. The Central Limit Theorem helps explain why, over time, many non-Gaussian distributions come close to the normal distribution. The Kalman filter is thus useful for a surprisingly wide range of linear systems, and many nonlinear extensions to the theory have also been developed.

On the other hand, the Kalman filter is called a filter because of its noise rejection properties. In some applications it is actually more of an averaging "filter," balancing the sensor noise against what is not understood about nor represented in the system dynamic model. In some cases where the sensor and process model uncertainties are of similar magnitudes, an averaging filter will perform basically as well as a Kalman filter.

Notwithstanding, the Kalman filter is a graceful mathematical means wherein recursive filtering is statistically justifiable as optimal. It can be derived statistically as a maximum likelihood estimator, using the expectation operator, by Baye's Law with appropriate probability distribution functions, as a weighted least squares estimator using the normal equations, by minimizing mean squared error or other measures of covariance including its trace or determinant, and by the geometric arguments of projection theory. Thus one of its main uses in this paper is as a measure of optimality in decentralized systems where optimality can be difficult to define. For centralized problems, the Kalman filter is the benchmark against which suboptimal filters may be compared to determine their performance. In decentralized estimation, it is often assumed that no optimal linear solution exists and therefore no benchmark is available or required. The present results provide a benchmark for optimality of decentralized estimation for an important class of problems under the standard Kalman filtering assumptions.

Decentralized Kalman Filtering

Decentralized Kalman filtering involves running two or more Kalman filters which share information. Since each filter communicates, each physical implementation must add communication to its capabilities which at a minimum include computation and memory. Each physical filter implementation is called either a node, agent, or intelligent sensor, etc. The way the nodes' functions are arranged is called its architecture, and it is often determined by a communication topology. A common topology, for example, is a hierarchical one where subprocessors preprocess data and send it to a central fusion center to be combined. There are many architecture and topology variations, however.

Some centralized memory arrangements are called blackboard architectures where all nodes have access to a shared memory—the blackboard. Just because the memory is shared and common to all nodes does not mean it is necessarily central, however, since it can be duplicated and synchronized between locations. This creates the synchronization problem. Consider also the centralized communications set-ups called broadcast architectures where all nodes are privy to every communication over a common medium, e.g. radio or sonar. Such systems introduce issues such as data time and origin stamping to keep track of all of it. If all signals are sent over the same frequency or cable, then communications are really centralized and bottleneck issues become prominent for high data rates. The hierarchical architecture mentioned above is really a centralization of computation, similar to an office computer network with a server feeding desktop applications to thin clients. Other common functions that may be centralized or redundant include power sources, system management, and buffer memory; these also may represent single point failures.

Returning to the Kalman filtering example, each filter may be represented by subscripting each state according to the node where it is running. Thus, there exists for each subdomain a true substate vector, related to the true state vector as [Willsky1982]

$$\mathbf{x}_a = \mathbf{U}_a \mathbf{x}_G \tag{10}$$

and each local filter makes its local prediction:

$$\hat{\mathbf{x}}_a(k/k-1) = \mathbf{F}_a(k)\hat{\mathbf{x}}_a(k-1/k-1) \tag{11}$$

$$\mathbf{P}_a(k/k-1) = \mathbf{F}_a(k)\mathbf{P}_a(k-1/k-1)\mathbf{F}_a^T(k) + \mathbf{Q}_a(k-1)\,. \tag{12}$$

The question of the optimality of this prediction, the filter's prior in a statistical sense, is immediately raised. The issue is often stated as, "will this local filter produce the same results as a global, all knowing, optimal filter would?" This definition of optimality is incomplete, however. In practice, a global filter is run and the result is then transformed to the local subspace and compared to the performance of the local filter. So the question actually answered is, "does a local filter produce the same result at the local subspace as a globally optimal filter transformed to its subspace?" This is important because it points to the need to include enough degrees of freedom in the local filter in the first place to accurately represent the local system dynamics. Furthermore, the interfaces between local filters and the global filter, as well as between the local filters themselves must be

properly defined. Oftentimes a global filter is decentralized by performing a global mathematical signal analysis in an attempt to partition the states "naturally" down to reduced order observers. While it is intellectually satisfying to hope that a highly complex system (or the world) could be ordered from the top down, the systems engineer will better understand the system if it is partitioned from the bottom up with well defined interfaces. Furthermore, if the bottom up system design approach is used, the process of adding new nodes and initializing them into the network is more easily automated.

