Estimating and Exploiting the Degree of Independent Information in Distributed Data Fusion

Simon J. Julier
Virtual Environments and Computer Graphics Group, Department of Computer Science,
University College London, Gower Street, London WC1E 6BT, UK.
S.Julier@cs.ucl.ac.uk

Abstract – *Double counting is a major problem in distributed data fusion systems. To maintain flexibility and scalability, distributed data fusion algorithms should just use local information. However globally optimal solutions only exist in highly restricted circumstances. Suboptimal algorithms can be applied in a far wider range of cases, but can be very conservative.*

In this paper we present preliminary work to develop distributed data fusion algorithms that can estimate and exploit the correlations between the estimates stored in different nodes in a distributed data fusion network. We show that partial information can be modelled as kind of "overweighted" Covariance Intersection algorithm. We motivate the need for an adaptive scheme by analysing the correlation behaviour of a simple distributed data fusion network and show that it is complicated and counterintuitive. Two simple approaches to estimate the correlation structure are presented and their results analysed. We show that significant advantages can be obtained.

Keywords: Tracking, filtering, estimation, distributed data fusion, covariance intersection, bounded covariance inflation, unmanned aerial vehicles.

1 Introduction

In many military and civilian applications, mobile and distributed sensor networks have the potential to revolutionise the way in which information is collected, fused and disseminated. For example, we are developing a system that uses a swarm of miniature quadrotor helicopters to aid in search and rescue operations [1]. Each quadrotor contains a sensing system (camera, GPS, inertial sensors) and can communicate wirelessly with one another and with a base station. This application can naturally be described as an instance of a distributed data fusion network (DDFN). A DDFN consists of a set of fusion nodes. Each node is equipped with zero or more sensors and can fuse information collected locally or disseminated by other nodes. A key assumption is that no single node has global knowledge of the topology or state of the entire DDFN. Instead, nodes only have access to and exploit1 *local* information. The locality assumption is central to many of the advantages of DDFNs including scalability, modularity and graceful degradation of performance in the presence of failures [2].

However, these advantages come at the risk of *double counting*: the same observation information can be implicitly used repeatedly [3], leading to over-confident estimates. Optimal solutions to this problem can be implemented locally only if the DDFN is guaranteed to be in either a fully-connected or tree-connected network topology [2]. These topologies are highly restrictive, and greatly limit the flexibility and generality of DDFNs.

Uhlmann argued that the difficulties arise from seeking an *optimal* solution [4], and proposed a principled suboptimal algorithm known as *Covariance Intersection* (CI) [4, 5]. Although CI is guaranteed to yield consistent estimates for *any* degree of correlation, it can lead to highly conservative estimates. Therefore, a number of algorithms including *Split Covariance Intersection* (SCI) [6] and *Bounded Covariance Inflation* (BCInf) [7] have been developed. These exploit available information about known independence in the network. However, quantifying this information is not always easy, especially in large and time-varying networks.

In this paper we present preliminary work which examines whether it is it possible to estimate and exploit the degree of dependence between the nodes. The structure is as follows. The problem statement is outlined in the next section. Section 3 considers the problem of developing low-order robust DDF algorithms that can exploit partial independent information. We show that the BCInf algorithm is a special case of the SCI algorithm, and propose a general class of algorithms for suboptimal fusion that "overweight" covariance intersection. We investigate the performance of the BCInf in a simulated network in Section 4 and show that the parameter value varies over time and that adapting could have impact on the performance of the system. Two adaptation strategies are discussed in Section 5 and conclusions drawn in Section 6.

2 Problem Statement

Suppose a pair of platforms A and B track a common target T. The state of T at the discrete time step k is the state vector $\mathbf{x}(k)$. The estimate of $\mathbf{x}(k)$ using observations up to time step j is $\{\hat{\mathbf{x}}(k \mid j),\ \mathbf{P}(k \mid j)\}$ where $\hat{\mathbf{x}}(k \mid j)$ is the estimated mean and $\mathbf{P}(k \mid j)$ the estimated covariance. Given that the error in the estimate is

$$\tilde{\mathbf{x}}(k \mid j) = \mathbf{x}(k) - \hat{\mathbf{x}}(k \mid j), \tag{1}$$

the estimate is said to be *covariance consistent* [8] if it obeys the condition

$$\mathbf{P}(k \mid j) - \mathrm{E}\left[\tilde{\mathbf{x}}(k \mid j)\,\tilde{\mathbf{x}}^T(k \mid j)\right] \geq \mathbf{0}, \tag{2}$$

where $\geq \mathbf{0}$ means that the difference is positive semidefinite.

