MANAGING DATA INCEST IN A DISTRIBUTED SENSOR NETWORK

Samuel McLaughlin

Dept. of Electrical
and Electronic Engineering
University of Melbourne
Victoria, 3010, Australia.
spmcl@ee.mu.oz.au

Vikram Krishnamurthy

Dept. of Electrical
and Computer Engineering
The University of British Columbia
Vancouver, B.C, Canada V6T 1Z4.
vikramk@ece.ubc.ca

Subhash Challa

Dept. of Electrical
and Electronic Engineering
University of Melbourne
Victoria, 3010, Australia.
s.challa@ee.mu.oz.au

ABSTRACT

Multisenor data fusion is one of the key evolving technologies that can be implemented on several architectures. Distributed sensor network architecture provides a good balance between cost, scalability and communication limits of links connecting multiple sensors. However, this architecture is prone to the problem of Data Incest. Data incest arises due to multiple usage of identical information as if it were independent information. In most cases, it will falsely increase the confidence of a biased overall estimate. It is a fundamental issue in the many-to-many sensor network configurations, where all network nodes are communicating with all nodes. This papers describes a fusion strategy which can be adopted for this distributed network structure that renders incest free estimates.

1. INTRODUCTION

Traditionally, multi-sensor tracking systems have opted for central fusion nodes as seen in [1]. Under this many-to-one network philosophy each sensor node forwards its local *measurement* to a central fusion hub where an estimate of the target dynamics, such as position and speed, are calculated. Network bandwidth limitations generally render this practice infeasible, due to the large size of the raw *measurements*. Without these limitations, this centralized architecture is theoretically optimal in terms of minimum variance.

In this paper, motivated by recent applications in network-centric (internet) warfare systems [2], we consider a distributed sensor network where each node comprises a sensor and a local fusion center. The sensor receives local noisy *measurements* of some dynamical system (e.g. a target) and remote *information* from other sensors. These are processed locally to generate an updated Bayesian estimate. The local processing involves a recursive Bayesian state estimator e.g. a Kalman filter [3] or Hidden Markov Model filter [4]. The updated Bayesian estimate is then broadcast on the common sensor network (e.g. internet) and is available to other nodes. This local processing can result in significant data compression and hence efficient utilization of the network if the Bayesian estimate (filtered density function) can be parameterized by a finite dimensional statistic, which typically has a much smaller bit size representation compared to the raw measurements.

Consider for example the case where each sensor records noisy measurements from a Gaussian state space process. The information broadcasted on the network is the mean and variance of the state estimate computed by a Kalman filtering algorithm. In the absence of common process noise, individual estimates of track one and track two (relating to the same target) can be fused with the resultant estimate using [5]

$$\hat{x} = (\frac{\hat{x}_1}{P_1} + \frac{\hat{x}_2}{P_2})P \tag{1}$$

where $\hat{x}_i$ and P_i are the mean and variance of track i and

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

Conversely, knowledge of track one and of the fused estimate of track one and two, allows for the extraction track two based on the general format of (1)and (2) (with the plus sign replaced by a minus sign).

An important issue in the design of a distributed sensor network is the need to manage *data incest*. Data incest arises due to the repeated use of identical information. Consider the example of a two node network with each node comprising of a sensor and a Kalman filter. The first Kalman filter receives a noisy measurement, updates its state estimate and broadcasts this information (over the internet for example). After a random delay, this estimate is available to the second Kalman filter. The second Kalman filter uses this received information together with its local measurement and computes an updated estimate. When this estimate is received at the first Kalman filter, blind incorporation would result in incest, as it does not know the fact that sensors 1's information was already embedded in the received information. In other words, if the first Kalman filter now updates its estimate by combining the estimate of the second Kalman filter and its own local estimate, it would have effectively used the first measurement twice. This is an example of data incest. Data incest occurs due to non-standard information patterns, where some of the data can unknowingly be used multiple times, resulting in a significantly biased estimate. Random delays complicate the issue. The inclusion of random delays in the problem space further complicates the issue.