For example, the nuclear power plant controller wants to know pressures and temperatures at specific locations because safety control features have been designed into those locations. These parameters form natural interfaces between monitored zones since a specific location's temperature and pressure must be identical by each node's reckoning. If both nodes are optimal, they will produce the same result, while agreeing results do not guarantee optimality.

Mathematically, Sandell [Sandell1978] handled the local to global prediction model relationship by arguing for a necessary and sufficient dynamic equivalence condition on the process model

$$\mathbf{F}_a\mathbf{U}_a = \mathbf{U}_a\mathbf{F}_G \tag{13}$$

so that the local system is matched to the global model and hopefully designed properly in the first place.

Next, the prediction must have good information to start with on which to base the prediction. Therefore the previous update needs to have been optimal and the one before that all the way back to an optimal filter initialization.

Sandwiched between the required optimal initialization and a dynamically equivalent prediction model are all the updates in between. The updates may be divided into two types: the local filter update with its own e.g. sensor data, and internodal updates with other nodes' data. The local update is simply a standalone filter without communication:

$$\begin{aligned}\hat{\mathbf{x}}_a(k/k) &= \hat{\mathbf{x}}_a(k/k-1) + \mathbf{K}_a(k)\{\mathbf{z}_a(k) - \mathbf{H}_a(k)\hat{\mathbf{x}}_a(k/k-1)\} \\ &= \{\mathbf{I} - \mathbf{K}_a(k)\mathbf{H}_a(k)\}\hat{\mathbf{x}}_a(k/k-1) + \mathbf{K}_a(k)\mathbf{z}_a(k)\end{aligned} \tag{14}$$

$$\begin{aligned}\mathbf{P}_a(k/k) &= \{\mathbf{P}_a^{-1}(k/k-1) + \mathbf{H}_a^T(k)\mathbf{R}_a^{-1}(k)\mathbf{H}_a(k)\}^{-1} \\ &= \{\mathbf{I} - \mathbf{K}_a(k)\mathbf{H}_a(k)\}\mathbf{P}_a(k/k-1)\end{aligned} \tag{15}$$

$$\begin{aligned}\mathbf{K}_a(k) &= \mathbf{P}_a(k/k-1)\mathbf{H}_a^T(k)\{\mathbf{H}_a(k)\mathbf{P}_a(k/k-1)\mathbf{H}_a^T(k) + \mathbf{R}_a(k)\}^{-1} \\ &= \mathbf{P}_a(k/k)\mathbf{H}_a^T(k)\mathbf{R}_a^{-1}(k)\end{aligned} \tag{16}$$

where for now it must be remembered that the update is a local update based only on the information from node a at time k.

For the nodes to share information, or for a global or hierarchical fusion center to globally update the data, requires relationships among the nodes. All decentralized

filtering is based on the partitioning of the observation model in [Hashemipur1988], greatly aided by the assumption of independence of observation noise:

$$\begin{aligned}\mathbf{z}(k) &= [\mathbf{z}_1^T(k),...,\mathbf{z}_m^T(k)]^T \\ \mathbf{H}(k) &= [\mathbf{H}_1^T(k),...,\mathbf{H}_m^T(k)]^T \\ \mathbf{v}(k) &= [\mathbf{v}_1^T(k),...,\mathbf{v}_m^T(k)]^T\end{aligned} \tag{17}$$

so that each of the m nodes takes local measurements $\mathbf{z}_a(k)$ of the global state vector according to the nodal observation equation

