Image Cover Sheet

CLASSIFICATION	SYSTEM NUMBER
UNCLASSIFIED	518346

TITLE

A Comparative Study of Data Fusion Algorithms for Target Tracking

System Number:

Patron Number:

Requester:

Notes:

DSIS Use only:

Deliver to: CL

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
OCT 2001		

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
A Comparative Study of Data Fusion Algorithms for Target Tracking	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Defence R&D Canada - Valcartier,2459 Pie-XI Blvd North,Quebec (Quebec) G3J 1X5 Canada, ,**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

An ongoing activity undertaken by the Situation Analysis Support Systems (SASS) Group in the Decision Support Systems (DSS) Section at Defence Research & Development Canada (DRDC) - Valcartier is the investigation of advanced concepts of data fusion and sensor management processes, and their adaptation and integration. These concepts could apply to the current Halifax Class frigates Above Water Warfare (AWW) sensor smte, as well as its possible future upgrades, in order to improve its performance against predicted future threats. As part of this exploration, a comparative study of the commonly used track-level fusion techniques/algorithms is reported here. In developing data fusion systems for surveillance and tracking purposes, the selection of such techniques and/or algorithms is a fundamental conceptual issue. For any given sensor suite configuration, there can be many different ways to combine data from the sensors into global tracks, and various trade-offs are generally required for the selection of the appropriate data fusion architecture, since each approach has its benefits and disadvantages. The considered methods are compared through the target tracking problem that presents both of the mdependent and correlated source fusion problems.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**75**	

THIS PAGE IS LEFT BLANK

THIS PAGE IS LEFT BLANK

A comparative study of data fusion algorithms for target tracking

Abder Rezak Benaskeur and Jean Roy
Decision Support Systems Section

Defence Research & Development Canada - Valcartier
Technical Report
TR 2001-224
2002-10-31

Author

A.L. BENASKEUR/R

Abder Benaskeur

Approved by

Éloi Bossé

Head/DSS Section

Abstract (U)

An ongoing activity undertaken by the Situation Analysis Support Systems (SASS) Group in the Decision Support Systems (DSS) Section at Defence Research & Development Canada (DRDC) - Valcartier is the investigation of advanced concepts of data fusion and sensor management processes, and their adaptation and integration. These concepts could apply to the current Halifax Class frigates Above Water Warfare (AWW) sensor suite, as well as its possible future upgrades, in order to improve its performance against predicted future threats. As part of this exploration, a comparative study of the commonly used track-level fusion techniques/algorithms is reported here. In developing data fusion systems for surveillance and tracking purposes, the selection of such techniques and/or algorithms is a fundamental conceptual issue. For any given sensor suite configuration, there can be many different ways to combine data from the sensors into global tracks, and various trade-offs are generally required for the selection of the appropriate data fusion architecture, since each approach has its benefits and disadvantages. The considered methods are compared through the target tracking problem that presents both of the independent and correlated source fusion problems.

Résumé (U)

Une activité de recherche entreprise par le Groupe systèmes d'aide à l'analyse de la situation (SAAS) de la Section systèmes d'aide à la décision (SAD), à Recherche et Dèvelopment pour la Défence Canada (RDDC) - Valcartier, concerne l'étude de concepts avancés de fusion des données et de gestion des capteurs, ainsi que de leur gestion et leur intégration. Ces concepts pourraient s'appliquer aux détecteurs et capteurs du système de guerre anti-aérienne et de surface des frégates de classe Halifax, ainsi qu'à ses améliorations possibles à venir, afin d'en accroître la performance contre les menaces futures anticipées. Le volet rapporté dans ce document concerne une étude comparative des techniques et des algorithmes de fusion de pistes les plus utilisés. La sélection de tels algorithmes et techniques représente un élément fondamental dans la conception des systèmes de fusion de données pour la surveillance et le pistage. Pour chaque configuration de capteurs, différentes approches peuvent être utilisées pour l'agrégation en pistes globales de données fournies pas les mesures. Aussi, étant donné que chaque approche présente ses avantages et inconvénients, des compromis sont souvent nécessaires dans la sélection de l'architecture la plus adéquate. Les méthodes retenues sont comparées dans le contexte du pistage de cibles, qui présente à la fois les deux situations : la fusion de sources corrélées et la fusion de sources indépendantes.

This page intentionally left blank.

Executive summary (U)

The data fusion architecture is an important issue in developing surveillance and tracking systems, since the benefits of the data fusion process are different depending on the way the source data are combined. The selection of appropriate data fusion algorithms and techniques also depends on the underlying architecture, which can range from highly centralized to highly distributed. The architecture classification depends upon the level at which the sensor data are fused. One possible type of architecture is based on maintaining sensor-level tracks using local information at each sensor site and combining the resulting tracks into centralized global tracks. This architecture is typically referred to as track-level fusion.

The primary alternative architecture assumes that all of the raw sensor measurements are sent directly to the centralized fusion node to be combined into global tracks. This architecture is typically referred to as contact-level fusion. Since in the latter the observation errors of the sources to be fused are statistically independent, the Kalman filter provides an optimal method for fusing information. This optimality will however be no longer guaranteed, if the errors of the fused sources are correlated, as in the case of the track-level fusion. However, if the cross covariance matrix is available, the independence assumption imposed by the Kalman filter can be relaxed. The filter will then still provide an optimal solution, by exploiting this additional information. If the correlation information is missing, the Kalman filter cannot theoretically be applied.

In many situations, to allow for the use of the Kalman filter, the error independence is often assumed and the correlation is simply ignored in the estimation process. We then talk about the "simple fusion" approach that makes the filter over optimistic, that is, it underestimates the covariance matrix. This phenomenon is referred to as the inconsistency of the estimation and may, in many situations, lead to divergence.

Different approaches are available to handle the inconsistency and they all result in an increase of the estimated error covariance matrix. Some methods try to directly increase the estimated covariance matrix, by introducing an empirically determined parameter. Since no rigorous method exists for choosing this parameter, the stability and the reliability of the fusion process can be greatly compromised. Some other methods aim at estimating an upper limit of the actual covariance to avoid underestimating it. The covariance intersection method and the largest ellipsoid method both fall within this category. They are both based upon an estimation of the intersection region for the covariances that represents, independently of the cross covariance, an upper bound for the actual error covariance. Another solution consists in estimating the missing cross covariance matrix to allow for the application of a general form of the Kalman filter. An example of algorithms for computing the cross covariance matrix is given by the weighted covariance method. Finally, since the inconsistency phenomenon is mainly caused by the double counting of the redundant information, an interesting solution consists in removing this redundancy. The tracklet fusion approach and the information filter method are two very practical methods that use very similar algorithms to achieve

this task. These methods are all compared in this document through the target tracking problem that presents both of the independent and correlated source fusion problems.

