12

General Decentralized Data Fusion with Covariance Intersection (CI)

Simon Julier
IDAK Industries

Jeffrey K. Uhlmann
University of Missouri

12.1 Introduction

One of the most important areas of research in the field of control and estimation is decentralized (or distributed) data fusion. The motivation for decentralization is that it can provide a degree of scalability and robustness that cannot be achieved with traditional centralized architectures. In industrial applications, decentralization offers the possibility of producing plug-and-play systems in which sensors can be slotted in and out to optimize a tradeoff between price and performance. This has significant implications for military systems as well because it can dramatically reduce the time required to incorporate new computational and sensing components into fighter aircraft, ships, and other types of platforms.

The benefits of decentralization are not limited to sensor fusion onboard a single platform; decentralization also can allow a network of platforms to exchange information and coordinate activities in a flexible and scalable fashion that would be impractical or impossible to achieve with a single, monolithic platform. Interplatform information propagation and fusion form the crux of the network centric warfare (NCW) vision for the U.S. military. The goal of NCW is to equip all battlespace entities — aircraft, ships, and even individual human combatants — with communication and computing capabilities to allow each to represent a node in a vast decentralized command and control network. The idea is that each

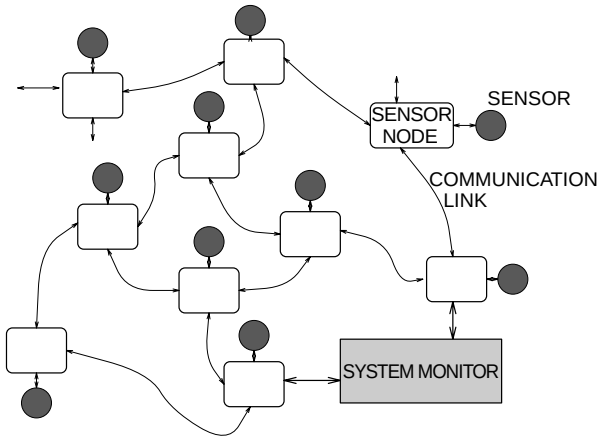


FIGURE 12.1 A distributed data fusion network. Each box represents a fusion node. Each node possesses 0 or more sensors and is connected to its neighboring nodes through a set of communication links.

entity can dynamically establish a communications link with any other entity to obtain the information it needs to perform its warfighting role.

Although the notion of decentralization has had strong intuitive appeal for several decades, achieving its anticipated benefits has proven extremely difficult. Specifically, implementers quickly discovered that if communications paths are not strictly controlled, pieces of information begin to propagate redundantly. When these pieces of information are reused (i.e., double-counted), the fused estimates produced at different nodes in the network become corrupted. Various approaches for avoiding this problem were examined, but none seemed completely satisfactory. In the mid-1990s, the redundant information problem was revealed to be far more than just a practical challenge; it is a manifestation of a fundamental theoretical limitation that could not be surmounted using traditional Bayesian control and estimation methods such as the Kalman filter.[1] In response to this situation, a new data fusion framework, based on covariance intersection (CI), was developed. The CI framework effectively supports all aspects of general decentralized data fusion.

The structure of this chapter is as follows: Section 12.2 describes the decentralized data fusion (DDF) problem. The CI algorithm is described in Section 12.3. Section 12.4 demonstrates how CI supports distributed data fusion and describes one such distribution architecture. A simple example of a network with redundant links is presented in Section 12.5. Section 12.6 shows how to exploit known information about network connectivity and/or information proliferation within the CI framework. This chapter concludes with a brief discussion of other applications of CI.

12.2 Decentralized Data Fusion

A decentralized data fusion system is a collection of processing nodes, connected by communication links (Figure 12.1), in which none of the nodes has knowledge about the overall network topology. Each node performs a specific computing task using information from nodes with which it is linked, but no "central" node exists that controls the network. There are many attractive properties of such decentralized systems,[2] including

- Decentralized systems are reliable in the sense that the loss of a subset of nodes and/or links does not necessarily prevent the rest of the system from functioning. In a centralized system, however, the failure of a common communication manager or a centralized controller can result in immediate catastrophic failure of the system.
- Decentralized systems are flexible in the sense that nodes can be added or deleted by making only local changes to the network. For example, the addition of a node simply involves the establishment of links to one or more nodes in the network. In a centralized system, however, the addition of a

new node can change the topology in such a way as to require massive changes to the overall control and communications structure.

The most important class of decentralized networks involves nodes associated with sensors or other information sources. Information from distributed sources propagates through the network so that each node obtains the data relevant to its own processing task. In a battle management application, for example, one node might be associated with the acquisition of information from reconnaissance photographs, another with ground-based reports of troop movements, and another with the monitoring of communications transmissions. Information from these nodes could then be transmitted to a node that estimates the position and movement of enemy troops. The information from this node could then be transmitted back to the reconnaissance photo node, which would use the estimated positions of troops to aid in the interpretation of ambiguous features in satellite photos.

In most applications, the information propagated through a network is converted to a form that provides the estimated state of some quantity of interest. In many cases, especially in industrial applications, the information is converted into means and covariances that can be combined within the framework of Kalman-type filters. A decentralized network for estimating the position of a vehicle, for example, could combine acceleration estimates from nodes measuring wheel speed, from laser gyros, and from pressure sensors on the accelerator pedal. If each independent node provides the mean and variance of its estimate of acceleration, fusing the estimates to obtain a better filtered estimate is relatively easy.

The most serious problem arising in decentralized data fusion networks is the effect of redundant information.[3] Specifically, pieces of information from multiple source cannot be combined within most filtering frameworks unless they are independent or have a known degree of correlation (i.e., known cross covariances). In the battle management example described above, the effect of redundant information can be seen in the following scenario, sometimes referred to as the "whispering in the hall" problem:

1. The photoreconnaissance node transmits information about potentially important features. This information then propagates through the network, changing form as it is combined with information at other nodes in the process.
2. The troop position estimation node eventually receives the information in some form and notes that one of the indicated features could possibly represent a mobilizing tank battalion at position x. There are many other possible interpretations of the feature, but the possibility of a mobilizing tank battalion is deemed to be of such tactical importance that it warrants the transmission of a low confidence hypothesis (a "heads up" message). Again, the information can be synopsized, augmented, or otherwise transformed as it is relayed through a sequence of nodes.
3. The photoreconnaissance photo node receives the low confidence hypothesis that a tank battalion may have mobilized at position x. A check of available reconnaissance photos covering position x reveals a feature that is consistent with the hypothesis. Because the node is unaware that the hypothesis was based on that same photographic evidence, it assumes that the feature that it observes is an independent confirmation of the hypothesis. The node then transmits high confidence information that a feature at position x represents a mobilizing tank battalion.
4. The troop position node receives information from the photoreconnaissance node that a mobilizing tank battalion has been identified with high confidence. The troop position node regards this as confirmation of its early hypothesis and calls for an aggressive response to the mobilization. The obvious problem is that the two nodes are exchanging redundant pieces of information but are treating them as independent pieces of evidence mounting in support of the hypothesis that a tank battalion has mobilized. The end result is that critical resources may be diverted in reaction to what is, in fact, a low probability hypothesis.

A similar situation can arise in a decentralized monitoring system for a chemical process:

1. A reaction vessel is fitted with a variety of sensors, including a pressure gauge.
2. Because the bulk temperature of the reaction cannot be measured directly, a node is added that uses pressure information, combined with a model for the reaction, to estimate temperature.
3. A new node is added to the system that uses information from the pressure and temperature nodes.

Clearly, the added node will always be using redundant information from the pressure gauge. If the estimates of pressure and temperature are treated as independent, then the fact that their relationship is always exactly what is predicted by the model might lead to over confidence in the stability of the system. This type of inadvertent use of redundant information arises commonly when attempts are made to decompose systems into functional modules. The following example is typical:

1. A vehicle navigation and control system maintains one Kalman filter for estimating position and a separate Kalman filter for maintaining the orientation of the vehicle.
2. Each filter uses the same sensor information.
3. The full vehicle state is determined (for prediction purposes) by combining the position and orientation estimates.
4. The predicted position covariance is computed essentially as a sum of the position and orientation covariances (after the estimates are transformed to a common vehicle coordinate frame).

The problem in this example is that the position and orientation errors are not independent. This means that the predicted position covariance will underestimate the actual position error. Obviously, such overly confident position estimates can lead to unsafe maneuvers.

To avoid the potentially disastrous consequences of redundant data on Kalman-type estimators, covariance information must be maintained. Unfortunately, maintaining consistent cross covariances in arbitrary decentralized networks is not possible.[1] In only a few special cases, such as tree and fully connected networks, can the proliferation of redundant information be avoided. These special topologies, however, fail to provide the reliability advantage because the failure of a single node or link results in either a disconnected network or one that is no longer able to avoid the effects of redundant information. Intuitively, the redundancy of information in a network is what provides reliability; therefore, if the difficulties with redundant information are avoided by eliminating redundancy, then reliability will be also be eliminated.

The proof that cross covariance information cannot be consistently maintained in general decentralized networks seems to imply that the purported benefits of decentralization are unattainable. However, the proof relies critically on the assumption that some knowledge of the degree of correlation is necessary in order to fuse pieces of information. This is certainly the case for all classical data fusion mechanisms (e.g., the Kalman filter and Bayesian nets), which are based on applications of Bayes' rule. Furthermore, independence assumptions are also implicit in many ad hoc schemes that compute averages over quantities with intrinsically correlated error components.*

The problems associated with assumed independence are often side stepped by artificially increasing the covariance of the combined estimate. This heuristic (or filter "tuning") can prevent the filtering process from producing nonconservative estimates, but substantial empirical analysis and "tweaking" is required to determine how much to increase the covariances. Even with this empirical analysis, the integrity of the Kalman filter framework is compromised, and reliable results cannot be guaranteed. In many applications, such as in large decentralized signal/data fusion networks, the problem is much more acute and no amount of heuristic tweaking can avoid the limitations of the Kalman filter framework.[2] This is of enormous consequence, considering the general trend toward decentralization in complex military and industrial systems.