This paper describes a fusion strategy which can be applied to a distributed network structure that automatically produces incest free estimates at the output of each node . Specifically, the problem will be investigated for the situation where there is a random transmission delay through the network (as in a packet switched network). Of principal interest, is an attempt to minimize extra storage and transmission requirements.

S. McLaughlin and S. Challa were supported by the Cooperative Research Centre for Sensor Signal and Information Processing and DSTO through TDFL.

Section 2 investigates feasible network architectures for a random delay environment. Section 3 then sets out the framework for the problem and includes important notation. Section 4 explains how the algorithm works and Section 5 concludes with some limited simulation results.

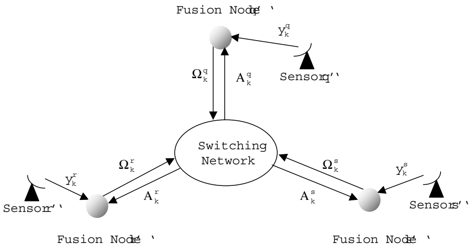


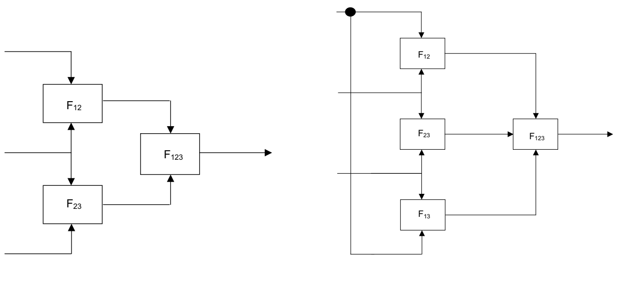
Fig. 1. GENERIC ARCHITECTURE

2. FUSION ARCHITECTURES WITH INCEST

The most generic architecture for a sensor network, where information is fused locally, is shown in Figure 1. The noisy local *measurement* ($\mathbf{y}_k^j$) and remote *information* ($\mathbf{A}_k^j$) arriving at sensor j in the discrete interval $[k-1, k]$ to be processed at time k. The destination of the subsequent information ($\mathbf{\Omega}_k^j$) is governed by the structure of the switching network. When only mean and variance statistics are passed through a network of fixed transmission delay, the network structure can dictate whether the overall system can be made incest free. Figure 2 illustrates two examples of a simple two stage fusion example where incest must be managed.

Consider the *partially interconnected* case. Without incest management the output of the final stage (F_{123}) will be the posterior $p(x_0|Y_k^1, Y_k^2, Y_k^2, Y_k^3)$, which incorporates information from sensor s_2 twice. This leads to over confidence in the overall estimate x_0 that is (undesirably) biased towards the measurement from s_2. If the likelihood of the sensor s_2 information is available at F_{123},

$$p(x_0|Y_k^1, Y_k^2, Y_k^3) = \frac{1}{\delta}\frac{p(x_0|Y_k^1, Y_k^2, Y_k^2, Y_k^3)}{p(Y_k^2|x_0)}, \tag{3}$$



(a) Partially Interconnected

(b) Fully Interconnected

Fig. 2. NETWORK ARCHITECTURES

removes the incest where δ is a constant. With a partially interconnected structure this likelihood is not computable from the information available at the second fusion stage (F_{123}). Thus incest removal is not possible with this configuration.