Abder Rezak Benaskeur and Jean Roy. 2002. A comparative study of data fusion algorithms for target tracking. TR 2001-224. Defence Research & Development Canada - Valcartier.

Sommaire (U)

L'architecture de fusion des données est cruciale pour la mise au point de systèmes de surveillance et de poursuite, car les avantages du processus de fusion des données varient selon la façon dont sont combinées les données des sources. Le choix des algorithmes et techniques appropriés de fusion des données dépend également de l'architecture sous-jacente, qui peut être largement centralisée ou décentralisée. Le type d'architecture dépend du type des données qui sont fusionnées. Il existe un type d'architecture basée sur le maintien de pistes au niveau des capteurs utilisant uniquement l'information locale. Par la suite les pistes locales sont combinées en pistes globales centralisées. On désigne ce type d'architecture comme la fusion au niveau des pistes.

L'alternative à ce type d'architecture présuppose que toutes les mesures brutes des capteurs sont envoyées directement à un noeud central de fusion afin de les combiner en pistes globales. On désigne ce type d'architecture comme la fusion au niveau des contacts. Les erreurs d'observation des sources à fusionner étant statistiquement indépendantes, le filtre de Kalman constitue la méthode optimale de fusion d'information dans ce type d'architecture. On ne peut toutefois plus s'attendre à obtenir le degré optimal, lorsque les erreurs des sources à fusionner sont corrélées, comme c'est le cas dans la fusion de pistes. Par contre, la condition d'indépendance imposée par le filtre de Kalman peut être relaxée lorsque la matrice d'inter-covariance est disponible. Le filtre exploitant cette information additionnelle, celui-ci constitue toujours la meilleure solution. Si l'inter-covariance n'est pas disponible, le filtre de Kalman ne peut être utilisé, du moins en théorie.

Dans bien des cas, afin de permettre l'utilisation du filtre de Kalman, on présuppose généralement que les erreurs sont indépendantes, et le processus d'estimation ne tient tout simplement pas compte de la corrélation. Dans ces cas, la simple fusion tend sous-estimer la matrice de covariance de l'erreur. On parle alors d'inconsistance de l'estimation.

Diverses méthodes permettent d'éviter cette inconsistance. Toutes ces méthode entraînent un accroissement de l'estimé de la matrice de covariance de l'erreur. Certaines d'entre elles tentent d'accroître directement l'estimé de la matrice de covariance de l'erreur par l'introduction d'un paramètre déterminé de façon empirique. Comme il n'existe pas de méthode rigoureuse pour déterminer ce paramètre, la stabilité et la fiabilité du processus de fusion peuvent s'en trouver gravement compromises. D'autres méthodes permettent de déterminer la limite supérieure de la covariance réelle, ce qui évite de ne pas la sous-estimer. La méthode de l'intersection des covariances et celle du plus grand ellipsoïde présentent toutes deux cette caractéristique. Les deux se fondent sur l'évaluation de la zone d'intersection des covariances, laquelle représente la limite supérieure de la covariance réelle. Une autre solution consiste à évaluer la matrice d'inter-covariance manquante afin de permettre l'utilisation dans le filtre de Kalman. La méthode de la covariance pondérée comporte des exemples d'algorithmes

permettant de déterminer la matrice d'inter-covariance. Enfin, puisque l'incohérence est surtout attribuable au décompte d'information redondante, on peut simplement la supprimer. La méthode des tracklets et celle du filtre d'information constituent deux façons très pratiques d'utiliser des algorithmes tout à fait similaires pour y arriver. La poursuite de cibles, qui présente à la fois les caractéristiques des sources indépendantes et corrélées, permet de comparer toutes ces méthodes entre elles.

Abder Rezak Benaskeur and Jean Roy. 2002. Étude comparative des algorithmes de fusion de données pour le pistage des cibles. TR 2001-224. Recherche et Dèvelopment pour la Défence Canada - Valcartier.

Table of contents (U)

List of figures (U)

List of tables (U)

This page intentionally left blank.

1. Introduction

Recent years have witnessed a rapid development in the measurement and detection technology. Nonetheless, no one of the available systems can be trusted enough to be used as the single source, since they all depend on noisy, incomplete and/or inaccurate input. This is even truer for critical situations, such as military applications. Therefore, to achieve better accuracy, data from several sources are often combined. This aggregation/combination, that allows for an efficient extraction of the desired information, is referred to as "data fusion". In most situations, it yields better accuracy than could be achieved by single-source-based systems. Even though the need for data fusion is nowadays recognized in many areas, applications to which it has up to recently been applied have been mostly military in nature. In such a context, data fusion provides the decision maker with an efficient tool to manage the information he/she might receive from a variety of sources and improve his/her current situation awareness, by helping in producing an "as accurate as possible" explanatory picture of the battle environment.

An ongoing activity undertaken by the Situation Analysis Support Systems (SASS) Group in the Decision Support Systems (DSS) Section at Defence Research & Development Canada (DRDC) - Valcartier is the investigation of advanced concepts of data fusion and sensor management processes, and their adaptation and integration. These concepts could apply to the current Halifax Class frigates Above Water Warfare (AWW) sensor suite, as well as its possible future upgrades, in order to improve its performance against predicted future threats [1]. As part of this exploration, a highly modular testbed, called CASE_ATTI (Concept Analysis and Simulation Environment for Automatic Target Tracking and Identification) has been developed [2, 3]. CASE_ATTI provides the algorithm-level test and replacement capability required to study and compare the applicability and performance of advanced state-of-the-art data fusion techniques.

A fundamental conceptual issue in developing a data fusion system for surveillance and tracking purpose is the selection of an appropriate architecture [4–7]. For any given sensor suite configuration, there can be many different ways to combine data from the sensors into global tracks. This report presents a comparative study, making use of CASE_ATTI, of the main fusion techniques reported in the literature.

1.1 Fusion architecture

The data fusion architecture is an important issue in developing surveillance and tracking systems, since the benefits of the fusion process are different depending on the way the sensor data are combined [8]. The selection of the appropriate fusion algorithms and techniques also depends on the underlying architecture [8]. Hence, before a data fusion algorithm can be implemented within a combat system, it must be analyzed in terms of the different types of architectures and implementations that are

possible, the benefits and drawbacks of these architectures and, finally, in terms of how all this relates to the performance and mission requirements of the platform.