In summary, the only plausible way to simultaneously achieve robustness, exibility, and consistency in a general decentralized network is to exploit a data fusion mechanism that does not require independence assumptions. Such a mechanism, called Covariance Intersection (CI), satisfies this requirement.

*Dubious independence assumptions have permeated the literature over the decades and are now almost taken for granted. The fact is that statistical independence is an extremely rare property. Moreover, concluding that an approach will yield good approximations when "almost independent" is replaced with "assumed independent" in its analysis is usually erroneous.

12.3 Covariance Intersection

12.3.1 Problem Statement

Consider the following problem. Two pieces of information, labeled A and B, are to be fused together to yield an output, C. This is a very general type of data fusion problem. A and B could be two different sensor measurements (e.g., a batch estimation or track initialization problem), or A could be a prediction from a system model, and B could be sensor information (e.g., a recursive estimator similar to a Kalman filter). Both terms are corrupted by measurement noises and modeling errors, therefore, their values are known imprecisely and A and B are the random variables $\mathbf{a}$ and $\mathbf{b}$, respectively. Assume that the true statistics of these variables are unknown. The only available information are estimates of the means and covariances of $\mathbf{a}$ and $\mathbf{b}$ and the cross-correlations between them. These are $\{\mathbf{a}, \mathbf{P}_{aa}\}$, $\{\mathbf{b}, \mathbf{P}_{bb}\}$, and 0, respectively.*

$$\overline{\mathbf{P}}_{aa} = E\left[\tilde{\mathbf{a}}\tilde{\mathbf{a}}^T\right] \quad \overline{\mathbf{P}}_{ab} = E\left[\tilde{\mathbf{a}}\tilde{\mathbf{b}}^T\right] \quad \overline{\mathbf{P}}_{bb} = E\left[\tilde{\mathbf{b}}\tilde{\mathbf{b}}^T\right] \tag{12.1}$$

where $\tilde{\mathbf{a}} \triangleq \mathbf{a} - \overline{\mathbf{a}}$ and $\tilde{\mathbf{b}} \triangleq \mathbf{b} - \overline{\mathbf{b}}$ are the true errors imposed by assuming that the means are $\mathbf{a}$ and $\mathbf{b}$. Note that the cross-correlation matrix between the random variables, $\overline{\mathbf{P}}_{ab}$, is unknown and will not, in general, be 0.

The only constraint that we impose on the assumed estimate is consistency. In other words,

$$\begin{aligned} \mathbf{P}_{aa} - \overline{\mathbf{P}}_{aa} &\geq 0, \\ \mathbf{P}_{bb} - \overline{\mathbf{P}}_{bb} &\geq 0. \end{aligned} \tag{12.2}$$

This definition conforms to the standard definition of consistency.[4] The problem is to fuse the consistent estimates of A and B together to yield a new estimate C, $\{\mathbf{c}, \mathbf{P}_{cc}\}$, which is guaranteed to be consistent:

$$\mathbf{P}_{cc} - \overline{\mathbf{P}}_{cc} \geq 0 \tag{12.3}$$

where $\tilde{\mathbf{c}} \triangleq \mathbf{c} - \overline{\mathbf{c}}$ and $\overline{\mathbf{P}}_{cc} = \mathrm{E}\left[\tilde{\mathbf{c}}\tilde{\mathbf{c}}^T\right]$.

12.3.2 The Covariance Intersection Algorithm

In its generic form, the CI algorithm takes a convex combination of mean and covariance estimates that are represented information (inverse covariance) space. The intuition behind this approach arises from a *geometric* interpretation of the Kalman filter equations. The general form of the Kalman filter equation can be written as

$$\overline{\mathbf{c}} = \mathbf{W}_a\overline{\mathbf{a}} + \mathbf{W}_b\overline{\mathbf{b}} \tag{12.4}$$

$$\begin{aligned} \mathbf{P}_{cc} = {} & \mathbf{W}_a\mathbf{P}_{aa}\mathbf{W}_a^T + \mathbf{W}_a\mathbf{P}_{ab}\mathbf{W}_b^T \\ & + \mathbf{W}_b\mathbf{P}_{ba}\mathbf{W}_a^T + \mathbf{W}_b\mathbf{P}_{bb}\mathbf{W}_b^T \end{aligned} \tag{12.5}$$

where the weights $\mathbf{W}_a$ and $\mathbf{W}_b$ are chosen to minimize the trace of $\mathbf{P}_{cc}$. This form reduces to the conventional Kalman filter if the estimates are independent ($\mathbf{P}_{ab} = \mathbf{0}$) and generalizes to the Kalman filter with colored noise when the correlations are known.

*Cross correlation can also be treated as a nonzero value. For brevity, we do not discuss this case here.

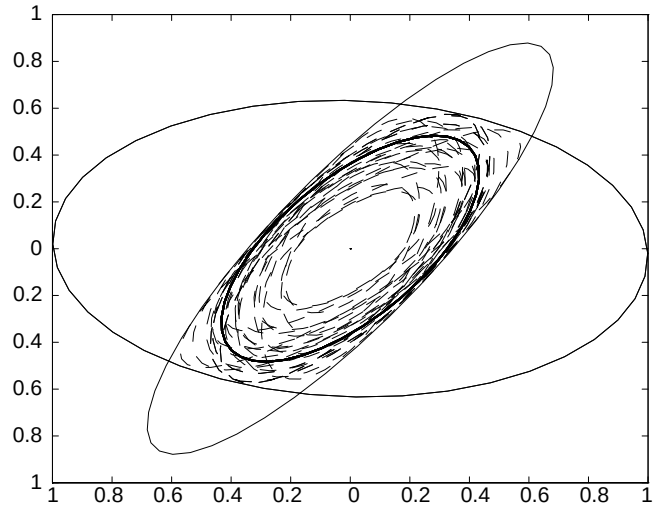


FIGURE 12.2 The shape of the updated covariance ellipse. The variances of $\mathbf{P}_{aa}$ and $\mathbf{P}_{bb}$ are the outer solid ellipses. Different values of $\mathbf{P}_{cc}$ that arise from different choices of $\mathbf{P}_{ab}$ are shown as dashed ellipses. The update with truly independent estimates is the inner solid ellipse.

These equations have a powerful geometric interpretation: If one plots the covariance ellipses (for a covariance matrix $\mathbf{P}$ this is the locus of points $\{\mathbf{p} : \mathbf{p}^T \mathbf{P}^{-1} \mathbf{p} = c\}$ where c is a constant), $\mathbf{P}_{aa}$, $\mathbf{P}_{bb}$, and $\mathbf{P}_{cc}$ for all choices of $\mathbf{P}_{ab}$, $\mathbf{P}_{cc}$ always lies within the *intersection* of $\mathbf{P}_{aa}$ and $\mathbf{P}_{bb}$. Figure 12.2 illustrates this for a number of different choices of $\mathbf{P}_{ab}$.

This interpretation suggests the following approach: if $\mathbf{P}_{cc}$ lies within the intersection of $\mathbf{P}_{aa}$ and $\mathbf{P}_{bb}$ for any possible choice of $\mathbf{P}_{ab}$, then an update strategy that finds a $\mathbf{P}_{cc}$ which encloses the intersection region must be consistent even if there is no knowledge about $\mathbf{P}_{ab}$. The tighter the updated covariance encloses the intersection region, the more effectively the update uses the available information.*

The intersection is characterized by the convex combination of the covariances, and the Covariance Intersection algorithm is:[5]

$$\mathbf{P}_{cc}^{-1} = \omega \mathbf{P}_{aa}^{-1} + (1-\omega)\mathbf{P}_{bb}^{-1} \tag{12.6}$$

$$\mathbf{P}_{cc}^{-1}\mathbf{c} = \omega \mathbf{P}_{aa}^{-1}\mathbf{a} + (1-\omega)\mathbf{P}_{bb}^{-1}\mathbf{b} \tag{12.7}$$

where $\omega \in [0, 1]$. Appendix 12.A proves that this update equation is consistent in the sense given by Equation 12.3 for all choices of $\mathbf{P}_{ab}$ and ω.

As illustrated in Figure 12.3, the free parameter ω manipulates the weights assigned to $\mathbf{a}$ and $\mathbf{b}$. Different choices of ω can be used to optimize the update with respect to different performance criteria, such as minimizing the trace or the determinant of $\mathbf{P}_{cc}$. Cost functions, which are convex with respect to ω, have only one distinct optimum in the range $0 \le \omega \le 1$. Virtually any optimization strategy can be used, ranging from Newton-Raphson to sophisticated semidefinite and convex programming[6] techniques, which can minimize almost any norm. Appendix 12.B includes source code for optimizing ω for the fusion of two estimates.

Note that some measure of covariance size must be minimized at each update in order to guarantee nondivergence; otherwise an updated estimate could be larger than the prior estimate. For example, if

*Note that the discussion of "intersection regions" and the plotting of particular covariance contours should not be interpreted in a way that confuses CI with ellipsoidal bounded region filters. CI does not exploit error bounds, only covariance information.