A *fully interconnected* configuration, which includes a third node (F_{13}) in the first fusion stage, accommodates for the local computation of the likelihood $p(Y_k^2|x_0)$. Bayes' rule and the mutual independence of the sensor measurements allows the decomposition

$$p(x_0|Y_k^1, Y_k^2) = \frac{p(Y_k^1|x_0)p(Y_k^2|x_0)p(x_0)}{p(Y_k^1, Y_k^2)}. \tag{4}$$

Division of this posterior by $p(x_0|Y_k^1, Y_k^3)$, gives

$$\frac{p(x_0|Y_k^1, Y_k^2)}{p(x_0|Y_k^1, Y_k^3)} = \frac{1}{\delta_1}\frac{p(Y_k^2|x_0)}{p(Y_k^3|x_0)}. \tag{5}$$

where the normalization constant δ_1 has no bearing on the mean and variance of the estimate of x_0. With the aid of the third (extra) fused PDF

$$\frac{1}{\delta_1}\frac{p(Y_k^2|x_0)}{p(Y_k^3|x_0)}\frac{p(x_0|Y_k^2, Y_k^3)}{p(x_0)} = \frac{1}{\delta_2}p(Y_k^2|x_0)p(Y_k^2|x_0). \tag{6}$$

gives the squared likelihood, whose vital statistics are not affected by δ_2. Using (1) and (2), it can easily be shown that the mean of the likelihood $p(Y_k^2|x_0)$ is identical to that of the squared likelihood, while the variance will be twice that of the squared version.

This section has highlighted two important results that are integral to the rest of the paper. Firstly, a fully interconnected sensor network will be required. Secondly, the prior $p(x_0)$ must be identical AND available at all sensor nodes.

3. NOTATIONS, DEFINITIONS AND PROBLEM FORMULATION

The problem presented here involves S distributed sensors employed to maintain a bearing of a *static* target. The discrete state space representation for the stationary target which is common to all sensors is

$$x_{k+1} = Fx_k \tag{7}$$

and the measurements from sensor s are given by

$$y_k^s = Hx_k + w_k^s \qquad \text{for all } s \in [0, S] \tag{8}$$

where $w_k^s \sim \mathcal{N}(0, R)$ is the measurement noise at sensor s. In the static case, the transition matrix F and measurement matrix H are both identity matrices, $I \in \mathbf{R}^{n\times n}$, where n is the dimension of the state x_k.

Recursive updates of the conditional mean estimate $\hat{x}_{k|k}^s = \mathbf{E}\{x_k|I_k^s\}$ and covariance $P_{k|k}^s = \mathbf{E}\{(x_k - \hat{x}_k^s)(x_k - \hat{x}_k^s)'|I_k^s\}$ are calculated at each node under the assumption that the prior $p(x_0)$ is Gaussian. Information available at node s is represented by I_k^s (see below). The first two moments of the PDF

$$p(x_k, I_k^s) \sim K\,\mathcal{N}(\hat{x}_{k|k}^s, P_{k|k}^s), \tag{9}$$

will constitute the essential information transmitted from node s to all S nodes subject to a random delay $D \in [D_{min}, D_{max}]$.

The remainder of this section is devoted to defining notation imperative to the understanding of the proposed solution.

Node Time Pair (NTP)

A NTP(s,k) is defined as an event taking place at fusion node s at time k.

Output Packet (OP)

The ideal transmission information between all sensors is simply the two moments of the joint Gaussian density as in (9) where

- $I_k^s = (Y_{\lambda_1}^1, \ldots, Y_{\lambda_S}^S)$ is the information from all sensors available at NTP(s,k)
- $Y_{\lambda_r}^r = (y_1^r, \ldots, y_{\lambda_r}^r)$ is the collection of measurements from remote sensor r available at the current node s
- $\lambda_r \in \{k-2D, \ldots, k\}$ indicates the time of the most recent *usable* information available from sensor r.

However, this is not sufficient to ensure the complete removal of incest. As a result information additional to the ideal case must be sent. Therefore the OP sent from NTP(s,k) is defined as

$$\Omega_k^s = (\hat{x}_{k|k}^s, P_{k|k}^s, \Lambda_k^s). \tag{10}$$

Note that all "packets" must contain all three of these components.