Consider the state of the system at the end of time step $k-1$. The estimate of T at node A is expressed as the mean and covariance $\{\hat{\mathbf{x}}_A(k-1 \mid k-1),\ \mathbf{P}_A(k-1 \mid k-1)\}$. The estimate at node B is $\{\hat{\mathbf{x}}_B(k-1 \mid k-1),\ \mathbf{P}_B(k-1 \mid k-1)\}$. The objective is to compute covariance consistent estimates $\{\bar{\mathbf{x}}_A(k \mid k),\ \bar{\mathbf{P}}_A(k \mid k)\}$ and $\{\bar{\mathbf{x}}_B(k \mid k),\ \bar{\mathbf{P}}_B(k \mid k)\}$. Each node should exploit information collected both locally at a node and information transmitted to it from other nodes. The steps required to complete this at node A are:

1. A uses its internal process model and the standard Kalman filter prediction equations to compute the prediction $\{\hat{\mathbf{x}}_A(k \mid k-1),\ \mathbf{P}_A(k \mid k-1)\}$.
2. A applies the Kalman filter update equations to fuse its prediction with its observation $\{\mathbf{z}_A(k), \mathbf{R}_A(k)\}$ to form the *distributed estimate* $\{\hat{\mathbf{x}}^*_A(k \mid k),\ \mathbf{P}^*_A(k \mid k)\}$[1].
3. A propagates its distributed estimate to B, and B propagates its distributed estimate $\{\hat{\mathbf{x}}^*_B(k \mid k),\ \mathbf{P}^*_B(k \mid k)\}$ to A.
4. A uses the distributed data fusion algorithm to fuse its prediction with B's distributed estimate to form the *partial update* $\{\hat{\mathbf{x}}^+_A(k \mid k),\ \mathbf{P}^+_A(k \mid k)\}$.
5. A uses a Kalman filter to fuse its observations $\{\mathbf{z}_A(k), \mathbf{R}_A(k)\}$ with the partial update to yield the final update $\{\bar{\mathbf{x}}_A(k \mid k),\ \bar{\mathbf{P}}_A(k \mid k)\}$.

The difficulty in implementing this algorithm lies in step 4. The reason is that, in general, $\{\hat{\mathbf{x}}^*_B(k \mid k),\ \mathbf{P}^*_B(k \mid k)\}$ and $\{\hat{\mathbf{x}}_A(k \mid k-1),\ \mathbf{P}_A(k \mid k-1)\}$ are not independent of one another. There are two reasons for this. First, the estimates have become correlated if the nodes have exchanged estimates of T in the past. Second, because both nodes are tracking the same target, the process noise is the same in both nodes. However, failure to account for this dependency leads to double counting. Estimates become over-confident, and can lead to catastrophic filter failure. The optimal solution to the double counting problem is to calculate and cancel out common information between the nodes [3]. However, in arbitrary network topologies this can only be carried out if nodes have global knowledge of the network topology and the estimate within each node. This undermines the flexibility and scalability of DDFN.

An alternative approach is to develop *suboptimal* distributed data fusion algorithms.

[1] Note that, in general, the state space of the distributed estimate could differ from the state space of A. One reason for this is that, to reduce bandwidth, only a subset of the state such as the position of the target might be distributed. In many situations, these relationships are linear, and this justifies our assumption of linearity in (3).

3 Suboptimal Distributed Data Fusion

3.1 Covariance Intersection

Consider the problem of fusing the distributed estimate into the local estimate. Suppressing time indices for convenience, suppose that the distributed estimate from B is related to the state at A according to the linear equation

$$\mathbf{x}^*_B = \mathbf{H}_B \mathbf{x}_A. \tag{3}$$

The inverse covariance form of the Kalman Filter (KF) update equation is[2]

$$\begin{aligned} \bar{\mathbf{P}}_A^{-1} &= \mathbf{P}_A^{-1} + \mathbf{H}_B^T \mathbf{P}_B^{-1} \mathbf{H}_B, \\ \bar{\mathbf{P}}_A^{-1} \bar{\mathbf{x}}_A &= \mathbf{P}_A^{-1} \hat{\mathbf{x}}_A + \mathbf{H}_B^T \mathbf{P}_B^{-1} \hat{\mathbf{x}}_B. \end{aligned} \tag{4}$$

However, as explained above the estimates are not independent of one another and this equation cannot be used. The Covariance Intersection (CI) algorithm replaces the addition with the convex combination [4, 5]

$$\begin{aligned} \bar{\mathbf{P}}_A^{-1} &= \omega \mathbf{P}_A^{-1} + (1-\omega) \mathbf{H}_B^T \mathbf{P}_B^{-1} \mathbf{H}_B, \\ \bar{\mathbf{P}}_A^{-1} \bar{\mathbf{x}}_A &= \omega \mathbf{P}_A^{-1} \hat{\mathbf{x}}_A + (1-\omega) \mathbf{H}_B^T \mathbf{P}_B^{-1} \hat{\mathbf{x}}_B. \end{aligned} \tag{5}$$

If $\{\hat{\mathbf{x}}_A, \mathbf{P}_A\}$ and $\{\hat{\mathbf{x}}_B, \mathbf{P}_B\}$ are covariance consistent, $\{\bar{\mathbf{x}}_A, \bar{\mathbf{P}}_A\}$ will be covariance consistent for *any* correlation between A and B and *any* choice of $\omega \in [0, 1]$.