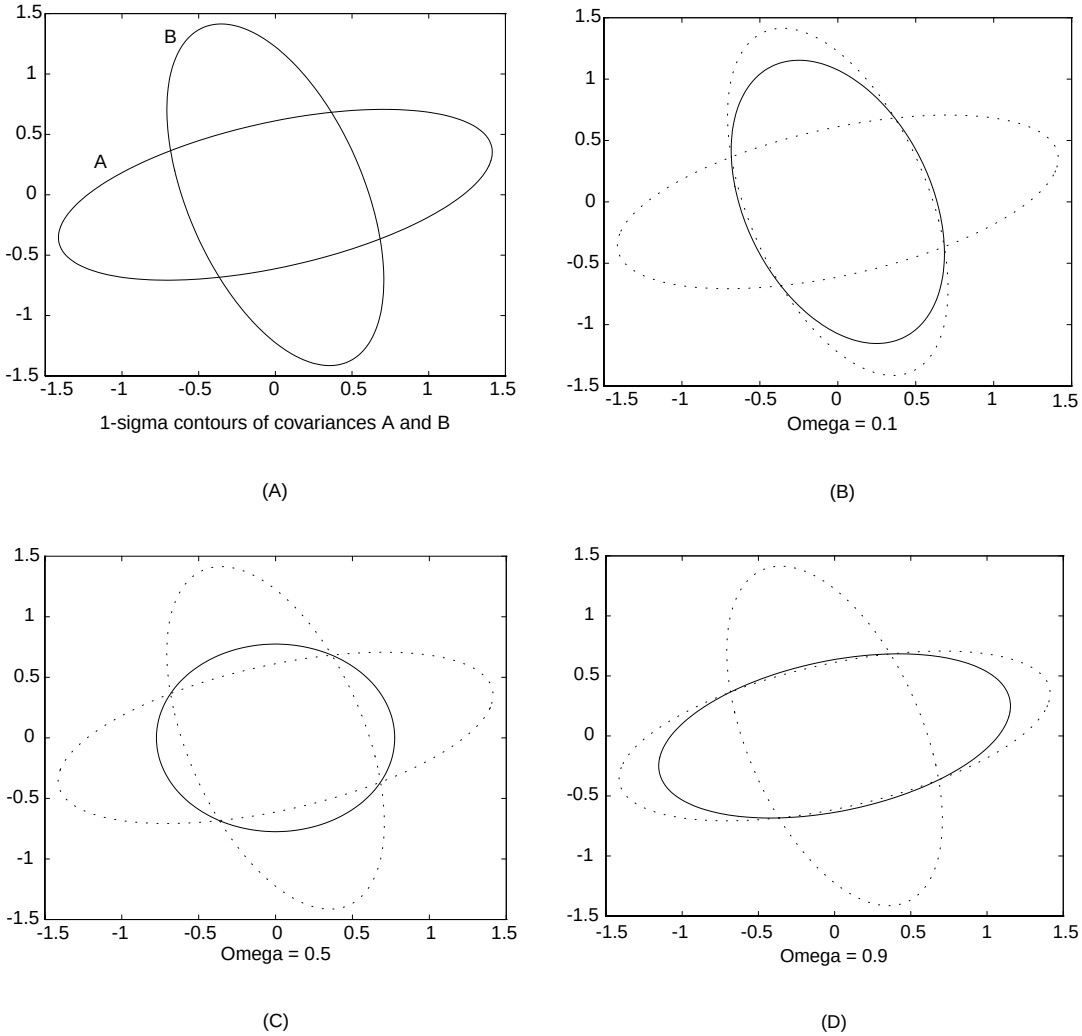


FIGURE 12.3 The value of ω determines the relative weights applied to each information term. (A) Shows the 1-sigma contours for 2-D covariance matrices A and B. (B)–(D) show the updated covariance C (drawn in a solid line) for several different values of ω. For each value of ω, C passes through the intersection points of A and B.

one were always to use $\omega = 0.5$, then the updated estimate would simply be the Kalman updated estimate with the covariance inflated by a factor of two. Thus, an update with an observation that has a very large covariance could result in an updated covariance close to twice the size of the prior estimate. In summary, the use of a fixed measure of covariance size with the CI equations leads to the nondivergent CI filter.

An example of the tightness of the CI update can be seen in Figure 12.4 for the case when the two prior covariances approach singularity:

$$\{\mathbf{a}, \mathbf{A}\} = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1.5 & 0.0 \\ 0.0 & \varepsilon \end{bmatrix} \right\} \tag{12.8}$$

$$\{\mathbf{b}, \mathbf{B}\} = \left\{ \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} \varepsilon & 0.0 \\ 0.0 & 1.0 \end{bmatrix} \right\} \tag{12.9}$$

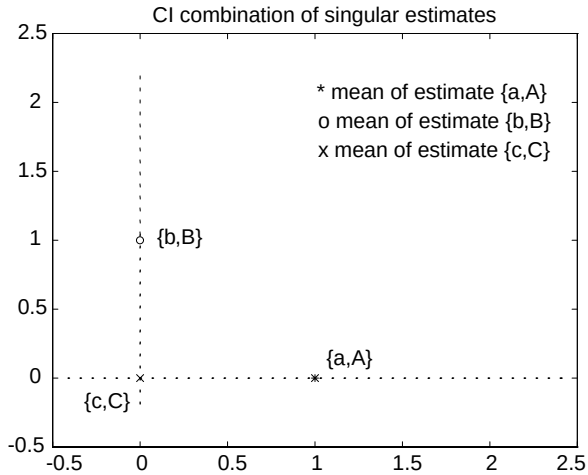


FIGURE 12.4 The CI update {c,C} of two 2-D estimates {a,A} and {b,B}, where A and B are singular, defines the point of intersection of the colinear sigma contours of A and B.

The covariance of the combined estimate is proportional to ε, and the mean is centered on the intersection point of the one-dimensional contours of the prior estimates. This makes sense intuitively because, if one estimate completely constrains one coordinate, and the other estimate completely constrains the other coordinate, there is only one possible update that can be consistent with both constraints.

CI can be generalized to an arbitrary number of $n > 2$ updates using the following equations:

$$\mathbf{P}_{cc}^{-1} = \omega_1 \mathbf{P}_{a_1 a_1}^{-1} + \cdots + \omega_n \mathbf{P}_{a_n a_n}^{-1} \tag{12.10}$$

$$\mathbf{P}_{cc}^{-1}\mathbf{c} = \omega_1 \mathbf{P}_{a_1 a_1}^{-1}\mathbf{a}_1 + \cdots + \omega_n \mathbf{P}_{a_n a_n}^{-1}\mathbf{a}_n \tag{12.11}$$

where $\Sigma_{i=1}^{n}\ \omega_i = 1$. For this type of batch combination of large numbers of estimates, efficient codes, such as the public domain MAXDET[7] and SPDSOL[8] are available.

In summary, CI provides a general update algorithm that is capable of yielding an updated estimate even when the prediction and observation correlations are unknown.

12.4 Using Covariance Intersection for Distributed Data Fusion

Consider again the data fusion network that is illustrated in Figure 12.1. The network consists of N nodes whose connection topology is completely arbitrary (i.e., it might include loops and cycles) and can change dynamically. Each node has information only about its local connection topology (e.g., the number of nodes with which it directly communicates and the type of data sent across each communication link). Assuming that the process and observation noises are independent, the only source of unmodeled correlations is the distributed data fusion system itself. CI can be used to develop a distributed data fusion algorithm which directly exploits this structure. The basic idea is illustrated in Figure 12.5. Estimates that are propagated from other nodes are correlated to an unknown degree and must be fused with the state estimate using CI. Measurements taken locally are known to be independent and can be fused using the Kalman filter equations.

Using conventional notation,[9] the estimate at the ith node is $\hat{\mathbf{x}}_i(k|k)$ with covariance $\mathbf{P}_i(k|k)$. CI can be used to fuse the information that is propagated between the different nodes. Suppose that, at time step $k + 1$, node i locally measures the observation vector $\mathbf{z}_i(k|k)$. A distributed fusion algorithm for propagating the estimate from timestep k to timestep $k + 1$ for node i is:

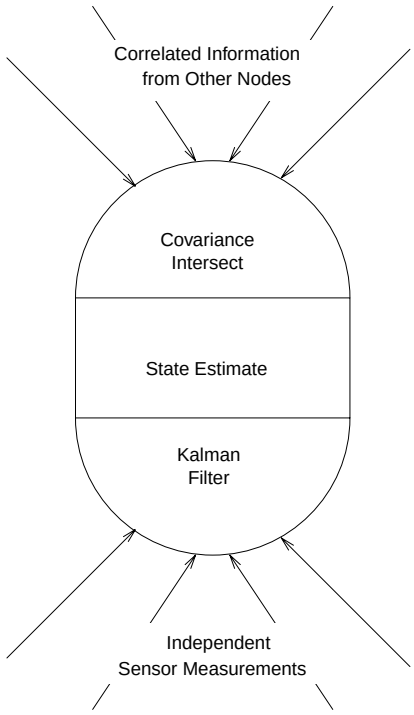


FIGURE 12.5 A canonical node in a general data fusion network that constructs its local state estimate using CI to combine information received from other nodes and a Kalman filter to incorporate independent sensor measurements.

1. Predict the state of node i at time $k + 1$ using the standard Kalman filter prediction equations.
2. Use the Kalman filter update equations to update the prediction with $\mathbf{z}_i(k + 1)$. This update is the distributed estimate with mean $\hat{\mathbf{x}}_i^*(k + 1|k + 1)$ and covariance $\mathbf{P}_i^*(k + 1|k + 1)$. It is not the final estimate, because it does not include observations and estimates propagated from the other nodes in the network.
3. Node i propagates its distributed estimate to all of its neighbors.
4. Node i fuses its prediction $\hat{\mathbf{x}}_i(k + 1|k)$ and $\mathbf{P}_i(k + 1|k)$ with the distributed estimates that it has received from all of its neighbors to yield the partial update with mean $\hat{\mathbf{x}}_i^+(k + 1|k + 1)$ and covariance $\mathbf{P}_i^+(k + 1|k + 1)$. Because these estimates are propagated from other nodes whose correlations are unknown, the CI algorithm is used. As explained above, if the node receives multiple estimates for the same time step, the batch form of CI is most efficient. Finally, node i uses the Kalman filter update equations to fuse $\mathbf{z}_i(k + 1)$ with its partial update to yield the new estimate $\hat{\mathbf{x}}_i(k + 1|k + 1)$ with covariance $\mathbf{P}_i(k + 1|k + 1)$. The node incorporates its observation last using the Kalman filter equations because it is known to be independent of the prediction or data which has been distributed to the node from its neighbors. Therefore, CI is unnecessary. This concept is illustrated in Figure 12.5.