Usable Time Vector (UTV)

The UTV is integral to the output posterior PDFs being free from incest when the packets are subject to random transmission delays. Its elements are the times of the most recent *usable* information available from all sensors at NTP(s,k). It is defined as

$$\Lambda_k^s = (\lambda_1, \ldots, \lambda_S).$$

Compound Likelihood Matrix (CLM)

The storage requirement for each local node is a $S \times 2D$ sliding window matrix, defined as the *compound likelihood matrix*

$$\mathbf{\Gamma}_k^s = \begin{bmatrix} \tilde{p}(Y_{k-2D}^1|x_0) & \tilde{p}(Y_{k-2D}^2|x_0) & \ldots & \tilde{p}(Y_{k-2D}^S|x_0) \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{p}(Y_{k-D}^1|x_0) & \tilde{p}(Y_{k-D}^2|x_0) & \ldots & \tilde{p}(Y_{k-D}^S|x_0) \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{p}(Y_{k-1}^1|x_0) & \tilde{p}(Y_{k-1}^2|x_0) & \ldots & \tilde{p}(Y_{k-1}^S|x_0) \end{bmatrix}$$

where

$$\tilde{p}(Y_k^s|x_0) = p(y_1^s|x_0) \times \ldots \times p(y_k^s|x_0) = \prod_{i=1}^{k} p(y_i^s/x_0).$$

Some of the CLM entries may be empty, indicating delayed information is yet to be received.

Arrival Buffer (AB)

The arrivals refer to the updated estimates from all sensor nodes. The *arrival buffer* at NTP(s,k) can potentially contains OPs from as many as $(S-1) \times (D_{max} - D_{min} + 1) + 1$ or as little as 1 depending on packet arrivals in the interval $[k-1, k]$. It is defined as

$$\mathbf{A}_k^s = \left\{ 2^{\mathcal{S}_1}, \ldots, 2^{\mathcal{S}_S} \right\},$$

where

$$\mathcal{S}_i = \begin{cases} \{\Omega_{k-D_{max}}^i, \ldots, \Omega_{k-D_{min}}^i\} & \text{if } i \neq s \\ \{\Omega_{k-1}^i\} & \text{otherwise.} \end{cases}$$

Each individual OP $(\mathbf{\Omega}_j^i)$ constitutes one element of the AB $(\mathbf{A}_k^s)$ for $k > j$.

Local Packet (LP)

The OP from NTP$(s,k-1)$ is classified as an arrival at NTP(s,k), despite not physically entering the network. This packet,

$$\Omega_{k-1}^s \in \mathbf{A}_k^s.$$

is defined as the *local packet* for NTP(s,k).

Current Packet (CP)

The result of fusion between the local measurement $(\mathbf{y}_k^s)$ with the LP $(\mathbf{\Omega}_{k-1}^s)$ is the *current packet*. Any further updates to the CP simply results in the CP taking on a different value, until ultimately it becomes the OP, when there is no further arrivals exist to process. The CP is defined as:

$$\bar{\mathbf{\Omega}}_k^s = (\bar{\hat{x}}_{k|k}^s, \bar{P}_{k|k}^s, \bar{\Lambda}_k^s)$$

4. INCEST MANAGEMENT

The basic principle underlying this solution methodology involves extracting only the most recent NTP information from all incoming packets (elements of the AB). Once this is done it is then fused with current local estimate.

The block diagram in Figure 3 outlines the steps involved in the fusion process at any given NTP(s,k). The first step, defined as the *filtering* stage, adds the new local measurement $(\mathbf{y}_k^s)$ to the existing local *information* (LP). The updates of mean and variance follow standard Kalman filter equations. The UTV $\mathbf{\Lambda}_{k-1}^s$ becomes $\bar{\mathbf{\Lambda}}_k^s$ by incrementing the s^{th} component of the vector.