[2] When A updates with distributed measurements from a set of nodes $\mathcal{N}_A$, this equation generalises to

$$\begin{aligned} \bar{\mathbf{P}}_A^{-1} &= \mathbf{P}_A^{-1} + \sum_{n \in \mathcal{N}_A} \mathbf{H}_n^T \mathbf{P}_n^{-1} \mathbf{H}_n, \\ \bar{\mathbf{P}}_A^{-1} \bar{\mathbf{x}}_A &= \mathbf{P}_A^{-1} \hat{\mathbf{x}}_A + \sum_{n \in N_A} \mathbf{H}_n^T \mathbf{P}_n^{-1} \hat{\mathbf{x}}_n. \end{aligned}$$

All the other update equations discussed in this paper generalise to the multiple distributed observation case in a similar fashion.

Although CI can be used to perform consistent fusion in arbitrary network topologies [6], it can lead to highly conservative estimates. The results for the simulation scenario presented in Section 4, for example, show that the mean covariance of the CI estimate can be 60% greater than that of the globally optimal solution. The reason for this poor performance is that CI is robust to *any* correlation structure which could arise between A and B. These include highly degenerate cases, such as B immediately sending A's distributed straight back. However, such cases would rarely (if ever) happen in practice. Therefore, several algorithms have been developed which exploit *partial independence* information. These include *Split Covariance Intersection* and *Bounded Covariance Information*.

3.2 Split Covariance Intersection

Split Covariance Intersection (SCI) exploits the assumption that the error in the estimates can be decomposed into two mutually independent components [6]

$$\tilde{\mathbf{x}}_A = \tilde{\mathbf{x}}_A^1 + \tilde{\mathbf{x}}_A^2; \quad \tilde{\mathbf{x}}_B = \tilde{\mathbf{x}}_B^1 + \tilde{\mathbf{x}}_B^2, \tag{6}$$

where the correlations between $\tilde{\mathbf{x}}_A^1$ and $\tilde{\mathbf{x}}_B^1$ are unknown, but the correlations between $\tilde{\mathbf{x}}_A^2$ and $\tilde{\mathbf{x}}_B^2$ are known. Furthermore, $\tilde{\mathbf{x}}_A^1$ and $\tilde{\mathbf{x}}_B^1$ are each independent of both $\tilde{\mathbf{x}}_A^2$ and $\tilde{\mathbf{x}}_B^2$.

Given a system of this form, the inverse covariance form of the SCI update equations are

$$\begin{aligned}
\bar{\mathbf{P}}_A^{-1} &= \omega \left(\mathbf{P}_A^1 + \omega \mathbf{P}_A^2\right)^{-1} \\
&\quad + (1-\omega)\mathbf{H}_B^T \left(\mathbf{P}_B^1 + (1-\omega)\mathbf{P}_B^2\right)^{-1} \mathbf{H}_B \\
\bar{\mathbf{P}}_A^{-1}\bar{\mathbf{x}}_A &= \omega \left(\mathbf{P}_A^1 + \omega \mathbf{P}_A^2\right)^{-1} \hat{\mathbf{x}}_A \\
&\quad + (1-\omega)\mathbf{H}_B^T \left(\mathbf{P}_B^1 + (1-\omega)\mathbf{P}_B^2\right)^{-1} \hat{\mathbf{x}}_B
\end{aligned}$$

Note that both the KF and CI update rules arise as special cases when $\{\mathbf{P}_A^1, \mathbf{P}_B^1\} = \mathbf{0}$ and $\{\mathbf{P}_A^2, \mathbf{P}_B^2\} = \mathbf{0}$ respectively. It can be shown that $\bar{\mathbf{P}}_A$ will be covariance consistent if $\mathbf{P}_A^2$ and $\mathbf{P}_B^2$ *under*-estimate the covariance of the independent component, but the sums $\mathbf{P}_A = \mathbf{P}_A^1 + \mathbf{P}_A^2$ and $\mathbf{P}_B = \mathbf{P}_B^1 + \mathbf{P}_B^2$ *over*-estimate the mean squared error of the error vectors $\tilde{\mathbf{x}}_A$ and $\tilde{\mathbf{x}}_B$.

Although the SCI algorithm can optimally exploit available information irrespective of its form, there are several limitations to its use. First, it can only be applied if the estimation errors can be decomposed into the form specified in (6). Although this can often be achieved naturally within a single node, maintaining the split structure across the entire network requires the calculation of common information. A second difficulty is that the estimate requires the manipulation of two covariance matrices and, in some applications, this could be prohibitively expensive.

3.3 Bounded Covariance Inflation

Bounded Covariance Inflation (BCInf) was developed as an alternative to SCI to reduce the computational costs [7]. The algorithm exploits the assumption that an *upper bound* on the absolute value of the cross correlations between the estimates can be established. This approach was was proposed by Hanebeck [9] and independently developed by Reece and Roberts [7].