$$\mathbf{z}_a(k) = \mathbf{H}_a(k)\mathbf{x}(k) + \mathbf{v}_a(k). \tag{18}$$

With noise uncorrelated between partitions, $\mathbf{R}_a^{-1}(k)$ takes on a block diagonal form allowing a simple relationship between local and global terms in the second halves of equations (5) and (15) and similarly for (4) with (15) and (6) with (16):

$$\begin{aligned}\mathbf{H}^T(k)\mathbf{R}^{-1}(k)\mathbf{H}(k) &= \sum_{a=1}^{m}\mathbf{H}_a^T(k)\mathbf{R}_a^{-1}(k)\mathbf{H}_a(k) \\ \mathbf{H}^T(k)\mathbf{R}^{-1}(k)\mathbf{z}(k) &= \sum_{a=1}^{m}\mathbf{H}_a^T(k)\mathbf{R}_a^{-1}(k)\mathbf{z}_a(k)\end{aligned}. \tag{19}$$

Substituting yields the equations for the update at a global fusion center:

$$\hat{\mathbf{x}}_G(k\,|\,k) = \mathbf{P}_G(k\,|\,k)\mathbf{P}_G^{-1}(k\,|\,k-1)\hat{\mathbf{x}}_G(k\,|\,k-1) + \mathbf{P}_G(k\,|\,k)\sum_{a=1}^{m}\mathbf{H}_a^T(k)\mathbf{R}_a^{-1}(k)\mathbf{z}_a(k) \tag{20}$$

using the definition of Kalman gain and the identity

$$\mathbf{1} - \mathbf{K}_G(k)\mathbf{H}_G(k) = \mathbf{P}_G(k\,|\,k)\mathbf{P}_G^{-1}(k\,|\,k-1) \tag{21}$$

where $\mathbf{1}$ is the identity matrix.

Before proceeding, note that there is no qualitative difference between information originating at the local node versus information arriving from another node. In fact, there is no qualitative difference between information from the prior or prediction compared to the sensor information or the update. Each of these quantities is orthogonal with respect to $\mathbf{P}_G^{-1}(k\,|\,k)$. Multiplying Equation (20) through by $\mathbf{P}_G^{-1}(k\,|\,k)$

$$\mathbf{P}_G^{-1}(k\,|\,k)\hat{\mathbf{x}}_G(k\,|\,k) = \mathbf{P}_G^{-1}(k\,|\,k-1)\hat{\mathbf{x}}_G(k\,|\,k-1) + \sum_{a=1}^{m}\mathbf{H}_a^T(k)\mathbf{R}_a^{-1}(k)\mathbf{z}_a(k) \tag{22}$$

shows a useful structure of information update which, by way of a simple change of variables, gives the Information filter.

The Information Filter

The information filter makes use of the algebraic form of the Kalman filter based on Fisher information. Fisher information has superficial similarities to Shannon information but is an entirely distinct quantity derived by wholly different notions. Fisher information, used by R. A. Fisher [Fisher1925], relates in its most simple sense to when equality is achieved in the Cramer-Rao lower bound on estimation error. In a scalar form:

$$e^2 I \geq 1 \tag{23}$$

where e^2 is the mean squared error in an estimate and I is the Fisher information. Thus, the best estimate possible, i.e. the estimate making the best use of the data from a minimum mean squared error perspective, is attained when

$$e = \sqrt{\frac{1}{I}}\,. \tag{24}$$

For the matrix case of the vector space Kalman filter, the scalar mean squared error becomes the covariance matrix so that in an optimal filter, the inverse of the covariance matrix is the Fisher information

$$I = \frac{1}{e^2} \rightarrow \mathbf{P}^{-1} = \mathbf{I}\,. \tag{25}$$

Defining an information matrix and an information vector associated with the covariance matrix and state estimate, respectively:

$$\begin{aligned} \mathbf{I}(k/k-1) &= \mathbf{P}^{-1}(k/k-1) \\ \mathbf{i}(k/k-1) &= \mathbf{P}^{-1}(k \mid k-1)\hat{\mathbf{x}}(k \mid k-1) \end{aligned} \tag{26}$$