An implementation of this algorithm is given in the next section. This algorithm has a number of important advantages. First, all nodes propagate their most accurate partial estimates to all other nodes without imposing any unrealistic requirements for perfectly robust communication. Communication paths may be uni- or bidirectional, there may be cycles in the network, and some estimates may be lost while others are propagated redundantly. Second, the update rates of the different filters do not need to be synchronized. Third, communications do not have to be guaranteed — a node can broadcast an estimate without relying on other nodes' receiving it. Finally, each node can use a different observation model: one node may have a high accuracy model for one subset of variables of relevance to it, and

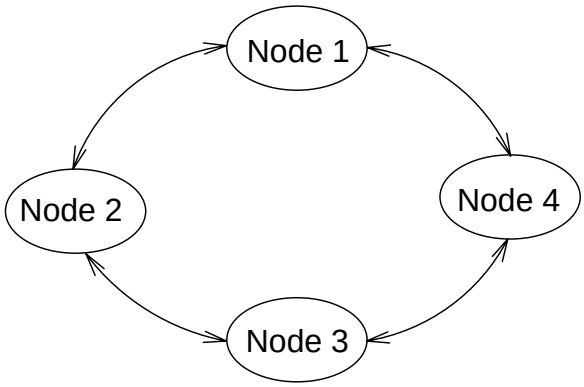


FIGURE 12.6 The network layout for the example.

another node may have a high accuracy model for a different subset of variables, but the propagation of their respective estimates allows nodes to construct fused estimates representing the union of the high accuracy information from both nodes.

The most important feature of the above approach to decentralized data fusion is that it is provably guaranteed to produce and maintain consistent estimates at the various nodes.* Section 5 demonstrates this consistency in a simple example.

12.5 Extended Example

Suppose the processing network, shown in Figure 12.6, is used to track the position, velocity and acceleration of a one-dimensional particle. The network is composed of four nodes. Node 1 measures the position of the particle only. Nodes 2 and 4 measure velocity and node 3 measures acceleration. The four nodes are arranged in a ring. From a practical standpoint, this configuration leads to a robust system with built-in redundancy: data can flow from one node to another through two different pathways. However, from a theoretical point of view, this configuration is extremely challenging. Because this configuration is neither fully connected nor tree-connected, optimal data fusion algorithms exist only in the special case where full knowledge of the network topology and the states at each node is known.

The particle moves using a nominal constant acceleration model with process noise injected into the jerk (derivative of acceleration). Assuming that the noise is sampled at the start of the timestep and is held constant throughout the prediction step, the process model is

$$X_{(k+1)} = \mathbf{F}x_{(k)} + \mathbf{G}v_{(K+1)} \tag{12.12}$$

where

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta T & \Delta T^2/2 \\ 0 & 1 & \Delta T \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{G} = \begin{bmatrix} \Delta T^3/6 \\ \Delta T^2/2 \\ \Delta T \end{bmatrix}$$

*The fundamental feature of CI can be described as consistent estimates in, consistent estimates out. The Kalman filter, in contrast, can produce an inconsistent fused estimate from two consistent estimates if the assumption of independence is violated. The only way CI can yield an inconsistent estimate is if a sensor or model introduces an inconsistent estimate into the fusion process. In practice this means that some sort of fault-detection mechanism needs to be associated with potentially faulty sensors.

$\upsilon(k)$ is an uncorrelated, zero-mean Gaussian noise with variance $\sigma_\upsilon^2 = 10$ and the length of the time step $\Delta T = 0.1s$.

The sensor information and the accuracy of each sensor is given in Table 12.1.

Assume, for the sake of simplicity, that the structure of the state space and the process models are the same for each node and the same as the true system. However, this condition is not particularly restrictive and many of the techniques of model and system distribution that are used in optimal data distribution networks can be applied with CI.[10]

TABLE 12.1 Sensor Information and Accuracy for Each Node from Figure 12.6

Node	Measures	Variance
1	x	1
2	$\dot{x}$	2
3	$\ddot{x}$	0.25
4	$\dot{x}$	3

The state at each node is predicted using the process model:

$$\hat{\mathbf{x}}_i(k+1|k) = \mathbf{F}\hat{\mathbf{x}}_i(k|k)$$

$$\mathbf{P}_i(k+1|k) = \mathbf{F}\mathbf{P}_i(k+1|k)\mathbf{F}^T + \mathbf{Q}(k)$$

The partial estimates $\hat{\mathbf{x}}_i^*(k+1|k+1)$ and $\mathbf{P}_i^*(k+1|k+1)$ are calculated using the Kalman filter update equations. If R_i is the observation noise covariance on the ith sensor, and H_i is the observation matrix, then the partial estimates are

$$v_i(k+1) = \mathbf{z}_i(k+1) - \mathbf{H}_i\hat{\mathbf{x}}_i(k+1|k) \tag{12.13}$$

$$\mathbf{S}_i(k+1) = \mathbf{H}_i\mathbf{P}_i(k+1|k)\mathbf{H}_i^T + \mathbf{R}_i(k+1) \tag{12.14}$$

$$\mathbf{W}_i(k+1) = \mathbf{P}_i(k+1|k)\mathbf{H}_i^T\mathbf{S}_i^{-1}(k+1) \tag{12.15}$$

$$\hat{\mathbf{x}}_i^*(k+1|k+1) = \hat{\mathbf{x}}_i(k+1|k) + \mathbf{W}_i(k+1)v_i(k+1) \tag{12.16}$$

$$\mathbf{P}_i^*(k+1|k+1) = \mathbf{P}_i(k+1|k) - \mathbf{W}_i(k+1)\mathbf{S}_i(k+1)\mathbf{W}_i^T(k+1) \tag{12.17}$$

Examine three strategies for combining the information from the other nodes:

1. The nodes are disconnected. No information flows between the nodes and the final updates are given by

$$\hat{\mathbf{x}}_i(k+1|k+1) = \hat{\mathbf{x}}_i^*(k+1|k+1) \tag{12.18}$$

$$\mathbf{P}_i(k+1|k+1) = \mathbf{P}_i^*(k+1|k+1) \tag{12.19}$$

2. Assumed independence update. All nodes are assumed to operate independently of one another. Under this assumption, the Kalman filter update equations can be used in Step 4 of the fusion strategy described in the last section.
3. CI-based update. The update scheme described in Section 12.4 is used.

The performance of each of these strategies was assessed using a Monte Carlo of 100 runs.

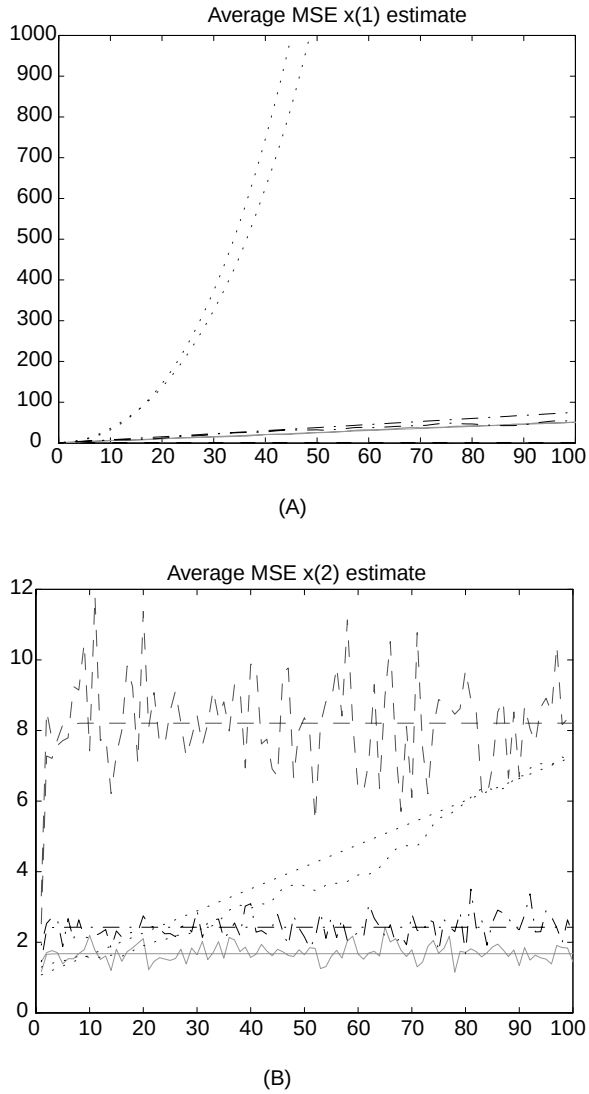
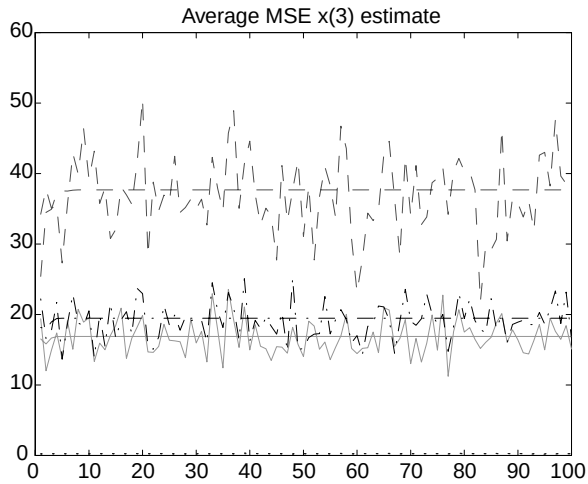


FIGURE 12.7 Disconnected nodes. (A) Mean squared error in x. (B) Mean squared error in $\dot{x}$. (C) Mean squared error in $\ddot{x}$. Mean squared errors and estimated covariances for all states in each of the four nodes. The curves for Node 1 are solid, Node 2 are dashed, Node 3 are dotted, and Node 4 are dash-dotted. The mean squared error is the rougher of the two lines for each node.

The results from the first strategy (no data distribution) are shown in Figure 12.7. As expected, the system behaves poorly. Because each node operates in isolation, only Node 1 (which measures x) is fully observable. The position variance increases without bound for the three remaining nodes. Similarly, the velocity is observable for Nodes 1, 2, and 4, but it is not observable for Node 3.