Consider the *joint covariance matrix* which specifies the correlation structure between A and B,

$$\mathbf{J}_O = \begin{bmatrix} \mathbf{P}_A & \mathbf{X}_{AB} \\ \mathbf{X}_{AB}^T & \mathbf{P}_B \end{bmatrix}. \tag{7}$$

The cross-correlation matrix $\mathbf{X}_{AB}$ is not known. However, suppose it is known that the maximum absolute value of any cross correlation coefficient cannot exceed S. In other words,

$$S^2\mathbf{P}_B - \mathbf{X}_{AB}^T\mathbf{P}_A^{-1}\mathbf{X}_{AB} \geq \mathbf{0}. \tag{8}$$

The BCInf exploits this condition to develop the update equations

$$\begin{aligned}
\bar{\mathbf{P}}_A^{-1} &= \omega_A \mathbf{P}_A^{-1} + \omega_B \mathbf{H}_B^T \mathbf{P}_B^{-1} \mathbf{H}_B, \\
\bar{\mathbf{P}}_A^{-1}\bar{\mathbf{x}}_A &= \omega_A \mathbf{P}_A^{-1}\hat{\mathbf{x}}_A + \omega_B \mathbf{H}_B^T \mathbf{P}_B^{-1}\hat{\mathbf{x}}_B,
\end{aligned} \tag{9}$$

where

$$\begin{aligned}
\omega_A &= \omega \left(\omega + (1-\omega)S\right)^{-1}, \\
\omega_B &= (1-\omega)\left((1-\omega) + S\omega\right)^{-1}
\end{aligned} \tag{10}$$

and $\omega \in [0,1]$. Again, the KF and CI update rules arise as special cases when $S = 0$ and $S = 1$ respectively. The update is guaranteed to be covariance consistent for any $\omega \in [0,1]$ as long as S is an *upper bound* on the cross correlation.

Despite their apparent dissimilarity, BCInf can be shown to be a special case of SCI.

3.4 Relationship Between BCInf, SCI, and CI

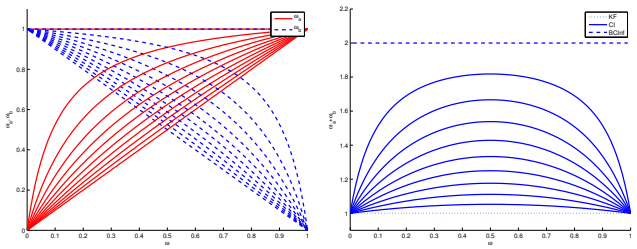
Theorem 1. *The BCInf update is a special case of the SCI update equation under the assumption that the error structures for $\tilde{\mathbf{x}}_A$ and $\tilde{\mathbf{x}}_B$ are*

$$\begin{aligned}
\tilde{\mathbf{x}}_A^1 &= \left(\sqrt{S}\right)\tilde{\mathbf{x}}_A; \quad \tilde{\mathbf{x}}_A^2 = \left(\sqrt{1-S}\right)\tilde{\mathbf{x}}_A \\
\tilde{\mathbf{x}}_B^1 &= \left(\sqrt{S}\right)\tilde{\mathbf{x}}_B; \quad \tilde{\mathbf{x}}_B^2 = \left(\sqrt{1-S}\right)\tilde{\mathbf{x}}_B
\end{aligned}$$

where S is the upper bound cross correlation as specified in (8).

Proof. Using the noise assumptions for SCI, the covariances of the different components can be written as

$$\begin{aligned}
\mathbf{P}_A^1 &= S\mathbf{P}_A, \ \mathbf{P}_A^2 = (1-S)\mathbf{P}_A, \\
\mathbf{P}_B^1 &= S\mathbf{P}_B, \ \mathbf{P}_B^2 = (1-S)\mathbf{P}_B.
\end{aligned}$$



(a) Plots of ω_A (solid) and ω_B (dashed).

(b) Plot of $\omega_A + \omega_B$.

Figure 1: The behaviour of $\omega_A(\omega, S)$ and $\omega_B(\omega, S)$ for different choices of $\omega = [0 : 0.1 : 1]$ and $S = [0 : 0.1 : 1]$.

Substituting into the SCI update,

$$\begin{aligned}\bar{\mathbf{P}}_A^{-1} &= \omega\left(\mathbf{P}_A^1 + \omega\mathbf{P}_A^2\right)^{-1} \\ &\quad + (1-\omega)\mathbf{H}_B^T\left(\mathbf{P}_B^1 + (1-\omega)\mathbf{P}_B^2\right)^{-1}\mathbf{H}_B \\ &= \omega\left(S\mathbf{P}_A + \omega(1-S)\mathbf{P}_A\right)^{-1} \\ &\quad + (1-\omega)\mathbf{H}_B^T\left(S\mathbf{P}_B + (1-\omega)(1-S)\mathbf{P}_B\right)^{-1}\mathbf{H}_B \\ &= \omega\Big(S + \omega(1-S)\Big)^{-1}\mathbf{P}_A^{-1} \\ &\quad + (1-\omega)\Big(S + (1-\omega)(1-S)\Big)^{-1}\mathbf{H}_B^T\mathbf{P}_B^{-1}\mathbf{H}_B\end{aligned}$$

By inspection, the coefficients are the same as those in (10). We can similarly show that the mean update equation is the same as BCInf. □

A further relationship can be drawn between the KF, CI and BCInf. All three algorithms can be written in the form of (9) but with different choices for the weights — for the KF $\{\omega_A = 1,\ \omega_B = 1\}$, for CI $\{\omega_A = \omega,\ \omega_B = (1-\omega)\}$ and for BCInf they are given in (10). The behaviour of these functions for different values of S and ω are plotted in Figure 1. Two behaviours are apparent:

1. $0 \leq \omega_A(\omega, S), \omega_B(\omega, S) \leq 1\ \forall\ (\omega, S)$. Therefore, each term is given a non-negative weight, but this cannot exceed one.