Within the surveyed literature, many different ways to combine data from multiple sources have been found, offering as many architectural options to the data fusion system designer [4–9]. The architecture of such a system can range from highly centralized (or monolithic) to highly distributed [10]. The classification depends upon the level at which the sensor data are fused (*i.e.*, signal, contact or track level).

One possible type of architecture is based on maintaining sensor-level tracks using local sensor information at each sensor site, finding (in a central fusion resource) the sensor tracks that potentially represent the same target and then combining these tracks into global tracks. This architecture is typically referred to as track-level fusion or decentralized data fusion. The primary alternative architecture assumes that all of the raw sensor measurements (*i.e.*, sensor contacts) are sent directly to the centralized fusion node to be combined into global tracks. This architecture is typically referred to as contact-level fusion or centralized data fusion. Although these two basic architectural approaches have predominated most published work, fusion at the signal-level is also possible. However, a discussion of this latter approach is out of the scope of this work.

1.2 Source modeling

In practice, the sources to be fused are often assumed corrupted by noises and therefore modeled as random variables, whose true statistics are unknown. Within the estimation theory framework, such stochastic variables are often represented in terms of means, or estimates, and covariance matrices, that is, the uncertainties associated with the estimation process. In this context, the main idea of data fusion is to get an estimate of some unknown variable from its two (or more) available noise-corrupted, direct or indirect, observations. In the case of two sources, s_1 and s_2, the statistical representation of the available data (observations) is given by the means $\hat{s}_1$ and $\hat{s}_2$, the assumed error covariance of each source

$$\hat{P}_1 = E\left[\tilde{s}_1 \tilde{s}_1^T\right] \tag{1}$$

$$\hat{P}_2 = E\left[\tilde{s}_2 \tilde{s}_2^T\right] \tag{2}$$

and the assumed cross covariance of the two sources

$$\hat{P}_c = E\left[\tilde{s}_1 \tilde{s}_2^T\right] \tag{3}$$

where the unknown estimation error terms, $\tilde{s}_1$ and $\tilde{s}_2$, are defined by

$$\tilde{s}_1 = \hat{s}_1 - s_1 \tag{4}$$

$$\tilde{s}_2 = \hat{s}_2 - s_2 \tag{5}$$

The fusion problem can then be characterized as finding a mathematical method to describe how the information about $\hat{s}_1$ and $\hat{s}_2$ can be used, in an efficient and provably optimal way, to construct a new estimate $\hat{s}$ (and an estimate of its associated measure of accuracy $\hat{P}$) that minimizes some cost function. This updated estimate is often obtained via a linear combination of the two available estimates $\hat{s}_1$ and $\hat{s}_2$.

$$\hat{s} = W_1\hat{s}_1 + W_2\hat{s}_2 \tag{6}$$

where the weights W_1 and W_2 are computed to minimize some norm of the resulting error covariance matrix $\hat{P}$, whose expression is given by

$$\hat{P} = W_1\hat{P}_1W_1^T + W_1\hat{P}^cW_2^T + W_2(\hat{P}^c)^TW_1^T + W_2\hat{P}_2W_2^T \tag{7}$$

1.3 Fusion of independent sources

If the sources to be fused are statistically independent, that is $P^c = 0$, the combined error covariance matrix (7) reduces to

$$\hat{P} = W_1\hat{P}_1W_1^T + W_2\hat{P}_2W_2^T \tag{8}$$

and an interesting solution to the above described optimization problem is then given by the weights of the Kalman filter [11]

$$W_1 = \hat{P}_2\left[\hat{P}_1 + \hat{P}_2\right]^{-1} = \left[\hat{P}_1^{-1} + \hat{P}_2^{-1}\right]^{-1}\hat{P}_1^{-1} \tag{9}$$

$$W_2 = \hat{P}_1\left[\hat{P}_1 + \hat{P}_2\right]^{-1} = \left[\hat{P}_1^{-1} + \hat{P}_2^{-1}\right]^{-1}\hat{P}_2^{-1} \tag{10}$$

Incontestably, the Kalman filter is one of the most important data fusion tools. If the sources to be fused are actually statistically independent[1], this filter provides, regardless of the underlying error distribution, a mathematically rigorous and provably stable method for fusing information in real-time, and performs in the Minimum Mean Square Error sense, better than any other linear filter. The Kalman filter results in the following fusion algorithm.

$$\hat{P}^{-1} = \hat{P}_1^{-1} + \hat{P}_2^{-1} \tag{11}$$

$$\hat{P}^{-1}\hat{s} = \hat{P}_1^{-1}\hat{s}_1 + \hat{P}_2^{-1}\hat{s}_2 \tag{12}$$

It is proven that the quality of the estimate yielded by the algorithm (11)-(12) improves, or at least remains constant, over the estimation process.

[1]Two random variables are independent, if their joint probability density is the product of their marginal (individual) probability densities.

1.4 Fusion of correlated sources

The optimality of the above-given Kalman filter will no longer be guaranteed, if the fused sources are correlated. This may even compromise the convergence of the estimation process. Nevertheless, if the cross covariance matrix (P^c), or at least an estimate of it, is available, the independence assumption imposed by the Kalman filter can be relaxed. The filter will then still provide an optimal solution, by exploiting this additional information. The resulting weights are given by

$$\begin{aligned} W_1 &= (\hat{P}_2 - (P^c)^T)\left[\hat{P}_1 + \hat{P}_2 - P^c - (P^c)^T\right]^{-1} \\ &= \left[(\hat{P}_1 - P^c)^{-1} + (\hat{P}_2 - (P^c)^T)^{-1}\right]^{-1}(\hat{P}_1 - P^c)^{-1} \end{aligned} \tag{13}$$

and

$$\begin{aligned} W_2 &= (\hat{P}_1 - P^c)\left[\hat{P}_1 + \hat{P}_2 - P^c - (P^c)^T\right]^{-1} \\ &= \left[(\hat{P}_1 - P^c)^{-1} + (\hat{P}_2 - (P^c)^T)^{-1}\right]^{-1}(\hat{P}_2 - (P^c)^T)^{-1} \end{aligned} \tag{14}$$

One can easily show that the resulting error covariance matrix is given by

$$\hat{P} = \hat{P}_1 - \left[\hat{P}_1 - P^c\right]\left[\hat{P}_1 - P^c + \hat{P}_2 - (P^c)^T\right]^{-1}\left[\hat{P}_1 - (P^c)^T\right] \tag{15}$$

Note that the introduction of the cross correlation matrix in the Kalman filter is beneficial only if this matrix is definite positive. Furthermore, if the correlation information is missing, the Kalman filter cannot theoretically be applied. In most situations, to allow the use of the Kalman filter, the independence is often assumed and the correlation is simply ignored in the estimation process. This approach, presented in Sub-section 3.1.2 and known as the simple fusion, makes the filter "over optimistic", that is, it underestimates the actual error covariance matrix. We then talk about the inconsistency of the estimation, that may in many situations lead to divergence [12]. This phenomenon is due to the double counting of the information, since the fusion algorithm may be handling redundant pieces of information, while treating them as independent. Different approaches are available to handle the inconsistency problem of the simple fusion. These methods result all in an increase of the estimated error covariance matrix, compared to the simple fusion. As shown below, the amount of this increase and the way it is computed differs from one method to another.