The results of the second strategy (all nodes are assumed independent) are shown in Figure 12.8. The effect of assumed independence observations is obvious: all of the estimates for all of the states in all of the nodes (apart from x for Node 3) are inconsistent. This clearly illustrates the problem of double counting.

Finally, the results from the CI distribution scheme are shown in Figure 12.9. Unlike the other two approaches, all the nodes are consistent and observable. Furthermore, as the results in Table 12.2 indicate, the steady-state covariances of all of the states in all of the nodes are smaller than those for case 1. In other words, this example shows that this data distribution scheme successfully and usefully propagates data through an apparently degenerate data network.



(C)

FIGURE 12.7 (continued).

This simple example is intended only to demonstrate the effects of redundancy in a general data distribution network. CI is not limited in its applicability to linear, time invariant systems. Furthermore, the statistics of the noise sources do not have to be unbiased and Gaussian. Rather, they only need to obey the consistency assumptions. Extensive experiments have shown that CI can be used with large numbers of platforms with nonlinear dynamics, nonlinear sensor models, and continuously changing network topologies (i.e., dynamic communications links).[11]

12.6 Incorporating Known Independent Information

CI and the Kalman filter are diametrically opposite in their treatment of covariance information: CI conservatively assumes that no estimate provides statistically independent information, and the Kalman filter assumes that every estimate provides statistically independent information. However, neither of these two extremes is representative of typical data fusion applications. This section demonstrates how the CI framework can be extended to subsume the generic CI filter and the Kalman filter and provide a completely general and optimal solution to the problem of maintaining and fusing consistent mean and covariance estimates.[22]

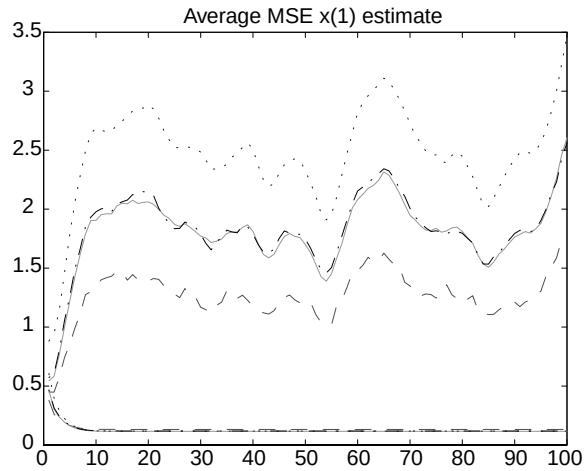
The following equation provides a useful interpretation of the original CI result. Specifically, the estimates $\{\mathbf{a}, \mathbf{A}\}$ and $\{\mathbf{b}, \mathbf{B}\}$ are represented in terms of their joint covariance:

$$\left\{\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}, \begin{bmatrix}\mathbf{A} & \mathbf{P}_{ab}\\ \mathbf{P}_{ab}^{T} & \mathbf{B}\end{bmatrix}\right\} \tag{12.20}$$

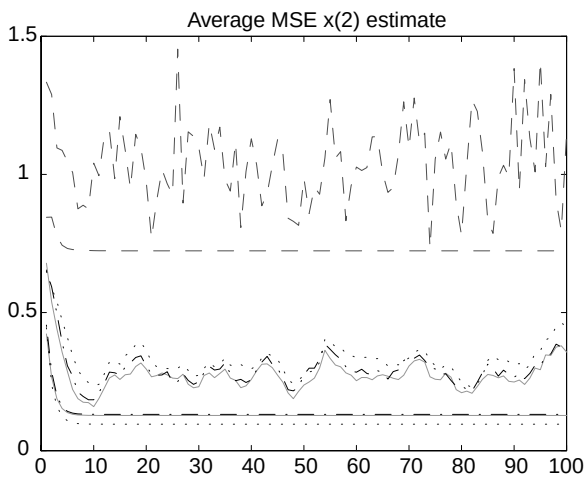
where in most situations the cross covariance, $\mathbf{P}_{ab}$, is unknown. The CI equations, however, support the conclusion that

$$\begin{bmatrix}\mathbf{A} & \mathbf{P}_{ab}\\ \mathbf{P}_{ab}^{T} & \mathbf{B}\end{bmatrix} \le \begin{bmatrix}\omega\mathbf{A}^{-1} & 0\\ 0 & (1-\omega)\mathbf{B}^{-1}\end{bmatrix}^{-1} \tag{12.21}$$

because CI must assume a joint covariance that is conservative with respect to the true joint covariance. Evaluating the inverse of the right-hand-side (RHS) of the equation leads to the following consistent/conservative estimate for the joint system:



(A)



(B)

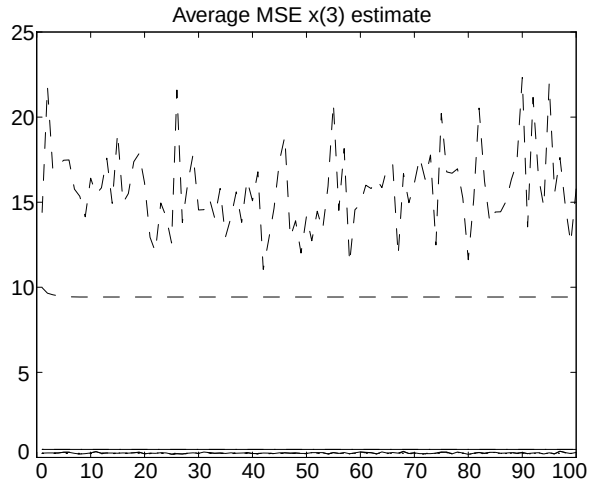
FIGURE 12.8 All nodes assumed independent. (A) Mean squared error in x. (B) Mean squared error in $\dot{x}$. (C) Mean squared error in $\ddot{x}$. Mean squared errors and estimated covariances for all states in each of the four nodes. The curves for Node 1 are solid, Node 2 are dashed, Node 3 are dotted, and Node 4 are dash-dotted. The mean squared error is the rougher of the two lines for each node.

$$\left\{\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}, \begin{bmatrix}\frac{1}{\omega}\mathbf{A} & 0\\ 0 & \frac{1}{1-\omega}\mathbf{B}\end{bmatrix}\right\} \tag{12.22}$$

From this result, the following generalization of CI can be derived:*

CI with Independent Error: Let $a = a_1 + a_2$ and $b = b_1 + b_2$, where a_1 and b_1 are correlated to an unknown degree, while the errors associated with a_2 and b_2 are completely independent of all others.

*In the process, a consistent estimate of the covariance of $\mathbf{a} + \mathbf{b}$ is also obtained, where a and b have an unknown degree of correlation, as $\frac{1}{\omega}\mathbf{A} + \frac{1}{1-\omega}\mathbf{B}$. We refer to this operation as *covariance addition* (CA).



(C)

FIGURE 12.8 (continued).

Also, let the respective covariances of the components be A_1, A_2, B_1, and B_2. From the above results, a consistent joint system can be formed as:

$$\left\{\begin{bmatrix}\mathbf{a}_1+\mathbf{a}_2\\ \mathbf{b}_1+\mathbf{b}_2\end{bmatrix}, \begin{bmatrix}\frac{1}{\omega}\mathbf{A}_1+\mathbf{A}_2 & 0\\ 0 & \frac{1}{1-\omega}\mathbf{B}_1+\mathbf{B}_2\end{bmatrix}\right\} \tag{12.23}$$

Letting $\mathbf{A}=\frac{1}{\omega}\mathbf{A}_1+\mathbf{A}_2$ and $\mathbf{B}=\frac{1}{1-\omega}\mathbf{B}_1+\mathbf{B}_2$, gives the following generalized CI equations:

$$\mathbf{C}=\left[\mathbf{A}^{-1}+\mathbf{B}^{-1}\right]^{-1}=\left[\left(\frac{1}{\omega}\mathbf{A}_1+\mathbf{A}_2\right)^{-1}+\left(\frac{1}{1-\omega}\mathbf{B}_1+\mathbf{B}_2\right)^{-1}\right]^{-1} \tag{12.24}$$

$$\mathbf{c}=\left[\mathbf{A}^{-1}\mathbf{a}+\mathbf{B}^{-1}\mathbf{b}\right]^{-1}=\mathbf{C}\left[\left(\frac{1}{\omega}\mathbf{A}_1+\mathbf{A}_2\right)^{-1}\mathbf{a}+\left(\frac{1}{1-\omega}\mathbf{B}_1+\mathbf{B}_2\right)^{-1}\mathbf{b}\right] \tag{12.25}$$

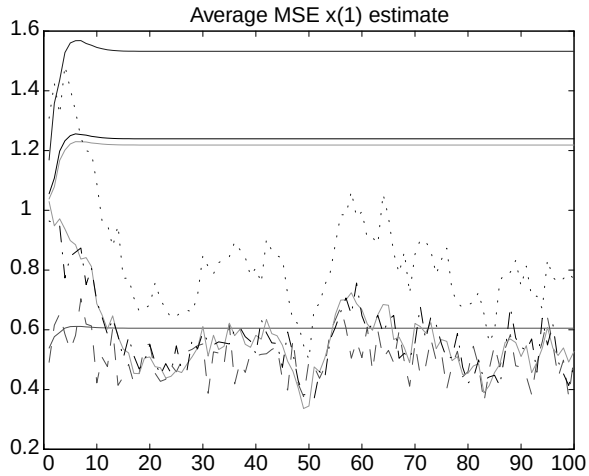
where the known independence of the errors associated with a_2 and b_2 is exploited.

Although the above generalization of CI exploits available knowledge about independent error components, further exploitation is impossible because the combined covariance C is formed from *both* independent and correlated error components. However, CI can be generalized even further to produce and maintain separate covariance components, C_1 and C_2, reflecting the correlated and known-independent error components, respectively. This generalization is referred to as Split CI.