2. $1 \leq \max_\omega\left(\omega_A(\omega, S) + \omega_B(\omega, S)\right) \leq 2\ \forall\ S$. Therefore, the sum of the weights are bounded from both below and above. These correspond to the CI and KF cases respectively.

In summary, we have discussed suboptimal fusion algorithms and we have shown that the KF and BCInf can be considered to be generalisations of CI which "overweight" the information from A and B, However, the BCInf algorithm requires a value of S. In the next section, we consider the behaviour of this parameter in a simulation scenario.

4 Behaviour of the Correlation Coefficient

Consider the situation shown in Figure 2: a set of six platforms $P_1, \ldots, P_6$ track a moving target T. T is largely constrained to line on the ground. We assume the platforms can measure their pose precisely. The state of the target is its 3D position and velocity

$$\mathbf{x}(k) = \begin{bmatrix} x(k) & \dot{x}(k) & y(k) & \dot{y}(k) & z(k) & \dot{z}(k)\end{bmatrix}.$$

The motion of the target in the x, y and z directions are described by three independent and damped random walk processes. In the x-direction, for example, the discrete time process model is

$$\begin{aligned}x(k+1) &= \alpha_x\left[x(k) + \Delta T\dot{x}(k)\right] + v_x(k) \\ \dot{x}(k+1) &= \alpha_x\dot{x}(k) + v_{\dot{x}}(k),\end{aligned}$$

where the process noise covariance is

$$\mathrm{E}\left[\begin{pmatrix}v_x(k)\\ v_{\dot{x}}(k)\end{pmatrix}\begin{pmatrix}v_x(k)\\ v_{\dot{x}}(k)\end{pmatrix}^T\right] = \begin{bmatrix}\Delta T^3/3 & \Delta T^2/2\\ \Delta T^2/2 & \Delta T\end{bmatrix}Q_x^2.$$

We use the parameters $\alpha_x = \alpha_y = 0.98, \alpha_z = 0.8$ to allow significant movement in the xy-plane but to strongly restrict movement in the z-direction. The noise terms are $Q_x = Q_y = Q_z = 1$.

The platforms are equipped with an ideal "displacement" sensor and use a "sequential communication topology". Both of these are unrealistic in practice, but were chosen to highlight the effects of a time varying network topology.

The displacement sensor for the ith platform measures the relative translation between the position of the target and the position of the UAV $(X_i,\ Y_i,\ Z_i)$,

$$\mathbf{z}_i^d(k) = \begin{pmatrix}x(k) - X_i(k)\\ y(k) - Y_i(k)\\ z(k) - Z_i(k)\end{pmatrix} + \mathbf{w}(k),$$

where $\mathbf{w}(k)$ is the observation noise. This is modelled as a zero-mean, Gaussian-distributed random whose covariance is the identity matrix.

Although this sensor model greatly simplifies the filtering problem (by making it linear), it also presents the DDFN algorithms with the worst case scenario that the covariance matrices for all nodes will be very similar. This means that the suboptimal algorithms, such as CI, which exploit differences in the shapes of covariance ellipses from the different nodes, will have the least amount of information to work with.

The action of the sequential communication topology is illustrated in Figure 2. Initially, no nodes are in communication. However, every 10s a new node joins the network which eventually forms a closed ring. This

topology makes it possible to see the effects of systematically entering the nodes into the network, as well as the effect of the loop closure. The distributed estimates communicated between each node are the positions of the target,

$$\mathbf{x}_i^*(k) = \begin{bmatrix} x(k) & y(k) & z(k) \end{bmatrix}^T.$$

The following DDFN algorithms were implemented:

- **No DDF (NDDF).** No distributed estimates are created or used. This is the worst case.
- **Globally Optimal (GO).** The full correlation structure is maintained and no approximations are required. It is implemented by stacking up the state estimates of all the nodes into a single state space. Schmidt-Kalman filters are used to selectively update subsets of states to simulate the fact that, for example, a distributed estimate only updates the node which receives it.
- **Covariance Intersection (CI).** As explained above, this linear network is a worst case scenario, and this algorithm is the baseline against which the suboptimal algorithms will be compared. The batch CI form [6] was used and the ω values were chosen to minimise the determinant.
- **Globally Optimal BCInf (GOBCInf).** This algorithm uses the same joint structure as the GO filter to maintain and update the cross correlations between all of the state estimates. However, rather than apply an optimal KF, the BCInf algorithm is used. S was computed as the maximum cross correlation coefficient of all distributed estimates. The same value of S was used[3] and the ω values were chosen to minimise determinant. This algorithm represents the best performance which can be achieved by the BCInf.
- **Conservative BCInf (CBCInf).** The BCInf algorithm is implemented for each node using a fixed value for all updates for all distributed estimates. We used the value $S = 0.55$, which is an upper bound on the history of S computed by GOBCInf. This algorithm is representative of a hand-tuned heuristic.

The performance of each filter was validated by computing the normalised estimation error squared (NEES) and ensuring that its value did not exceed the state dimension(six) [10].