1.4.1 Stabilizing noise

This approach consists in increasing directly the estimated covariance matrix, by introducing an empirically determined parameter, also called "fudge factor" [13]. Since there is no rigorous method for choosing this parameter,

the stability and the reliability of the data fusion process can be greatly compromised. This method is only mentioned for completeness and will not be considered in the remainder.

1.4.2 Upper bound

Instead of artificially increasing the estimated error covariance, some methods aim at finding an estimate for an upper limit of the actual error covariance, to avoid underestimating it. The covariance intersection method [14] and the recently proposed largest ellipsoid method [15], fall both within this category. They are both based upon an estimation of the intersection region that represents, independently of the cross covariance, an upper bound for the actual error covariance. These methods result often in a loose of performance, due to the overestimation of the actual covariance. They however have both the advantage of being independent of the fusion network architecture and the information flow. This makes them more widely applicable than the two methods presented below.

1.4.3 Estimation of the cross covariance

Another solution consists in estimating the missing cross covariance matrix, to allow for the application of the general form of the Kalman filter given by the equations (13)–(15). An example of algorithms for computing the cross covariance matrix is given by the weighted covariance method [16, 17]. Since maintaining such a covariance matrix is not possible in any arbitrary architecture [18], this method is only applicable in the case of "small" networks, where the information flow is well defined.

1.4.4 Removal of the redundant information

Since the inconsistency phenomenon is mainly caused by the double counting of the redundant information, an interesting solution consists in removing this redundancy. The tracklet fusion approach [19, 20] and the information filter method [21] are two very practical methods that use very similar algorithms to achieve this task. Due to the amount of information required for their implementation, these methods are also only applicable in the case of "small" networks, with a known information flow.

1.5 Application to track fusion

To allow for the comparison of the above-described methods, an application to the target's tracking problem is considered. This comparison is mainly based on the estimation accuracy and consistency. Figure 1 gives the architecture of the fusion network considered. Each local node has its own fusion mechanism that generates an estimate (or a track) based only on its locally available measurement data (or contacts).

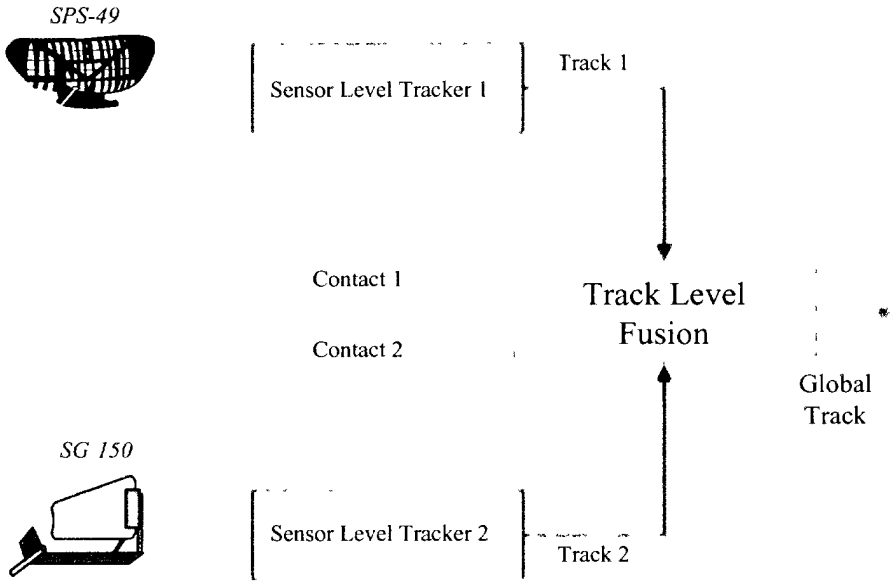

Figure 1: *Fusion network architecture*

We assume that both nodes send their tracks and contacts to the central node that fuses them, in some way, to obtain a combined track. Such a tracking problem presents both of the above-mentioned situations, that is, independent and correlated source fusion problems. The independent source case occurs at the sensor level, where the fused contacts are actually independent. As shown in Chapter 2, this problem can be optimally solved by the standard Kalman filter.

The correlated source situation is encountered at the central tracker level. The case, where the latter uses only the contacts, and ignores the local tracks, is known as the contact-level fusion, and falls within the independent source category. The standard Kalman filter can therefore optimally solve it. Furthermore, since it yields the best performance (by using the standard Kalman filter), it will serve as a reference to evaluate the performance of the presented methods. A more challenging problem occurs however when the central node fuses the sensor-level tracks, instead of the contacts. Since the sensor-level trackers share, at least, the same process noise, their tracks are correlated. The predicted estimate of the central tracker is also correlated with each of these sensor-level tracks, since they share the same error history. This problem of track-level fusion falls therefore within the correlated source case (see Chapter 3).

This report is organized as follows. The Kalman filter is first introduced in Chapter 2, as an optimal solution to the contact-level fusion problem. In Chapter 3 is presented the track-level fusion problem. The inconsistency of the simple fusion is shown, and alternative solutions are presented. In Chapter 4, these solutions are compared for the target's tracking problem. The conclusion is presented in Chapter 5. Some useful

materials and further developments will also be presented in Appendices A and B.

2. Contact-level fusion

To illustrate the contact-level fusion problem, the estimation of the target's kinematic properties will be considered. In such a context, the contact-level fusion consists in combining each new observation with the available estimate of the target's state vector, to provide an "improved" estimate of the actual target's kinematics. Since each observation is statistically independent from the others and from the maintained state estimate, the fusion problem falls within the independent source class. Therefore, as discussed in Chapter 1, the Kalman filter provides an optimal solution.