along with the information associated with a measurement $\mathbf{z}_a(k)$:

$$\begin{aligned} \mathbf{J}(\mathbf{z}_a(k)) &= \mathbf{H}_a^T(k)\mathbf{R}_a^{-1}(k)\mathbf{H}_a(k) \\ \mathbf{j}(\mathbf{z}_a(k)) &= \mathbf{H}_a^T(k)\mathbf{R}_a^{-1}(k)\mathbf{z}_a(k) \end{aligned} \tag{27}$$

gives the Information filter update equations from (4) and (22):

$$\begin{aligned} \mathbf{I}(k/k) &= \mathbf{I}(k/k-1) + \sum_a \mathbf{J}(\mathbf{z}_a(k)) \\ \mathbf{i}(k/k) &= \mathbf{i}(k/k-1) + \sum_a \mathbf{j}(\mathbf{z}_a(k)) \end{aligned}\,. \tag{28}$$

In symbolic form the information filter is $\mathbf{i}_G(\text{update}) = \mathbf{i}_G(\text{prediction}) + \sum_a \mathbf{j}_a(\text{new data})$.

The information filter is used for two reasons: it has an advantageous algebraic form and

it serves as a bridge to relate decentralized estimation to the broad range of current Fisher information advances which derive most every law of physics using Fisher information [Frieden1998]. This is helpful when modeling cooperative systems as obeying analogous laws of physics. The information filter is related to the Kalman filter by the matrix inversion lemma with variable substitutions, and its form is algebraically identical to the update stage of the filter both locally and when information from other filters is fused in to a local filter. Furthermore, the information form of the filter is helpful in deriving the update at a local node using data from other nodes.

Implementing Decentralized Optimal Filters

Equations (22) and (28) are global dimension updates at a fusion center assumed to have all system information. The easy way to ensure all nodes are globally optimal is to duplicate a global processing station at each node [Speyer1979], each running the same global update equation. If initialized identically, each will arrive at the same, globally optimal results "independently". This is not a scalable approach, however, since it requires full connectivity with a huge communication overhead. If the application is a database, a second, tandem processor makes a nice backup, but a dozen identical processors creates database compromise vulnerabilities and a synchronization problem.

What is required is distribution of the computational, communication, and memory requirements to a set of processors specializing in functions. For some systems, absolute minimum communication is a requirement. A local node not running the full state model should require only the state and observation information relevant to its local state subspace in order to calculate an update which is globally optimal over that local space. This requirement drives the communication topology required of the nodal network. Some nodes will not need to be connected, and some will require only one-way communications. A fully decentralized network will have aspects of hierarchy and autonomy depending on local and regional requirements.

To derive the globally optimal update at a local node, the problem of fusing in information from another node needs to be solved. The internodal data may be treated as though it came from the same node, provided the covariance issues of overlap, differing data confidence, and transformation of the substate space relationships are accounted for. The method of accomplishing this is to relate each local state and observation space to a common space and use the common space to relate them to each other. For simplicity, all states will be related to the global state space, though computation can be reduced by accomplishing this on a regional level. The complexity of the region needs to be large enough to describe the nodal interface, but not necessarily of global dimension. There is a tradeoff between creating more regions to reduce the local computation, and the management overhead of keeping track of all the regions and their interface descriptions.

One way to attack the problem is to derive a local update completely free of global information including the covariance information, i.e. the $\hat{\mathbf{x}}_G$ and $\mathbf{P}_G$ terms. But note that

by the definition of covariance and the state relationships (10) the local covariance is related to the global covariance by premultiplication and postmultiplication:

$$\mathbf{P}_a(k/k-1) = \mathbf{U}_a(k)\mathbf{P}(k/k-1)\mathbf{U}_a^T(k)\,. \tag{29}$$

This relationship significantly complicates distribution of system resources because it reflects the physical realities of correlation between nodes that are observing the same physical phenomena. As a tradeoff, we can perform the covariance calculations in global terms by precomputing them. Since the covariance terms are not data dependent and stabilize to constant terms, they may be computed offline and stored in local memory. If a new node is added, it can supply a new covariance term as part of the initialization procedure as it comes online.