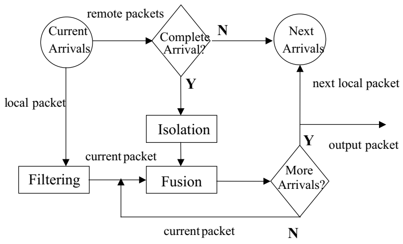


Fig. 3. FUSION CENTRE FOR NTP(s,k)

The AB is checked for packets from other nodes. If there are no arrivals (other than the LP) the CP $(\bar{\mathbf{\Omega}}_k^s)$ becomes the OP $(\mathbf{\Omega}_k^s)$ and there is no requirement for fusion at this NTP. Having removed the LP from the AB, any remaining elements are then processed on an individual basis, beginning with the received packet with oldest send time. The ordering of nodes is insignificant as the minimum delay can not be zero. The first decision diamond in Figure 3 determines whether each of the remaining packets is *usable* or not.

An *incomplete arrival* occurs when a packet from NTP(r,i) arrives at node s later than a packet from NTP(r,j) where $j > i$. Figure 4 illustrates the two possible scenarios that lead to an incomplete arrival. An incomplete direct arrival (IDA) involves $\mathbf{\Omega}_j^r$ arriving at node s already fused with another packet, before it arrives directly. An incomplete indirect arrival (IIA) is slightly more complicated. It occurs when information from NTP(q,j), incorporating information from NTP(r,i), arrives at a third node s before $\mathbf{\Omega}_i^r$ arrives at that node directly. For both the IDA and IIA case,

$$\exists \quad \lambda_r = i,$$

where the rth component of each of the arrival buffers $\mathbf{\Lambda}_j^r$ and $\mathbf{\Lambda}_j^q$ is equal to λ_r. The absence of $\tilde{p}(Y_i^r|x)$ in the local CLM, prevents the isolation of both $p(y_j^r/x)$ and $p(y_j^q/x)$.

In reference to an incoming packet, the terms *usable* (constituting the first word of the acronym UTV) and *complete arrival* can be used interchangeably. If

$$\exists \quad \lambda_i \in \Omega_j^r \quad s.t \quad \forall i = 1, 2, \ldots, S, \qquad \tilde{p}(Y_{\lambda_i}^i|x_0) \quad \in \quad \Gamma_k^s$$

then $\mathbf{\Omega}_j^r$ is a complete arrival at NTP(s, k). It was shown in Section 2 that, if all S PDFs satisfying $\tilde{p}(Y_n^r|x_0)$ are in $\mathbf{\Gamma}_k^s$, the likelihood $p(y_j^r|x_0)$ can be isolated from the incoming PDF $p(x_0, I_j^r)$.

A packet classified as an incomplete arrival is moved from the current AB to the next AB. As delayed measurements become available, all incomplete arrivals are eventually upgraded to complete arrival. Classification as a complete arrival results in the packet being moved to the isolation stage.

The purpose of the *isolation stage* is to isolate information in the complete arrival that is not yet contained in the CP. Due to the classification stage, the only new information that will be available will be related to the most recent source NTP(i, r). The isolated likelihood is

$$p(y_i^r|x_0) = \frac{p(x_0, I_{\lambda_r+1}^r)}{\tilde{p}(Y_{\lambda_1}^s|x_0) \times \ldots \times \tilde{p}(Y_{\lambda_S}^s|x_0) \times p(x_0)}, \tag{11}$$

which is generated through the use of the CLM elements. This likelihood, along with the corresponding UTV ($\bar{\mathbf{\Lambda}}_i^r$) are passed to the fusion stage for further processing.