The performance of the filters are plotted in Figure 3 for a single cycle of the sequential topology. The most

[3]This is sufficient for this linear example but, in general, a different value of S should be calculated for each pair of nodes.

Algorithm	Min	Mean	Max
NODDF	0.74	0.74	0.74
CI	0.73	0.73	0.75
GO	0.48	0.54	0.74
GOBCInf	0.49	0.57	0.74
CBCInf	0.58	0.62	0.75
ABCInf	0.54	0.63	0.75

Table 1: Average covariance of the x-state for the target in node 1 for each algorithm.

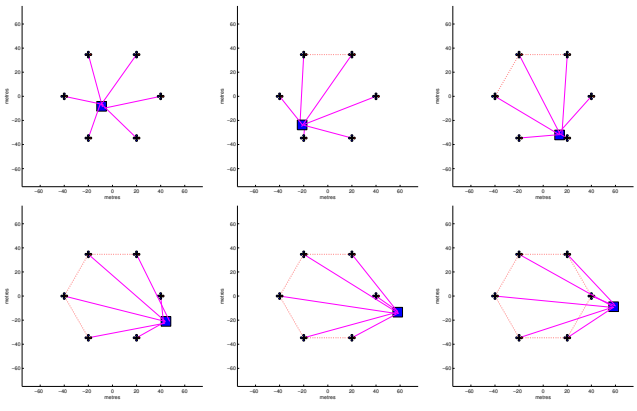
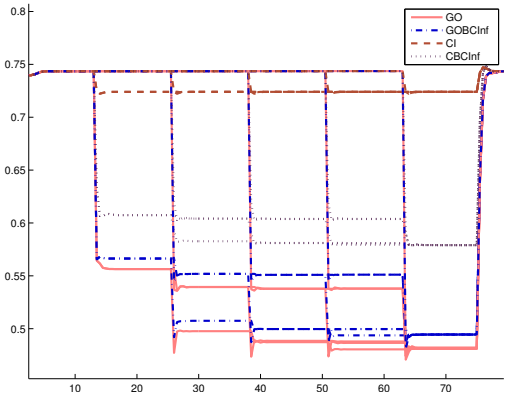
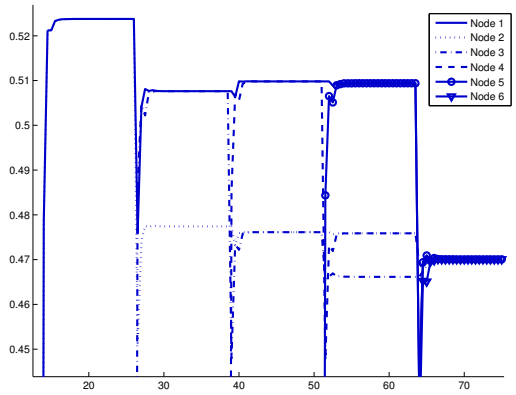


Figure 2: Initial simulation scenario. A set of static platforms (denoted as crosses) arranged in a ring observe the motion of a target (denoted as a rectangle) using a simple displacement sensor. The observations are shown as solid lines, the current communication connections as dashed lines.



(a) Covariance history. Note that due to aliasing, some of the lines appear mislabelled. In particular, the covariance of the CI filters do not fall below 0.7; the apparently dashed lines at approximately 0.55 are for the GOBCInf algorithm.



(b) S history for GOBCInf. This can only be defined for a node once it has entered the DDFN.

Figure 3: Time histories of the covariance of x for all filters, and S history for the GOBCInf algorithm.

obvious features are that all of the DDFN algorithms exhibit a series of pronounced vertical bars. These occur when a node P_N enters the network and begins to communicate with its neighbours. Two related events occur. First, P_N gains access to observation information (collected from nodes P_1 to P_{N-1}) which it has not had access to before. This influx of new, independent information causes its covariance to sharply decline. Second, P_N contributes its own collected observation information (which is independent from the estimates at nodes P_1 to P_{N-1}). This can most clearly be seen as a sharp momentary drop in the GO curve at about 25s. These events can be seen in the history for S. When the nodes first begin to communicate, the value is low (reflecting the independent observations) but not zero (because of the common process noise). As the nodes exchange information, their estimates rapidly become correlated with one another and their S values converge.

The graphs also show that, as the simulation progresses and more nodes enter the network, the overall covariance for many of the DDFN algorithms continue to decline, but at a decreasing rate. However, the S curve shows the seemingly anomalous behaviour that, when P_3 enters the network, all the S values *decline*. The reason for this is a property of the change in the network topology itself:

- Between 15s and 25s, only P_1 and P_2 communicate with one another. They directly exchange distributed estimates with one another at each time step and so the state in both is the same. Therefore, their covariance and S values are identical.

- Between 25s and 35s, the network now consists of P_1, P_2 and P_3. However, there are now two sets of values for the covariances and S values. These reflect the fact that the connection topology is not the same for all nodes. Nodes P_1 and P_3 only communicate with a single node (P_2) but P_2 communicates with two nodes (P_1 and P_3). As a result, the covariances of P_1 and P_3 are the same, and the covariance of P_2 is significantly smaller. The S value declines as well because the correlation between P_1 and P_2 is less — this reflects the fact that, at each time step, the estimate at P_2 includes the distributed information from P_3 which will not propagate to P_1 until the next time step.