2.1 Target's dynamical model

The dynamics of a target, whether it is an aircraft, a missile or a ship, can be described by the evolution of its *state*. The latter refers to a set of variables $\boldsymbol{x}$ that, at a specific instant of time k, describe the target's kinematic properties, which are partially reported by a sensor or a set of sensors. Besides the target position estimate (*e.g.*, range and bearing, bearing and elevation, etc.), the reported information, which is commonly referred to as a contact, may also contain a time tag, an estimate of the measurement quality (S/N ratio, variance estimate, etc.) and, potentially, target identity information data (e.g., IFF reply data). The target's dynamical behavior is often modelled using the dynamical equation

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k \boldsymbol{x}_k + \boldsymbol{G}_k \boldsymbol{u}_k + \boldsymbol{v}_k \tag{16}$$

This equation, known as the linearized state equation, may be prone to random disturbances $\boldsymbol{v}$, resulting from the model simplification[2] and/or the target's maneuvers. The matrix $\boldsymbol{F}_k$ and $\boldsymbol{G}_k$ are respectively the state transition matrix[3] and the control/calibration gain matrix. The equation (16) tells how the future state vector $\boldsymbol{x}_{k+1}$ of the target evolves from its current value $\boldsymbol{x}_k$ when it is affected by the control $\boldsymbol{u}_k$ and the process noise $\boldsymbol{v}_k$. In practice, the individual state variables $\boldsymbol{x}$ cannot always be determined exactly by direct measurements. The observations usually provide only a partial information about the complete state, *i.e.*, they are "incomplete" observations. Furthermore, since the sensors do not provide perfect data, the measurements (or contacts) are often, if not always, corrupted by random noises $\boldsymbol{w}$

$$\boldsymbol{z}_k = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{w}_k \tag{17}$$

This equation is known as the linearized measurement (or observation) equation, where $\boldsymbol{H}_k$ is the observation matrix.

The tracking problem then consists in combining the predicted state vector $\hat{\boldsymbol{x}}_{k+1|k}$ and the newly available data $\boldsymbol{z}_{k+1}$, to provide a refined estimate $\hat{\boldsymbol{x}}_{k+1|k+1}$ of the state vector.

[2]The objective of the model is to represent only the dominant modes of the dynamics.

[3]For non-linear systems, the matrices $\boldsymbol{F}_k$, $\boldsymbol{G}_k$ and $\boldsymbol{H}_k$ are the Jacobians, obtained from the partial derivatives of the actual non-linear dynamics.

The measurement noise w and the process noise v are assumed zero-mean, that is,

$$E\left[w_k\right] = 0 \qquad \text{and} \qquad E\left[v_k\right] = 0 \tag{18}$$

and not auto–, *i.e.*,

$$E\left[w_k w_l^T\right] = R\delta(k-l) \qquad \text{and} \qquad E\left[v_k v_l^T\right] = Q\delta(k-l) \tag{19}$$

nor cross–, *i.e.*,

$$E\left[v_k w_l^T\right] = 0, \qquad \forall k, l \tag{20}$$

correlated. Under these assumptions, the Kalman filter provides the best unbiased estimate $\hat{x}_{k+1|k+1}$ of the actual state x_{k+1}, given the statistics of the noises v and w.

2.2 Kalman filter

Within the Kalman filtering framework, the fusion problem can be reformulated as a linear combination of the *a priori* estimate $\hat{x}_{k+1|k}$ of the state x_{k+1} and the measured variable z_{k+1}. These represent the (independent) pieces of information to be fused, and whose fusion results in the updated (or *a posteriori*) estimate $\hat{x}_{k+1|k+1}$ given by

$$\hat{x}_{k+1|k+1} = W_x \hat{x}_{k+1|k} + W_z x_z \tag{21}$$

From (9) and (10), the Kalman gains are rewritten as

$$W_x = \left[\hat{P}_{k+1|k}^{-1} + P_z^{-1}\right]^{-1} \hat{P}_{k+1|k}^{-1} \tag{22}$$

$$W_z = \left[\hat{P}_{k+1|k}^{-1} + P_z^{-1}\right]^{-1} P_z^{-1} \tag{23}$$

where

$$\hat{P}_{k+1|k} = E\left[\left(\hat{x}_{k+1|k} - x_{k+1}\right)\left(\hat{x}_{k+1|k} - x_{k+1}\right)^T\right] \tag{24}$$

is the *a priori* estimation error covariance matrix[4], and

$$P_z = E\left[\tilde{x}_z \tilde{x}_z^T\right] \tag{25}$$

is the error covariance of the (indirectly) measured state. The combined estimate covariance is then rewritten as

$$\hat{P}_{k+1|k+1} = \left[\hat{P}_{k+1|k}^{-1} + P_z^{-1}\right]^{-1} \tag{26}$$

[4]This is the predicted value of the covariance at time instant $k+1$, based on observations up to, and including, the time instant k.

2.2.1 Covariance computation

The *a priori* estimation error is given by

$$\tilde{\boldsymbol{x}}_{k+1|k} = \hat{\boldsymbol{x}}_{k+1|k} - \boldsymbol{x}_{k+1} \tag{27}$$

$$= \boldsymbol{F}_k \tilde{\boldsymbol{x}}_k - \boldsymbol{v}_k \tag{28}$$

which leads to the following *a priori* covariance matrix

$$\hat{\boldsymbol{P}}_{k+1|k} = E\left[\left(\boldsymbol{F}_k \tilde{\boldsymbol{x}}_k - \boldsymbol{v}_k\right)\left(\boldsymbol{F}_k \tilde{\boldsymbol{x}}_k - \boldsymbol{v}_k\right)^T\right] \tag{29}$$

$$= \boldsymbol{F}_k E\left[\tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_k^T\right] \boldsymbol{F}_k^T + E\left[\boldsymbol{v}_k \boldsymbol{v}_k^T\right] - \boldsymbol{F}_k E\left[\tilde{\boldsymbol{x}}_k \boldsymbol{v}_k^T\right] - E\left[\boldsymbol{v}_k \tilde{\boldsymbol{x}}_k^T\right] \boldsymbol{F}_k^T \tag{30}$$

Under the error/noise independence assumption, the last two terms are zero, and the covariance reduces to

$$\hat{\boldsymbol{P}}_{k+1|k} = \boldsymbol{F}_k \hat{\boldsymbol{P}}_{k|k} \boldsymbol{F}_k^T + \boldsymbol{Q} \tag{31}$$

where $\hat{\boldsymbol{P}}_{k|k}$ is the value at time instant k of the assumed *a posteriori* covariance matrix. Now, even if the state is (in general) not directly accessible to the measurement, its expression will be used to compute the measurement error covariance matrix. The available information about the target's state is contained in the observation vector $\boldsymbol{z}_{k+1}$ that is related to the state by the observation equation (17). The (non-completely) measured state can then be expressed as follows

$$\boldsymbol{x}_z = \boldsymbol{H}_{k+1}^{-1} \boldsymbol{z}_{k+1} \tag{32}$$

$$= \boldsymbol{x}_{k+1} + \boldsymbol{H}_{k+1}^{-1} \boldsymbol{w}_{k+1} \tag{33}$$