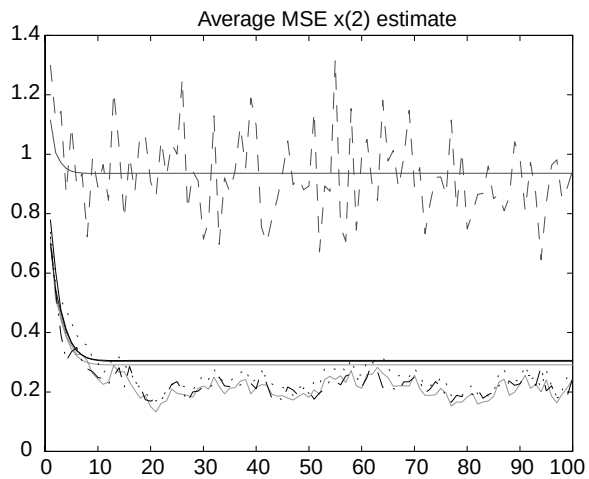
If we let $\tilde{\mathbf{a}}_1$ and $\tilde{\mathbf{a}}_2$ be the correlated and known-independent error components of a, with $\tilde{\mathbf{b}}_1$ and $\tilde{\mathbf{b}}_2$ similarly defined for b, then we can express the errors $\tilde{\mathbf{c}}_1$ and $\tilde{\mathbf{c}}_2$ in information (inverse covariance) form as

$$\mathbf{C}^{-1}\left(\tilde{\mathbf{c}}_1+\tilde{\mathbf{c}}_2\right)=\mathbf{A}^{-1}\left(\tilde{\mathbf{a}}_1+\tilde{\mathbf{a}}_2\right)+\mathbf{B}^{-1}\left(\tilde{\mathbf{b}}_1+\tilde{\mathbf{b}}_2\right) \tag{12.26}$$

from which the following can be obtained after premultiplying by C:



(A)



(B)

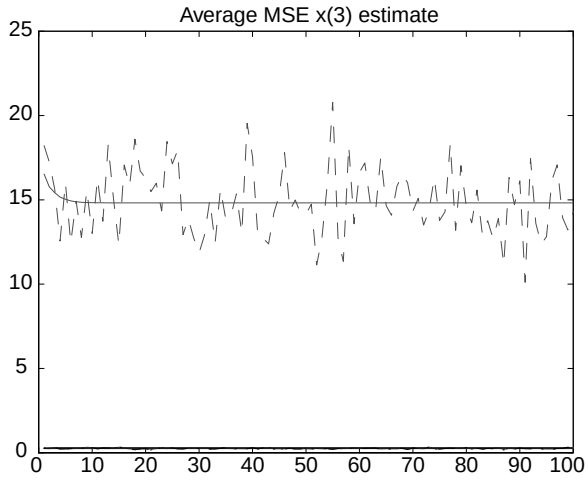
FIGURE 12.9 CI distribution scheme. (A) Mean squared error in x. (B) Mean squared error in $\dot{x}$. (C) Mean squared error in $\ddot{x}$. Mean squared errors and estimated covariances for all states in each of the four nodes. The curves for Node 1 are solid, Node 2 are dashed, Node 3 are dotted, and Node 4 are dash-dotted. The mean squared error is the rougher of the two lines for each node.

$$\left(\tilde{\mathbf{c}}_1 + \tilde{\mathbf{c}}_2\right) = \mathbf{C}\left[\mathbf{A}^{-1}\left(\tilde{\mathbf{a}}_1 + \tilde{\mathbf{a}}_2\right) + \mathbf{B}^{-1}\left(\tilde{\mathbf{b}}_1 + \tilde{\mathbf{b}}_2\right)\right] \tag{12.27}$$

Squaring both sides, taking expectations, and collecting independent terms* yields:

$$\mathbf{C}_2 = \left(\mathbf{A}^{-1} + \mathbf{B}^{-1}\right)^{-1}\left(\mathbf{A}^{-1}\mathbf{A}_2\mathbf{A}^{-1} + \mathbf{B}^{-1}\mathbf{B}_2\mathbf{B}^{-1}\right)\left(\mathbf{A}^{-1} + \mathbf{B}^{-1}\right)^{-1} \tag{12.28}$$

*Recall that $\mathbf{A} = \frac{1}{\omega}\mathbf{A}_1$ and $\mathbf{B} = \frac{1}{1-\omega}\mathbf{B}_1 + \mathbf{B}_2$.



(C)

FIGURE 12.9 (continued).

TABLE 12.2 The Diagonal Elements of the Covariance Matrices for Each Node at the End of 100 Timesteps for Each of the Consistent Distribution Schemes

Node	Scheme	σ_x^2	σ_x^2	σ_x^2
1	NONE	0.8823	8.2081	37.6911
	CI	0.6055	0.9359	14.823
2	NONE	50.5716*	1.6750	16.8829
	CI	1.2186	0.2914	0.2945
3	NONE	77852.3*	7.2649*	0.2476
	CI	1.5325	0.3033	0.2457
4	NONE	75.207	2.4248	19.473
	CI	1.2395	0.3063	0.2952

Note: NONE – no distribution, and CI – the CI algorithm). The asterisk denotes that a state is *unobservable* and its variance is increasing without bound.

where the nonindependent part can be obtained simply by subtracting the above result from the overall fused covariance $\mathbf{C} = (\mathbf{A}^{-1} + \mathbf{B}^{-1})^{-1}$. In other words,

$$\mathbf{C}_1 = \left(\mathbf{A}^{-1} + \mathbf{B}^{-1}\right)^{-1} - \mathbf{C}_2 \tag{12.29}$$

Split CI can also be expressed in batch form analogously to the batch form of original CI. Note that the covariance addition equation can be generalized analogously to provide Split CA capabilities.

The generalized and split variants of CI optimally exploit knowledge of statistical independence. This provides an extremely general filtering, control, and data fusion framework that completely subsumes the Kalman filter.

12.6.1 Example Revisited

The contribution of generalized CI can be demonstrated by revisiting the example described in Section 12.5. The scheme described earlier attempted to exploit information that is independent in the observations. However, it failed to exploit one potentially very valuable source of information — the fact that the distributed estimates ($\hat{\mathbf{x}}_i^*(k+1|k+1)$ with covariance $\mathbf{P}_i^*(k+1|k+1)$) contain the observations taken at time step $k+1$. Under the assumption that the measurement errors are uncorrelated, generalized CI can be exploited to significantly improve the performance of the information network. The distributed estimates are split into the (possibly) correlated and known independent components, and generalized CI can be used to fuse the data remotely.

The estimate of node i at time step k is maintained in split form with mean $\hat{\mathbf{x}}_i(k|k)$ and covariances $\mathbf{P}_{i,1}(k|k)$ and $\mathbf{P}_{i,2}(k|k)$. As explained below, it is not possible to ensure that $\mathbf{P}_{i,2}(k|k)$ will be independent of the distributed estimates that will be received at time step k. Therefore, the prediction step combines the correlated and independent terms into the correlated term, and sets the independent term to 0:

$$\begin{aligned}
\hat{\mathbf{x}}_i(k+1|k) &= \mathbf{F}\hat{\mathbf{x}}_i(k|k) \\
\mathbf{P}_{i,1}(k+1|k) &= \mathbf{F}\left(\mathbf{P}_{i,1}(k+1|k)+\mathbf{P}_{i,2}(k+1|k)\right)\mathbf{F}^T+\mathbf{Q}(k) \\
\mathbf{P}_{i,2}(k+1|k) &= 0
\end{aligned} \tag{12.30}$$

The process noise is treated as a correlated noise component because each sensing node is tracking the same object. Therefore, the process noise that acts on each node is perfectly correlated with the process noise acting on all other nodes.

The split form of the distributed estimate is found by applying split CI to fuse the prediction with $\mathbf{z}_i(k+1)$. Because the prediction contains only correlated terms, and the observation contains only independent terms ($\mathbf{A}_2 = \mathbf{0}$ and $\mathbf{B}_1 = \mathbf{0}$ in Equation 12.24) the optimized solution for this update occurs when $\omega = 1$. This is the same as calculating the normal Kalman filter update and explicitly partitioining the contributions of the predictions from the observations. Let $\mathbf{W}_i^*(k+1)$ be the weight used to calculate the distributed estimate. From Equation 12.30 its value is given by,

$$\mathbf{S}_i^*(k+1) = \mathbf{H}_i\mathbf{P}_{i,1}(k+1|k)\mathbf{H}_i^T + \mathbf{R}_i(k+1) \tag{12.31}$$

$$\mathbf{W}_i^*(k+1) = \mathbf{P}_{i,1}(k+1|k)\mathbf{H}_i^T\mathbf{S}_i^*(k+1)^{-1} \tag{12.32}$$

Note that the Covariance Addition equation can be generalized analogously to provide Split CA capabilities.

Taking outer products of the prediction and observation contribution terms, the correlated and independent terms of the distributed estimate are

$$\begin{aligned}
\mathbf{P}_{i,1}^*(k+1|k+1) &= \mathbf{X}(k)+1\mathbf{P}_i(k+1|k)\mathbf{X}^T(k+1) \\
\mathbf{P}_{i,2}^*(k+1|k+1) &= \mathbf{W}(k+1)+1\mathbf{R}(k)+1\mathbf{W}^T(k+1)
\end{aligned} \tag{12.33}$$

where $\mathbf{X}(k+1) = \mathbf{I} - \mathbf{W}_i^*(k+1)\,\mathbf{H}(k+1)$.

The split distributed updates are propagated to all other nodes where they are fused with split CI to yield a split partial estimate with mean $\hat{\mathbf{x}}_i^+(k+1|k+1)$ and covariances $\mathbf{P}_{i,1}^+(k+1|k+1)$ and $\mathbf{P}_{i,2}^+(k+1|k+1)$.