- As further nodes enter the network, the network topology continues to change further. When there are five nodes, for example, there are three different correlation topologies (P_1 and P_5; P_2 and P_4; P_3).

- Wen the loop is closed at 65s, the connection topology becomes the same for all nodes and their covariance and S values all converge to the same value.

The NODDF and CI algorithms do not exhibit these behaviours. The NODDF algorithm does not distribute information. Because of the similarity of the structure of the CI algorithms, the performance of a node improves when it enters the network, but this improvement is fixed and does not change with the number of nodes.

The impacts of these behaviours in terms of the quantitative performances of the algorithms are shown in Table 1. Not surprisingly, the NODDF algorithm performs most poorly. CI shows a very slight improvement. The performance of the GOBCInf algorithm is very close to that of GO. The CBCInf algorithm is significantly better than CI, but it is clear that there is significant scope for improvement.

In summary, even this simple example shows that the behaviour of a DDFN is a function of the *global* network topology and can be both complicated and counterintuitive. Furthermore, the results show that BCInf can provide robust estimates that are close to optimal. Furthermore, a simple upper bound tuning heuristic can be sufficient. However, it is not necessarily clear how the upper bound can be computed. Furthermore, the results also suggest that further performance improvements could be achieved if S could be adapted automatically in response to changes in correlation structure within the DDFN. Although a point solution has been developed for the case of a distributed mapping application [7], closed form solutions for arbitrary network topologies cannot be computed and methods for estimating S must be developed.

5 Adaptive Bounded Covariance Inflation

The problem of developing an Adaptive BCInf (ABCInf) algorithm can be posed as follows. Given the set of all measurements available at node i up to time k,, $\mathbf{Z}_i^k$, compute an estimate of the upper bound cross correlation coefficient. Two approaches have been considered — a probabilistic Rao-Blackwellised approach, and a NEES matching adapted from self-tuning Kalman filters.

5.1 Probabilistic Approach

In the probabilistic approach, we assume that $S_i(k)$ is a random variable whose value evolves over time to reflect the changes in the network topology and in the state within each node. The presence of the BCInf algorithm with its optimisation step means that the functions are nonlinear and analytic solutions cannot be derived. Therefore, we used a Rao-Blackwellised particle filter. The distribution of S is represented as a set of weighted particles. Conditioned on each value for S, each node maintains several mean and covariance estimates for T. As a result, our implementation resembles an MHT filter [10], with each hypothesis corresponding to one of the particle values for $S_i(k)$. We used the assumption of Gaussian distributions throughout. The algorithm was run in two steps. First, to construct the distributed estimates, the local measurements were fused to create a distributed estimate particle set. The weights of this set were updated using the observation as well. The mean and covariance of this set was computed using the conventional multiple model MHT equations [10]. Each node fused the *same* distributed estimate into its particle set to create a set of partial updates. The local measurements were then fused, the weights updated, and residual resampling applied if the effective number of particles was less than 75% of the total number of particles.

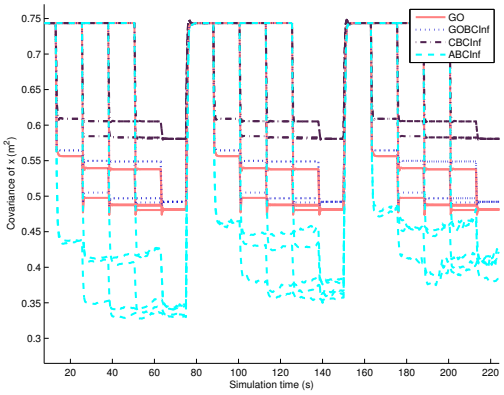


Figure 4: Covariance history for the probabilistic adaptation scheme.

The results of this algorithm using 10 particles are shown in Figure 4. As can be seen, the adaptive scheme is biased towards over estimating the degree of independence. We investigated a number of modifications to the algorithm including changes in the number of particles, the state dynamics, bounds on the upper and lower state values, and the use of alternative measurement likelihood distributions such as the Student's-t Distribution. However, all of these methods yielded qualitatively the same results.

5.2 Deterministic Approach

The problem of estimating S can also be viewed as a kind of noise estimation problem and many techniques have been developed to create self-tuning Kalman filters [8]. The basic technique is to adapt the level of noise based on the normalised innovation until its long-term average value equals the dimension of the state. Specifically, given the innovation vector $\boldsymbol{\nu}(k+1)$ and its predicted covariance $\mathbf{S}(k+1)$ from the KF equations at time $k+1$, the normalised innovation is

$$q(k) = \boldsymbol{\nu}^T(k+1)\mathbf{S}^{-1}(k+1)\boldsymbol{\nu}(k+1).$$

The BCInf parameterisation offers a particularly simple relationship between $S_i(k)$ and $q(k)$. If $S_i(k)$ is an underestimate of the actual cross correlation, the covariance of the partial update $\mathbf{P}_i^+(k \mid k)$ will not be consistent. As a result, $\mathbf{S}(k+1)$ will not be a consistent estimate of the covariance of $\boldsymbol{\nu}(k+1)$. Conversely, if $S_i(k)$ is overestimates the coefficient, and so both $\mathbf{P}_i^+(k \mid k)$ and $\mathbf{S}(k+1)$ are conservative. Therefore, we implemented the simple linear update rule

$$\hat{S}(k \mid k) = \hat{S}(k \mid k-1) + \mu\left(\frac{q(k)}{n_z}\right)$$

where n_z is the dimension of the estimate and μ is a scale factor. We initialised all estimates to 1. We used the value $\mu = 0.02$.