The measurement error is then given by

$$\tilde{\boldsymbol{x}}_z = \boldsymbol{H}_{k+1}^{-1} \boldsymbol{w}_{k+1} \tag{34}$$

The covariance of the measurement error can be written as

$$\boldsymbol{P}_z = E\left[\left(\boldsymbol{H}_{k+1}^{-1} \boldsymbol{w}_{k+1}\right)\left(\boldsymbol{H}_{k+1}^{-1} \boldsymbol{w}_{k+1}\right)^T\right] \tag{35}$$

$$= \boldsymbol{H}_{k+1}^{-1} \boldsymbol{R} (\boldsymbol{H}_{k+1}^{-1})^T \tag{36}$$

which can be expressed in the information (inverse) space as

$$\boldsymbol{P}_z^{-1} = \boldsymbol{H}_{k+1}^T \boldsymbol{R}^{-1} \boldsymbol{H}_{k+1} \tag{37}$$

2.2.2 Inverse form

Gathering the equations (21) to (37) leads to the following algorithm for the Kalman filter, known as the inverse form. This is very useful representation, also known as the information space representation.

$$\hat{P}^{-1}_{k+1|k+1} = \left[F_k \hat{P}_{k|k} F_k^T + Q\right]^{-1} + H_{k+1}^T R^{-1} H_{k+1} \tag{38}$$

$$\hat{P}^{-1}_{k+1|k+1}\hat{x}_{k+1|k+1} = \left[F_k \hat{P}_{k|k} F_k^T + Q\right]^{-1}\left[F_k \hat{x}_{k|k} + G_k u_k\right] + H_{k+1}^T R^{-1} z_{k+1} \tag{39}$$

2.2.3 Direct form

Another representation of the Kalman filter, known as the direct form or the covariance space representation, can be obtained by considering the expression of the *a posteriori* covariance matrix

$$\hat{P}^{-1}_{k+1|k+1} = \hat{P}^{-1}_{k+1|k} + H_{k+1}^T R^{-1} H_{k+1} \tag{40}$$

to which is applied the following matrix inversion lemma.

$$(A^{-1} + BDC)^{-1} = A - AB(CAB + D^{-1})^{-1}CA \tag{41}$$

This results in

$$\hat{P}_{k+1|k+1} = \hat{P}_{k+1|k} - \hat{P}_{k+1|k} H_{k+1}^T \left[H_{k+1}\hat{P}_{k+1|k}H_{k+1}^T + R\right]^{-1} H_{k+1}\hat{P}_{k+1|k} \tag{42}$$

$$= \hat{P}_{k+1|k} - W_{k+1} H_{k+1} \hat{P}_{k+1|k} \tag{43}$$

where

$$W_{k+1} = \hat{P}_{k+1|k} H_{k+1}^T \left[H_{k+1}\hat{P}_{k+1|k}H_{k+1}^T + R\right]^{-1} \tag{44}$$

$$= \left[\hat{P}^{-1}_{k+1|k} + H_{k+1}^T R^{-1} H_{k+1}\right]^{-1} H_{k+1}^T R^{-1} \tag{45}$$

$$= \hat{P}_{k+1|k+1} H_{k+1}^T R^{-1} \tag{46}$$

is what is commonly referred to as the Kalman gain matrix. From equations (21) and (26), the *a posteriori* estimate can then be expressed as

$$\hat{x}_{k+1|k+1} = \hat{P}_{k+1|k+1}\left[\hat{P}^{-1}_{k+1|k}\hat{x}_{k+1|k} + P_z^{-1} x_z\right] \tag{47}$$

$$= \hat{x}_{k+1|k} + \hat{P}_{k+1|k+1} P_z^{-1}\left[x_z - \hat{x}_{k+1|k}\right] \tag{48}$$

$$= \hat{x}_{k+1|k} + \hat{P}_{k+1|k+1} P_z^{-1} H_{k+1}^{-1}\left[z_{k+1} - H_{k+1}\hat{x}_{k+1|k}\right] \tag{49}$$

Using equations (37) and (46), the above update rule is rewritten in the following form

$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{W}_{k+1}\left[\boldsymbol{z}_{k+1} - \boldsymbol{H}_{k+1}\hat{\boldsymbol{x}}_{k+1|k}\right] \tag{50}$$

Grouping together equations (31), (43), (44) and (50) leads to the well-known "prediction-update cycle" (or the direct) form of the Kalman filter.

Prediction

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{F}_k\hat{\boldsymbol{x}}_{k|k} + \boldsymbol{G}_k\boldsymbol{u}_k \tag{51}$$

$$\hat{\boldsymbol{P}}_{k+1|k} = \boldsymbol{F}_k\hat{\boldsymbol{P}}_{k|k}\boldsymbol{F}_k^T + \boldsymbol{Q} \tag{52}$$

Update

$$\boldsymbol{W}_{k+1} = \hat{\boldsymbol{P}}_{k+1|k}\boldsymbol{H}_{k+1}^T\left[\boldsymbol{H}_{k+1}\hat{\boldsymbol{P}}_{k+1|k}\boldsymbol{H}_{k+1}^T + \boldsymbol{R}\right]^{-1} \tag{53}$$

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{W}_{k+1}\left[\boldsymbol{z}_{k+1} - \boldsymbol{H}_{k+1}\hat{\boldsymbol{x}}_{k+1|k}\right] \tag{54}$$

$$\hat{\boldsymbol{P}}_{k+1|k+1} = \left[\boldsymbol{I} - \boldsymbol{W}_{k+1}\boldsymbol{H}_{k+1}\right]\hat{\boldsymbol{P}}_{k+1|k} \tag{55}$$

2.2.4 Innovation form

Another widely used form of the Kalman filter uses the innovation[5] ($\boldsymbol{\nu}$) and its covariance matrix ($\boldsymbol{S}$). This allows rewriting the Kalman filter as given below.