The state update is more problematic and several approaches have been suggested. In [Berg1993], an internodal transformation is suggested based on a projection theory approach but the internodal transformation turns out to be applicable only to a very narrow range of situations. The internodal transformation approach is coopted in [Mutambara1998] with additional errors. Willsky [Willsky1983] suggests a scattering framework using a superposition principle, also employs the information filter form, and addresses the problem of incomplete local models, but unfortunately assumes a central fusion center. Speyer [Speyer1979] states that only a minimum sized vector needs to be communicated, provided an additional calculation is performed at each node to account for it being suboptimal. This theoretical result can be improved on in practice to the point where only local sized information is transmitted. There is a much wider range of practical tradeoffs between computation, communication, and memory in theory as well.

The most straightforward way to derive a filter that can be implemented at every local filter is to rearrange (22)

$$\hat{\mathbf{x}}_a(k\,|\,k) = \mathbf{U}_a\mathbf{P}_G(k\,|\,k)\mathbf{P}_G^{-1}(k\,|\,k-1)\hat{\mathbf{x}}_G(k\,|\,k-1) + \mathbf{U}_a\mathbf{P}_G(k\,|\,k)\sum_{b=1}^{m}\mathbf{H}_b^T\mathbf{R}_b^{-1}\mathbf{z}_b(k) \tag{30}$$

where this time $\hat{\mathbf{x}}_a(k\,|\,k)$ is not a local update, but an update of all global information relevant to the node. It is interesting to note the similarity to the forms in [Frieden1998] where the intrinsic information variable (I in his notation) is diagonal while the correlations are accounted for in the extrinsic information (Frieden's J). The observation model may be reduced to a local order sized observation model $\mathbf{C}_a(k)$ [Willsky1982] by a linear transformation

$$\mathbf{H}_a(k) = \mathbf{C}_a(k)\mathbf{S}_a(k)\,. \tag{31}$$

That $\mathbf{S}_a(k)$ is distinct from $\mathbf{U}_a(k)$ reflects the possibility that the filter may be operating as an observer reconstructing states it does not directly observe, but either obtains through communication, or calculates using its process model. The partitions become

$$\mathbf{H}^T(k)\mathbf{R}^{-1}(k)\mathbf{H}(k) = \sum_{a=1}^{m}\mathbf{S}_a^T(k)\mathbf{C}_a^T(k)\mathbf{R}_a^{-1}(k)\mathbf{H}_a(k)$$
$$\mathbf{H}^T(k)\mathbf{R}^{-1}(k)\mathbf{z}(k) = \sum_{a=1}^{m}\mathbf{S}_a^T(k)\mathbf{C}_a^T(k)\mathbf{R}_a^{-1}(k)\mathbf{z}_a(k) \qquad (32)$$

making (30)

$$\hat{\mathbf{x}}_a(k\,|\,k) = \mathbf{U}_a\mathbf{P}_G(k\,|\,k)\mathbf{P}_G^{-1}(k\,|\,k-1)\hat{\mathbf{x}}_G(k\,|\,k-1) + \mathbf{U}_a\mathbf{P}_G(k\,|\,k)\sum_{b=1}^{m}\mathbf{S}_b^T(k)\mathbf{i}_b(k) \qquad (33)$$

where

$$\mathbf{i}_b(k) = \mathbf{C}_b^T(k)\mathbf{R}_b^{-1}(k)\mathbf{z}_b(k)\,. \qquad (34)$$

Therefore, only the small, local, data dependent vector need be communicated and it can be updated directly with most other terms in (33) not data dependent and therefore precomputable. All $\mathbf{U}_a\mathbf{P}_G(k\,|\,k)\mathbf{S}_b^T(k)$ terms could be precomputed and stored to tradeoff more memory against less processing for increased speed. Similarly, this approach may be used at some nodes and not others depending on nodal requirements. Furthermore, the prediction that is depicted in global variables need only be perfomed over variables of local relevance. It should therefore be a sparse matrix operation and may be dynamically equivalent to the local prediction with local covariance sized terms under some circumstances.