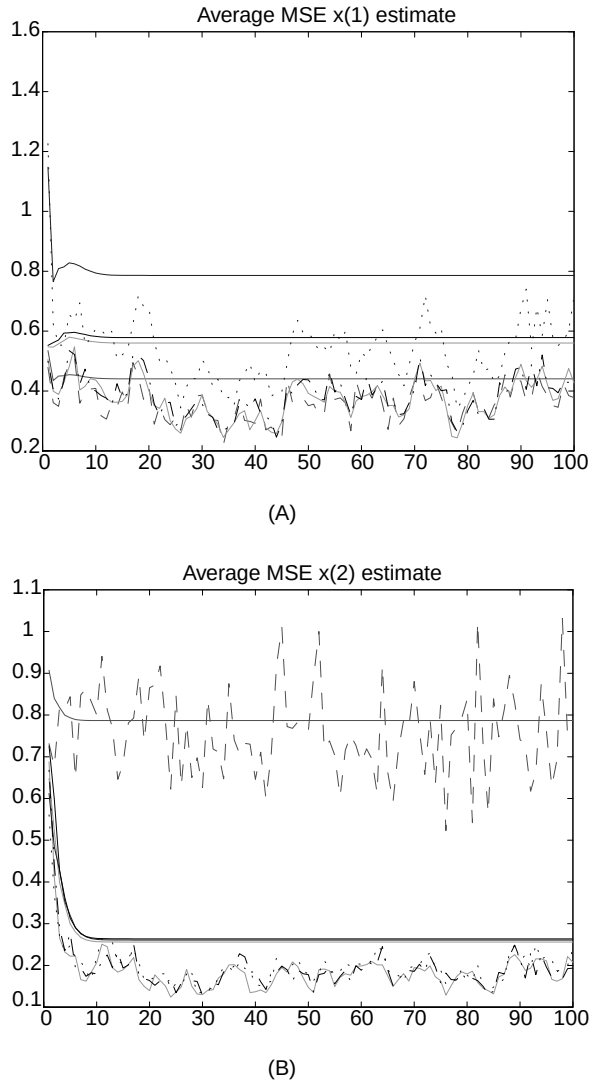


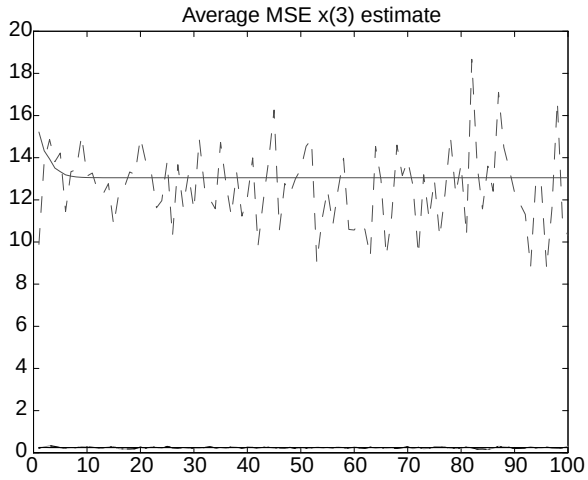
FIGURE 12.10 Mean squared errors and estimated covariances for all states in each of the four nodes. (A) Mean squared error in x. (B) Mean squared error in $\dot{x}$. (C) Mean squared error in $\ddot{x}$. The curves for Node 1 are solid, Node 2 are dashed, Node 3 are dotted, and Node 4 are dash-dotted. The mean squared error is the rougher of the two lines for each node.

Split CI can now be used to incorporate $\mathbf{z}(k)$. However, because the observation contains no correlated terms ($B_1 = 0$ in Equation 12.24), the optimal solution is always $\omega = 1$.

The effect of this algorithm can be seen in Figure 12.10 and in Table 12.3. As can be seen, the results of generalized CI are dramatic. The most strongly affected node is Node 2, whose position variance is reduced almost by a factor of 3. The least affected node is Node 1. This is not surprising, given that Node 1 is fully observable. Even so, the variance on its position estimate is reduced by more than 25%.

12.7 Conclusions

This chapter has considered the extremely important problem of data fusion in arbitrary data fusion networks. It described a general data fusion/update technique that makes no assumptions about the



(C)

FIGURE 12.10 (continued).

TABLE 12.3 The Diagonal Elements of the Covariance Matrices for Each Node at the End of 100 Timesteps for Each of the Consistent Distribution Schemes

Node	Scheme	σ_x^2	$\sigma_{\hat{x}}^2$	$\sigma_{\tilde{x}}^2$
1	NONE	0.8823	8.2081	37.6911
	CI	0.6055	0.9359	14.823
	GCI	0.4406	0.7874	13.050
2	NONE	50.5716*	1.6750	16.8829
	CI	1.2186	0.2914	0.2945
	GCI	0.3603	0.2559	0.2470
3	NONE	77852.3*	7.2649*	0.2476
	CI	1.5325	0.3033	0.2457
	GCI	0.7861	0.2608	0.2453
4	NONE	75.207	2.4248	19.473
	CI	1.2395	0.3063	0.2952
	GCI	0.5785	0.2636	0.2466

Note: NONE — no distribution; CI — the CI algorithm; GCI — generalized CI algorithm, which is described in Section 12.6. An asterisk denotes that a state is *unobservable* and its variance is increasing without bound. The covariance used for the GCI values is $P_i(k|k) = P_{i,1}(k|k) + P_{i,2}(k|k)$.

independence of the estimates to be combined. The use of the covariance intersection framework to combine mean and covariance estimates without information about their degree of correlation provides a direct solution to the distributed data fusion problem.

However, the problem of unmodeled correlations reaches far beyond distributed data fusion and touches the heart of most types of tracking and estimation. Other application domains for which CI is highly relevant include:

- *Multiple model filtering* — Many systems switch behaviors in a complicated manner, so that a comprehensive model is difficult to derive. If multiple approximate models are available that capture different behavioral aspects with different degrees of fidelity, their estimates can be combined to achieve a better estimate. Because they are all modeling the same system, however, the different estimates are likely to be highly correlated.[12,13]
- *Simultaneous map building and localization for autonomous vehicles* — When a vehicle estimates the positions of landmarks in its environment while using those same landmarks to update its own position estimate, the vehicle and landmark position estimates become highly correlated.[5,14]
- *Track-to-track data fusion in multiple-target tracking systems* — When sensor observations are made in a dense target environment, there is ambiguity concerning which tracked target produced each observation. If two tracks are determined to correspond to the same target, assuming independence may not be possible when combining them, if they are derived from common observation information.[11,12]
- *Nonlinear filtering* — When nonlinear transformations are applied to observation estimates, correlated errors arise in the observation sequence. The same is true for time propagations of the system estimate. Covariance intersection will ensure nondivergent nonlinear filtering if every covariance estimate is conservative. Nonlinear extensions of the Kalman filter are inherently flawed because they require independence regardless of whether the covariance estimates are conservative.[5,15-20]

Current approaches to these and many other problems attempt to circumvent troublesome correlations by heuristically adding "stabilizing noise" to updated estimates to ensure that they are conservative. The amount of noise is likely to be excessive in order to guarantee that no covariance components are underestimated. Covariance intersection ensures the best possible estimate, given the amount of information available. The most important fact that must be emphasized is that the procedure makes no assumptions about independence, nor the underlying distributions of the combined estimates. Consequently, covariance intersection likely will replace the Kalman filter in a wide variety of applications where independence assumptions are unrealistic.

Acknowledgments

The authors gratefully acknowledge the support of IDAK Industries for supporting the development of the full CI framework and the Office of Naval Research (Contract N000149WX20103) for supporting current experiments and applications of this framework. The authors also acknowledge support from RealityLab.com and the University of Oxford.

Appendix 12.A The Consistency of CI

This appendix proves that covariance intersection yields a consistent estimate for any value of ω and $\bar{\mathbf{P}}_{ab}$ providing that a and b are consistent.[21]

The CI algorithm calculates its mean using Equation 12.7. The actual error in this estimate is

$$\tilde{\mathbf{c}} = \mathbf{P}_{cc}\left\{\omega\mathbf{P}_{aa}^{-1}\tilde{\mathbf{a}} + \left(1-\omega\right)\mathbf{P}_{bb}^{-1}\tilde{\mathbf{b}}\right\} \tag{12.34}$$

By taking outer products and expectations, the actual mean squared error which is committed by using Equation 12.7 to calculate the mean is

$$\begin{aligned} E\left[\tilde{\mathbf{c}}\tilde{\mathbf{c}}^T\right] = \mathbf{P}_{cc}\Big\{&\omega^2\mathbf{P}_{aa}^{-1}\bar{\mathbf{P}}_{aa}\mathbf{P}_{aa}^{-1} + \omega\left(1-\omega\right)\mathbf{P}_{aa}^{-1}\bar{\mathbf{P}}_{ab}\mathbf{P}_{bb}^{-1} \\ &+ \omega\left(1-\omega\right)\mathbf{P}_{bb}^{-1}\bar{\mathbf{P}}_{ba}\mathbf{P}_{aa}^{-1} + \left(1-\omega\right)^2\mathbf{P}_{bb}^{-1}\bar{\mathbf{P}}_{bb}\mathbf{P}_{bb}^{-1}\Big\}\mathbf{P}_{cc} \end{aligned} \tag{12.35}$$

Because $\bar{\mathbf{P}}_{ab}$ is not known, the true value of the mean squared error cannot be calculated. However, CI implicitly calculates an upper bound of this quantity. If Equation 12.35 is substituted into Equation 12.3, the consistency condition can be written as

$$\begin{aligned}\mathbf{P}_{cc} - \mathbf{P}_{cc}\Big\{&\omega^2\mathbf{P}_{aa}^{-1}\bar{\mathbf{P}}_{aa}\mathbf{P}_{aa}^{-1} + \omega\left(1-\omega\right)\mathbf{P}_{aa}^{-1}\bar{\mathbf{P}}_{ab}\mathbf{P}_{bb}^{-1} \\ &+ \omega\left(1-\omega\right)\mathbf{P}_{bb}^{-1}\bar{\mathbf{P}}_{ba}\mathbf{P}_{aa}^{-1} + \left(1-\omega\right)^2\mathbf{P}_{bb}^{-1}\bar{\mathbf{P}}_{bb}\mathbf{P}_{bb}^{-1}\Big\}\mathbf{P}_{cc} \geq \mathbf{0}\end{aligned} \tag{12.36}$$