Prediction

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{F}_k\hat{\boldsymbol{x}}_{k|k} + \boldsymbol{G}_k\boldsymbol{u}_k \tag{56}$$

$$\hat{\boldsymbol{P}}_{k+1|k} = \boldsymbol{F}_k\hat{\boldsymbol{P}}_{k|k}\boldsymbol{F}_k^T + \boldsymbol{Q} \tag{57}$$

Innovation

$$\boldsymbol{\nu}_{k+1} = \boldsymbol{z}_{k+1} - \boldsymbol{H}_{k+1}\hat{\boldsymbol{x}}_{k+1|k} \tag{58}$$

$$\boldsymbol{S}_{k+1} = \boldsymbol{H}_{k+1}\hat{\boldsymbol{P}}_{k+1|k}\boldsymbol{H}_{k+1}^T + \boldsymbol{R} \tag{59}$$

Update

$$\boldsymbol{W}_{k+1} = \hat{\boldsymbol{P}}_{k+1|k}\boldsymbol{H}_{k+1}^T\boldsymbol{S}_{k+1}^{-1} \tag{60}$$

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{W}_{k+1}\boldsymbol{\nu}_{k+1} \tag{61}$$

$$\hat{\boldsymbol{P}}_{k+1|k+1} = \hat{\boldsymbol{P}}_{k+1|k} - \boldsymbol{W}_{k+1}\boldsymbol{S}_{k+1}\boldsymbol{W}_{k+1}^T \tag{62}$$

[5] Or the residual.

2.3 Single sensor target tracking

In the case of a single sensor configuration, the tracking can be performed using any one of the above-given forms of the Kalman filter. These forms all lead to the same (optimal) solution, given that the independence assumptions are satisfied.

In the architecture of Figure 1, the single sensor configuration is used within the sensor-level trackers. This represents the simplest case.

2.4 Multi-sensor target tracking

When a centralized multi-sensor fusion architecture is used, each individual sensor transmits (with negligible delay through large bandwidth communication links) its raw observations to the fusion centre where the required processes are performed to generate and update global tracks, within a single master track file [8]. The central fusion process performs the functions of data alignment, data association and target state estimation. Because of the actual independence of the fused contacts, the contact-level fusion approach yields, as in the single source case, an optimal performance by using the standard Kalman filter. There are however three different solutions possible to fuse the independent contacts received by the central tracker from the sensors. Their performances are very similar (and even identical in the linear case) and only the computation requirements may differ from one solution to another [22].

2.4.1 Sequential fusion

A Kalman filter, as in the case of a single sensor configuration, treats all the measurements sequentially. The fusion of these measurements is however performed without a prediction. Only one prediction is computed for all the received (at the same time) measurements.

For the first sensor

$$\hat{P}^{-1}_{k+1|k+1} = \left[F_k \hat{P}_{k|k} F_k^T + Q\right]^{-1} + H^T_{1_{(k+1)}} R_1^{-1} H_{1_{(k+1)}} \tag{63}$$

$$\hat{x}_{k+1|k+1} = F_k \hat{x}_{k|k} + \hat{P}_{k+1|k+1} H^T_{1_{(k+1)}} R_1^{-1} \left[z_{1_{(k+1)}} - H_{1_{(k+1)}} F_k \hat{x}_{k|k}\right] \tag{64}$$

For the others ($i = 2$ to N)

$$\hat{P}^{-1}_{k+1|k+1} = \hat{P}^{-1}_{k+1|k+1} + H^T_{i_{(k+1)}} R_i^{-1} H_{i_{(k+1)}} \tag{65}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k+1} + \hat{P}_{k+1|k+1} H^T_{i_{(k+1)}} R_i^{-1} \left[z_{i_{(k+1)}} - H_{i_{(k+1)}} \hat{x}_{k+1|k+1}\right] \tag{66}$$

2.4.2 Parallel fusion

This method is based upon the use of a stacked vector of the measurements to process them in parallel. A composite equation is defined by

$$z_{k+1} = H_{k+1}x_{k+1} + w_{k+1} \tag{67}$$

where

$$z_{k+1} = \left[z^T_{1_{(k+1)}}, \ldots, z^T_{N_{(k+1)}}\right]^T \tag{68}$$

$$H_{k+1} = \left[H^T_{1_{(k+1)}}, \ldots, H^T_{N_{(k+1)}}\right]^T \tag{69}$$

$$w_{k+1} = \left[w^T_{1_{(k+1)}}, \ldots, w^T_{N_{(k+1)}}\right]^T \tag{70}$$

and

$$R = diag\left[R_1, \ldots, R_N\right] \tag{71}$$

These matrices and vectors are used as inputs (and parameters) to a standard Kalman filter.

$$\hat{P}^{-1}_{k+1|k+1} = \left[F_k\hat{P}_{k|k}F^T_k + Q\right]^{-1} + H^T_{k+1}R^{-1}H_{k+1} \tag{72}$$

$$\hat{x}_{k+1|k+1} = F_k\hat{x}_{k|k} + \hat{P}_{k+1|k+1}H^T_{k+1}R^{-1}\left[z_{k+1} - H_{k+1}F_k\hat{x}_{k|k}\right] \tag{73}$$

2.4.3 Data compression

All the contacts $(z_{i_{(k+1)}}, R_i)$ from the different sensors are first fused together to form unique vector z_{k+1} and covariance R, of the same dimension as $z_{i_{(k+1)}}$ and R_i respectively. The obtained information is used as a unique contact by a central Kalman filter. The compression algorithm is given

$$R^{-1} = \sum_{i=1}^{n} R_i^{-1} \tag{74}$$

$$R^{-1}z_{k+1} = \sum_{i=1}^{n} R_i^{-1}z_{i_{(k+1)}} \tag{75}$$

Note that the data compression and the parallel fusion apply only in the case of synchronous systems, *i.e.*, all the contacts from the sensors are received at the same time. The sequential fusion can be used in either synchronous or random transmission cases. A different prediction mechanism is however used in the random case. From the computation requirement viewpoint, the data compression is the most efficient, while the parallel fusion is the least [22].

2.5 Benefits of contact-level fusion

One advantage of the contact-level (or centralized) fusion architecture is the increased reaction time [8]. Track initiation, confirmation and reporting are much quicker with the centralized approach. All sensors that detect a target are used to initiate a track on the target and contribute to the promotion of the track to the status of firm. Also, the mutual support provided by the various sensors reduces the number of "holes" in the data due to sensor fades (*e.g.*, radar fade zones or IR sensor blockage). Moreover, the aggregate scan period of the suite of sensors is significantly less than the fastest sensor. For example, if a sensor is in a 10 seconds scan mode and another sensor is scanning the same volume at the rate of once every 10 seconds, then the average update rate for a target being seen by both sensors is once every 5 seconds. Hence, each composite track contains more data per unit of time than the corresponding sensor-level tracks. Thus, by using raw detections from all sensors for each track, the probability of confirming the track can be improved over that for a single sensor. The increased reaction time translates into longer range at target detection and track confirmation. Clearly, this is of high interest in a military environment.