Conclusions

Decentralized systems are increasingly important in a widening array of applications from information systems, sensor networks, cooperative robotics, communications systems, high performance computing, and power grids. Such decentralized implementations do not just benefit from a systems design approach but require it for their effective functioning. The Kalman filter is an excellent vehicle for exploring decentralized system tradeoffs because of its optimality and linearity. This paper provides that understandable benchmark and reference framework by developing a decentralized Kalman filter for a decentralized topology. Previous efforts have concentrated on either decentralized filters on centralized or hierarchical topologies or essentially centralized filters on decentralized topologies. This implementation assumes a truly decentralized architecture requiring no central facilities. Several decentralized results spanning four decades are brought together to decentralize the filter algorithm, topology, and services through tradeoffs of computation and memory that minimize communication. The result is an optimal, linear filter which needs only to share nodal dimension information between autonomous nodes. The results recognize the Information filter forms, which provide ties to an increasingly rich body of research that will allow modeling complex information systems using the laws of physics.

Bibliography

N. Bekhouche, A. Feliachi, Mar. 20, 1988, "Decentralized Discrete-Time Filters." In *Proceedings of the Southeastern Symposium on System Theory*, pp. 81-83.

T. M. Berg, 1993, "Model Distribution in Decentralised Multisensor Data Fusion." D.Phil. Thesis, Department of Engineering Science, University of Oxford, United Kingdom.

N. A. Carlson, May 23, 1988. "Information-Sharing Approach To Federated Kalman Filtering." In *Proceedings of the IEEE National Aerospace and Electronics Conference,* p. 1581.

D. A. Castanon, D. Teneketzis, May 1985, "Distributed Estimation Algorithms for Nonlinear Systems." *IEEE Transactions on Automatic Control,* 30(5):418-425.

K. C. Chang, 2000, "Evaluating Hierarchical Track Fusion with Information Matrix Filter." In *Proceedings of Fusion '00,* Paris, France.

K. C. Chang, C. Y. Chong, Y. Bar-Shalom, 1992, "Distributed Estimation in Distributed Sensor Networks." Large-Scale Stochastic Systems Detection, Estimation, Stability and Control, Chapter 2, Marcel Dekker.

K. C. Chang, Zhi Tian, R. K. Saha, July 1998, "Performance Evaluation of Track Fusion with Information Filter," In *Proceedings of. Fusion '98*, Las Vegas, USA.

C. Y. Chong, July 1998, "Distributed Architecture for Data Fusion." In *Proceedings of Fusion '98*, Las Vegas, USA.

C. Y. Chong, 1979, "Hierarchical Estimation." In *Proceedings of the MIT/ONR Workshop on* C^3.

C. Y. Chong, S. Mori, W. H. Barker, K. C. Chang, Jul. 1999, "Architectures and Algorithms for Track Association and Fusion." In *Proceedings of Fusion '99,* Sunnyvale, USA.

C. Y. Chong, Shozo Mori, Aug. 2001, "Convex Combination and Covariance Intersection Algorithms in Distributed Fusion." In *Proceedings of Fusion '01,* Montreal, Canada.

Oliver E. Drummond, Jul. 1999, "On Features and Attributes in Multisensor Multitarget Tracking." In *Proceedings of Fusion '99,* Sunnyvale, USA.

A. Feliachi, Nov. 14, 1989, "On The Decentralized Control Of Large-Scale Systems." In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics,* pp. 940-944.

R. A. Fisher, 1925, “Theory of Statistical Estimation.” In *Proceedings of the Cambridge Philosophical Society*, 22(5):700--725.

David Hall, Amulya K. Garga, Jul. 1999, “Pitfalls in Data Fusion,” In *Proceedings of Fusion ’99,* Sunnyvale, USA.

H. R. Hashemipour, S. Roy, A. J. Laub, Jan. 1, 1988, “Decentralized Structures For Parallel Kalman Filtering.” *IEEE Transactions on Automatic Control,* 33(1):88-94.