Pre- and postmultiplying both sides by $\mathbf{P}_{cc}^{-1}$ and collecting terms, gives

$$\begin{aligned}\mathbf{P}_{cc}^{-1} - \omega^2\mathbf{P}_{aa}^{-1}\bar{\mathbf{P}}_{aa}\mathbf{P}_{aa}^{-1} - \omega\left(1-\omega\right)&\mathbf{P}_{aa}^{-1}\bar{\mathbf{P}}_{ab}\mathbf{P}_{bb}^{-1} \\ - \omega\left(1-\omega\right)\mathbf{P}_{bb}^{-1}\mathbf{P}_{ba}\mathbf{P}_{aa}^{-1} - \left(1-\omega\right)^2&\mathbf{P}_{bb}^{-1}\bar{\mathbf{P}}_{bb}\mathbf{P}_{bb}^{-1}\Big\}\mathbf{P}_{cc} \geq \mathbf{0}\end{aligned} \tag{12.37}$$

An upper bound on $\mathbf{P}_{cc}^{-1}$, which can be found and expressed using $\mathbf{P}_{aa}$, $\mathbf{P}_{bb}$, $\bar{\mathbf{P}}_{aa}$, and $\bar{\mathbf{P}}_{bb}$. From the consistency condition for **a**,

$$\mathbf{P}_{aa} - \bar{\mathbf{P}}_{aa} \geq \mathbf{0} \tag{12.38}$$

or, by pre- and postmultiplying by $\mathbf{P}_{aa}^{-1}$,

$$\mathbf{P}_{aa}^{-1} \geq \mathbf{P}_{aa}^{-1}\bar{\mathbf{P}}_{aa}\mathbf{P}_{aa}^{-1} \tag{12.39}$$

A similar condition exists for b and, substituting these results in Equation 12.6,

$$\mathbf{P}_{cc}^{-1} = \omega\mathbf{P}_{aa}^{-1} + \left(1-\omega\right)\mathbf{P}_{bb}^{-1} \tag{12.40}$$

$$\geq \omega\mathbf{P}_{aa}^{-1}\bar{\mathbf{P}}_{aa}\mathbf{P}_{aa}^{-1} + \left(1-\omega\right)\mathbf{P}_{bb}^{-1}\bar{\mathbf{P}}_{bb}\mathbf{P}_{bb}^{-1} \tag{12.41}$$

Substituting this lower bound on $\mathbf{P}_{cc}^{-1}$ into Equation 12.37 leads to

$$\omega\left(1-\omega\right)\left(\mathbf{P}_{aa}^{-1}\bar{\mathbf{P}}_{aa}\mathbf{P}_{aa}^{-1} - \mathbf{P}_{aa}^{-1}\bar{\mathbf{P}}_{ab}\mathbf{P}_{bb}^{-1} - \mathbf{P}_{bb}^{-1}\bar{\mathbf{P}}_{ba}\mathbf{P}_{aa}^{-1}\bar{\mathbf{P}}_{bb}\mathbf{P}_{bb}^{-1}\right) \geq \mathbf{0} \tag{12.42}$$

or

$$\omega\left(1-\omega\right)\mathrm{E}\left[\left\{\mathbf{P}_{aa}^{-1}\tilde{\mathbf{a}} - \mathbf{P}_{bb}^{-1}\tilde{\mathbf{b}}\right\}\left\{\mathbf{P}_{aa}^{-1}\tilde{\mathbf{a}} - \mathbf{P}_{bb}^{-1}\tilde{\mathbf{b}}\right\}^T\right] \geq \mathbf{0} \tag{12.43}$$

Clearly, the inequality must hold for all choices of $\bar{\mathbf{P}}_{ab}$ and $\omega \in [0, 1]$.

Appendix 12.B MATLAB Source Code

This appendix provides source code for performing the CI update in MATLAB.

12.B.1 Conventional CI

```
function [c,C,omega]=CI(a,A,b,B,H)
%
% function [c,C,omega]=CI(a,A,b,B,H)
%
% This function implements the CI algorithm and fuses two estimates
% (a,A) and (b,B) together to give a new estimate (c,C) and the value
% of omega which minimizes the determinant of C.  The observation
% matrix is H.

Ai=inv(A);
Bi=inv(B);

% Work out omega using the matlab constrained minimiser function
% fminbnd().
f=inline('1/det(Ai*omega+H''*Bi*H*(1-omega))', ...
           'omega', 'Ai', 'Bi', 'H');
omega=fminbnd(f,0,1,optimset('Display','off'),Ai,Bi,H);

% The unconstrained version of this optimisation is:
% omega = fminsearch(f,0.5,optimset('Display','off'),Ai,Bi,H);
% omega = min(max(omega,0),1);

% New covariance
C=inv(Ai*omega+H'*Bi*H*(1-omega));

% New mean
nu=b-H*a;
W=(1-omega)*C*H'*Bi;
c=a+W*nu;
```

12.B.2 Split CI

```
function [c,C1,C2,omega] = SCI(a,A1,A2,b,B1,B2,H)
%
% function [c,C1,C2,omega] = SCI(a,A1,A2,b,B1,B2,H)
%
% This function implements the split CI algorithm and fuses two
% estimates (a,A1,A2) and (b,B1,B2) together to give a new estimate
% (c,C1,C2) and the value of omega which minimizes the determinant of
% (C1+C2). The observation matrix is H.
%

% Work out omega using the matlab constrained minimiser function
% fminbnd().

f=inline('1/det(omega*inv(A1+omega*A2)+(1-omega)*H''*inv(B1+(1-
omega)*B2)*H)', ...
           'omega', 'A1', 'A2', 'B1', 'B2', 'H');
omega = fminbnd(f,0,1,optimset('Display','off'),A1,A2,B1,B2,H);
```

```
% The unconstrained version of this optimisation is:
% omega = fminsearch(f,0.5,optimset('Display','off'),A1,A2,B1,B2,H);
% omega = min(max(omega,0),1);

Ai=omega*inv(A1+omega*A2);
HBi=(1-omega)*H'*inv(B1+(1-omega)*B2);

% New covariance
C=inv(Ai+HBi*H);
C2=C*(Ai*A2*Ai'+HBi*B2*HBi')*C;
C1=C-C2;

% New mean
nu=b-H*a;
W=C*HBi;
c=a+W*nu;
```

References

1. Utete, S.W., Network management in decentralised sensing systems, Ph.D. thesis, Robotics Research Group, Department of Engineering Science, University of Oxford, 1995.
2. Grime, S. and Durrant-Whyte H., Data fusion in decentralized sensor fusion networks, *Control Engineering Practice*, 2(5), 849, 1994.
3. Chong, C., Mori, S., and Chan, K., Distributed multitarget multisensor tracking, *Multitarget Multisensor Tracking*, Artech House Inc., Boston, 1990.
4. Jazwinski, A.H., *Stochastic Processes and Filtering Theory*, Academic Press, New York, 1970.
5. Uhlmann, J.K., Dynamic map building and localization for autonomous vehicles, Ph.D. thesis, University of Oxford, 1995/96.
6. Vandenberghe, L. and Boyd, S., Semidefinite programming, *SIAM Review*, March 1996.
7. Wu, S.P., Vandenberghe, L., and Boyd, S., Maxdet: Software for determinant maximization problems, alpha version, Stanford University, April 1996.
8. Boyd, S. and Wu, S.P., *SDPSOL: User's Guide*, November 1995.
9. Bar-Shalom, Y. and Fortmann, T.E., *Tracking and Data Association*, Academic Press, New York, 1988.
10. Mutambara, A.G.O., *Decentralized Estimation and Control for Nonlinear Systems*, CRC Press, 1998.
11. Nicholson, D. and Deaves, R., Decentralized track fusion in dynamic networks, in *Proc. 2000 SPIE Aerosense Conf.,* 2000.
12. Bar-Shalom, Y. and Li, X.R., *Multitarget-Multisensor Tracking: Principles and Techniques*, YBS Press, Storrs, CT, 1995.
13. Julier, S.J. and Durrant-Whyte, H., A horizontal model fusion paradigm, *Proc. SPIE Aerosense Conf.,* 1996.
14. Uhlmann, J., Julier, S., and Csorba, M., Nondivergent simultaneous map building and localization using covariance intersection, in *Proc. 1997 SPIE Aerosense Conf.,* 1997.
15. Julier, S.J., Uhlmann, J.K., and Durrant-Whyte, H.F., A new approach for the nonlinear transformation of means and covariances in linear filters, *IEEE Trans. Automatic Control*, 477, March 2000.
16. Julier, S.J., Uhlmann, J.K., and Durrant-Whyte, H.F., A new approach for filtering nonlinear systems, in *Proc. American Control Conf.,* Seattle, WA, 1995, 1628.
17. Julier, S.J. and Uhlmann, J.K., A new extension of the Kalman filter to nonlinear systems, in *Proc. AeroSense: 11th Internat'l. Symp. Aerospace/Defense Sensing, Simulation and Controls*, SPIE, 1997.
18. Julier, S.J. and Uhlmann, J.K., A consistent, debiased method for converting between polar and Cartesian coordinate systems, in *Proc. of AeroSense: 11th Internat'l. Symp. Aerospace/Defense Sensing, Simulation and Controls*, SPIE, 1997.

19. Juliers, S.J., A skewed approach to filtering, *Proc. AeroSense: 12th Internat'l. Symp. Aerospace/Defense Sensing, Simulation and Controls*, SPIE, 1998.
20. Julier, S.J., and Uhlmann, J.K., A General Method for Approximating Nonlinear Transformations of Probability Distributions, published on the Web at http://www.robots.ox.ac.uk/~siju, August 1994.
21. Julier, S.J. and Uhlmann, J.K., A non-divergent estimation algorithm in the presence of unknown correlations, *American Control Conf.,* Albuquerque, NM, 1997.
22. Julier, S.J. and Uhlmann, J.K., Generalized and split covariance intersection and addition, Technical Disclosure Report, Naval Research Laboratory, 1998.