The *fusion* stage blends new information with the old. Firstly, the two components of the CP which describe the PDF of the likelihood are updated by

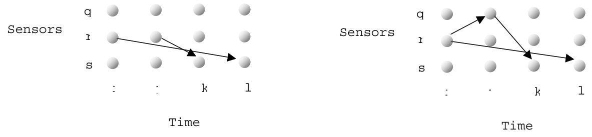
$$\bar{p}(x_0|I_k^s) = \bar{p}(x_0|I_k^s) \times p(y_i^r|x_0), \tag{12}$$

remembering that $\bar{p}(x_0, I_k^s)$ can be a range of intermediate values in the process of updating $p(x_0, I_{k-1}^s)$ to $p(x_0, I_k^s)$. Inclusion of this new information is acknowledged by the update $\lambda_r = i$ where $\lambda_r \in \bar{\mathbf{\Lambda}}_k^s$.

The previous updating related directly to improving the current estimate, whilst avoiding the data incest. An equally important aspect of the fusion stage is the provision for future estimates to be incest free. This requires the updating of the CLM with

$$\tilde{p}(Y_i^r|x_0) = \tilde{p}(Y_{i-1}^r|x_0) \times p(y_i^r|x_0). \tag{13}$$

This concludes the amalgamation of the completed arrival in question.



(a) Incomplete Direct Arrival (IDA)

(b) Incomplete Indirect Arrival (IIA)

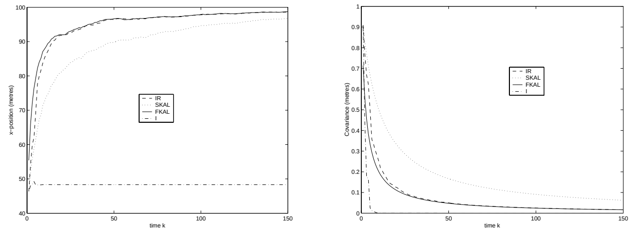
Fig. 4. INCOMPLETE ARRIVALS

At this point in the process, the final decision diamond in Figure 3 is implemented. If the complete arrival just processed was the last in the AB, the algorithm takes the next one in the queue and begins again at the first decision diamond.

For the case where more pre-classified packets exist in the AB, the processing for NTP(s, k) is completed. In this case the CP automatically becomes the OP (which will be the LP for NTP$(s, k+1)$) to be sent to all other sensors. The CLM now remains fixed at $\mathbf{\Gamma}_{k+1}^s$ while the arrival vector $\mathbf{A}_{k+1}^s$ will begin accepting incoming packets from remote sensors until the next processing time $k + 1$.

5. RESULTS

In an example simulation the estimated mean was initialized at 40m and is tracking a static state of 100m, while the covariance was initialized at 1m with the measurement noise variance fixed at 50m. The minimum delay is 1 and the maximum is 3. Figure 5 displays the results of simulation. The *SKAL* line represents a simple Kalman filter using only the measurements taken from the local sensor. *FKAL* involves all measurements from all sensors. *I* simply fuses the incoming PDFs with the current PDFs without isolating the new information. *IR* is the result of the aforementioned philosophy where the incest is removed. As anticipated, the covariance of the IR case, is bounded below by the FKAL case. Discrete points where they touch implies that all packets, from all S sensors, have arrived at this particular node and have been fused. This validates the fusion strategy.



(a) Estimated mean of x_0

(b) Estimated cov. of x_0

Fig. 5. VITAL STATISTIC COMPARISON

6. REFERENCES

[1] Y. Bar-Shalom, editor. *Multitarget-Multisensor Tracking: Advanced Applications.*, volume 1. ARTECH HOUSE, 1990.

[2] L. J Sciacca and R. J. Evans. Cooperative sensor networks with bandwidth constraints. In *Proceedings of Aerosense SPIE Conference, Battlespace Digitisation and Network-Centric Warfare*, Orlando Florida, April 2002.

[3] B. D. O. Anderson and J. B. Moore. *Optimal Filtering*. Prentice-Hall, 1979.

[4] J. S. Evans and V. Krishnamurthy. Hidden markov model signal processing over packet switched networks. *IEEE Transactions Signal Processing*, Vol. 47, No. 8:pp.2157–2166, August 1999.

[5] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, Inc., Orlando, Florida, 1988.