R. E. Kalman, 1960, “A new approach to linear filtering and prediction problems.” *Transactions of the ASME -- Journal of Basic Engineering*, pages 35--45.

R. L. Kashyap, S. G. Oh, R. N. Madan, Jan. 1990, “Robust Estimation Of Sinusoidal Signals With Colored Noise Using Decentralized Processing.” *IEEE Transactions on Acoustics, Speech and Signal Processing,* 38(1): 91-104.

B. C. Levy, D. A. Castanon, G. C. Verghese, A. S. Willsky, July1983, “A Scattering Framework for Decentralized Estimation Problems.” *Automatica,* 19(4):373-384.

S. Mori, K. C. Chang, C. Y. Chong, 1992, “Performance Analysis of Optimal Data Association with Application to Multiple Target Tracking.” Multitarget-Multisensor Tracking: Applications and Advances, Vol. II, Chapter 7, Artech House.

S. Mori, C. Y. Chong, E. Tse, R. P. Wishner, May 1986, “Tracking and Classifying Multiple Targets without *A Priori* Identification.” *IEEE Transactions on Automatic Control,* 31(5):401-409.

Arthur G. O. Mutambara, 1998, *Decentralized Estimation and Control for Multisensor Systems*, CRC Press.

B. S. Y. Rao, H. F. Durrant-Whyte, 1991, “A Fully Decentralised Algorithm for Multi-sensor Kalman Filtering.” *IEE Transactions Schedule D*, 138(5):413-420.

D. B. Reid, Dec. 1979, “An Algorithm For Tracking Multiple Targets.” *IEEE Transactions on Automatic Control,* 24(6):843-854.

X. Rong Li, Keshu Zhang, Aug. 2001, “Optimal Linear Estimation Fusion – Part IV: Optimality and Efficiency of Distributed Fusion.” In *Proceedings of Fusion ’01,* Montreal, Canada.

X. Rong Li, Y. Zhu, C. Han, 2000, “Unified Optimal Linear Estimation Fusion – Part I: Unified Models and Fusion Results.” In *Proceedings of Fusion ’00,* Paris, France.

S. Roy, R.A. Iltis, Nov. 1991, "Decentralized Linear Estimation In Correlated Measurement Noise." *IEEE Transactions on Aerospace and Electronic Systems,* 27(6):939-941.

N. R. Sandell Jr., P. Varaiya, M. Athans, and M. G. Safonov, Apr. 1978, "Survey Of Decentralized Control Methods For Large Scale Systems." *IEEE Transactions on Automatic Control*, 23(2):108-128.

J. L. Speyer, Apr. 1979, "Computation and Transmission Requirements for a Decentralized Linear-Quadratic-Gaussian Control Problem." *IEEE Transactions on Automatic Control,* 24(2):266-269.

A. S. Willsky, M. G. Bello, D. A. Castanon, B. C. Levy, G. C. Verghese, Aug. 1982, "Combining and Updating of Local Estimates and Regional Maps Along Sets of One-Dimensional Tracks." *IEEE Transactions on Automatic Control,* 27(4):799-813.

Yuan Xin, Wu Deping, May 25, 1992, "Development Of Information Integration In Navigation." *Proceedings of the IEEE International Symposium on Industrial Electronics,* pp. 57-60.

Luo Zhi-Quan, J. N. Tsitsiklis, Sep.1994, "Data Fusion With Minimal Communication." *IEEE Transactions on Information Theory,* 40(5):1551-1563.

Distribution

1	MS 0741	R. D. Robinett, 06200
4	MS 1204	T. M. Berg, 05911
1	MS 9103	W. R. Bolton, 08120
5	MS 1004	RMSEL Technical Library, 15221
1	MS 9018	Central Technical Files, 8945-1
2	MS 0899	Technical Library, 9616
1	MS 0612	Review & Approval Desk, 9612 For DOE/OSTI
1	MS 0188	D. Chavez, LDRD Office, 4001