The covariance histories using this adaptation algorithm are shown in Figure 5 for three cycles of the network. The first two cycles show significant "burn in" as the correlation values decline from their initial high value. The slow decline is due to the small value for μ. Although the estimates are not always less than those for CBCInf, none of the covariance values are smaller than the GO algorithm. NEES tests show that all filters are operating consistently. The quantitative results in Table 1 show a slight improvement in performance.

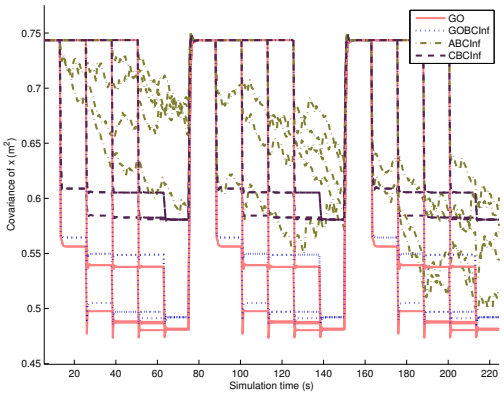


Figure 5: Covariance history for the deterministic adaptation scheme.

6 Conclusions

In this paper we have presented preliminary work to develop a DDF algorithm that automatically estimates and adapts to the level of independent information in a DDFN. To develop a low-order parameterisation of dependent information, we have shown that the KF, CI and BCInf approaches can be unified into a general "overweighting" formulation for CI. We have analysed the behaviour of the BCInf algorithm in a simple DDFN and shown that, even in this case, it can be complicated and counterintuitive. We discussed probabilistic and deterministic techniques for adaptation, and our results indicate that the deterministic technique is promising, but the probabilistic technique is not.

There are a number of avenues for future work. First, we shall continue to analyse the formulation of the probabilistic adaptation algorithm. We believe that its consistent bias towards overly optimistic estimates is likely to be the result of an incorrect formulation of the problem. Second, we shall extend the adaptation mechanism to include the distributed estimates. This means that adaptation will be able to occur even if sensor observations are not available at a given timestep. Finally, we shall implement these algorithms in non-trivial applications, such as camera-based tracking using swarms of small UAVs.

Acknowledgments

The work in this paper was supported under the EPSRC-funded project "SUAAVE: Sensing Unmanned Autonomous Aerial VEhicles" (EP/F064179/1). The author would also like to thank Simon Maskell at QinetiQ for a question which lead to the problem considered in this paper.

References

[1] W. T. L. Teacy, J. Nie, S. McClean, G. Parr, S. Hailes, S. Julier, N. Trigoni, and S. Cameron, "Collaborative Sensing by Unmanned Aerial Vehicles," in *Proceedings of the 3rd International Workshop on Agent Technology for Sensor Networks*, Budapest, Hungary, 2009.

[2] S. Grime and H. F. Durrant-Whyte, "Data fusion in decentralized sensor fusion networks," *Control Engineering Practice*, vol. 2, no. 5, pp. 849–863, 1994.

[3] M. E. L. II, C.-Y. Chong, I. Kadar, M. G. Alford, V. Vannicola, and S. Thomopoulos, "Distributed fusion architectures and algorithms for target tracking," *Proceedings of the IEEE*, vol. 85, no. 2, pp. 95–107, 1997.

[4] J. K. Uhlmann, "Dynamic map building and localization for autonomous vehicles," Ph.D. dissertation, University of Oxford, 1995.

[5] S. J. Julier and J. K. Uhlmann, "A Non-divergent Estimation Algorithm in the Presence of Unknown Correlations," in *Proceedings of the IEEE American Control Conference*, vol. 4, Albuquerque NM, USA, June 1997, pp. 2369–2373.

[6] ——, "General Decentralized Data Fusion With Covariance Intersection (CI)," in *Handbook of Data Fusion*, D. Hall and J. Llinas, Ed. Boca Raton FL, USA: CRC Press, 2001.

[7] S. Reece and S. Roberts, "Robust, Low-Bandwidth, Multi-Vehicle Mapping," in *Proceedings of the 2005 FUSION Conference*, Philadelphia, PA, USA, 2005, pp. 1319–1326.

[8] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Academic Press, San Diego, CA, 1970.

[9] U. D. Hanebeck, K. Brechle, and J. Horn, "A Tight Bound for the Joint Covariance of Two Random Vectors with Unknown but Constrained Cross-Correlation," in *Proceedings of the International Conference on Multisensor Fusion and Integration for Intelligent Systems, 2001. MFI 2001.*, Kongresshaus, Baden-Baden, Germany, 20–22 August 2001, pp. 85–90.

[10] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. New York NY, USA: Academic Press, 1988.