The contact-level fusion architecture has also the advantage of good track continuity. In hard environmental conditions, each individual sensor may have a hard time to sustain a track. However, looking at the overall sensor suite, one sensor may be blocked while another has contact with the target being tracked. Hence, with multiple sensors updating a track, the probability is small that all sensors are experiencing a fade at the same point in space and time. It can typically be seen that one sensor "fills the gaps" of the other sensor's fade zones and that the combined probability of detection is better than the maximum probability of each sensor at each range. For example, in a system with a radar and an infrared (IR) sensor, IR detections can maintain a track that might otherwise be lost during a fade in the radar return [17]. The probability of sustaining a track is thus increased over that for a single sensor by using raw contacts from all sensors. Finally, as an additional factor that contributes to the track continuity, each composite track contains more data per unit of time than the corresponding sensor-level tracks. The data from each sensor is used immediately to confirm target maneuvers before the track is lost or corrupted [8].

Use of the maximum available information in the centralized architecture results in less track uncertainty, and potentially a corresponding decrease in mis-correlation [5, 8, 17]. For example, there will be a higher probability of correct correlation in areas affected by clutter because the gate sizes will be smaller due to more frequent track updates [17]. Furthermore, since the data from each sensor is used immediately to confirm target maneuvers before the track is lost or corrupted, the contact-level fusion approach should have good false track suppression [8]. Moreover, a centralized fusion system must, by necessity, incorporate advanced clutter suppression algorithms to offset the increased data rate available from multiple sensors [8]. These algorithms are specifically designed to suppress the types of clutter encountered by maritime surveillance sensors (sea, weather, chaff and land clutter). Finally, as presented above,

tracking algorithms conceptually similar to single-sensor tracking algorithms can be used to implement centralized fusion [5].

2.6 Drawbacks of contact-level fusion

The contact-level fusion approach suffers primarily from the large communication load resulting from the amount of raw observation data that must be transferred from the sensors to the central processor through a high bandwidth communication link. The computer processing power required to centrally process this raw data may also be very high. Other potential problems with the centralized fusion relate to military environments [5]. Centralized systems are highly vulnerable to failure (or destruction) because of the critical nature of the central node in such architecture. From a tactical perspective, because it represents a physically large, easily recognized critical element in the intelligence preparation and dissemination processes, a centralized system is highly vulnerable to attack [10]. Hence, the centralized approach requires backup processing and track files for survivability in case the platform containing the central track file fails [8]. Centralized systems are also vulnerable to degraded (or corrupted) sensor data. This problem occurs when the data from one sensor can become degraded (and this degradation not immediately sensed), and thus lead to poor central-level tracking. In this case, the possible combination of good data from un-degraded sensors with bad data, in effect, will negate the value of the good data [23].

With the contact-level fusion architecture, one seeks to determine if raw data are observations of the same entity. This data association process may be difficult because of the large amount of data that need to be processed simultaneously, and because of the dissimilarity between the sensors. Even the association of identical sensors may be difficult if the sensors are displaced geographically. Fusion of raw data requires commensurate sensor data [9]. Commensurate sensors observe the same physical manifestation of an entity. Examples of commensurate sensors include identical sensors or sensors whose observational data may be closely compared or merged (*e.g.*, an infrared image and a visual image). Finally, the implementation of the centralized fusion architecture typically requires modifications to the current autonomous sensors already in use to gain access to the raw contact data. These modifications may be costly.

3. Track-level fusion

The track-level (or decentralized) fusion architecture allows each sensor to perform a maximum amount of pre-processing to generate sensor output decisions (such as state vectors and declarations of identity) for the various entities in the environment [9]. Independent target detection, features extraction, state estimation and identification are thus potentially performed within the signal processor and tracker of each sensor [8]. The resulting target track data is normally stored in a track file. Hence, each sensor individually maintains its own track file based exclusively upon its own measurement data processed by the local tracker [23]. These sensor-level tracks are then transmitted to a central fusion process responsible for both finding the sensor tracks that likely represent the same target, and for combining or fusing these tracks into composite tracks to form a master track file. Within the central fusion process, data alignment, gating, assignment and fusion (positional and/or identity) are performed on state vectors rather than on raw data.

3.1 Track-level fusion algorithms

Fundamental to the problem of combining sensor-level tracks is to first determine whether two tracks from different systems (*i.e.*, different sensor track files) potentially represent the same target [23]. To accomplish this parametric association process, one must define for each track-to-track pairing an appropriate association metric that quantifies the closeness of the tracks. With the use of the Kalman filtering, the association metric can be a statistical distance applied to the track state vector differences for the two tracks. However, because the process noise introduced by the target behavior is observed by all sources tracking a common entity, Bar-Shalom [6] has shown that the track estimation errors may be correlated, and that this correlation must be considered in the association metric. After track-to-track associations have been determined, the local tracks that correspond to the same targets as seen by different sensors must eventually be combined at a later stage.

Nevertheless, if the central track is updated with sensor-level tracks, the usual assumption (valid for the case of raw measurements with uncorrelated measurement error) of error independence from one update period to another is not valid. The dependence between the estimation errors from the track files arises from the common process noise entering into the state equation of the tracking filters (*i.e.*, the common error source due to the target dynamics), given that the information processors follow the same target. For example, a sudden target maneuver can lead to a bias error for all the tracking filters. The fact that the measurement noise sequences processed by these different filters can be assumed independent is not sufficient to ensure the independence of the estimation errors in the track files. The tracks dependence can be taken into account in the processing, but it forces additional complexity [6, 23]. Techniques for combining sensor-level tracks into a global track file are described in many references [6, 9, 10, 17, 23]. The next sections give an overview and a comparative study of most of the reported techniques.

3.1.1 Best track

This is the simplest way to obtain a global track from the sensor-level tracks. It consists in choosing, according to some decision strategy, the best among all the available tracks. One approach consists in selecting the sensor-level track that yields the smallest (norm of the) error covariance matrix. Since this method uses only one track at time, it does not take advantage of the information contained in the other tracks, that is, the information brought by the other sensors. In this case, if one sensor is slightly more accurate than the others, the central track will reduce to the track generated by the tracker associated with that sensor. Since it does not take advantage of the available fusion architecture, the performance of the resulting fusion will correspond to that of single-sensor architecture.

3.1.2 Simple fusion

The simple fusion method, where the sensor-level tracks are assumed independent, is another simple way to generate global tracks. As a consequence of the independence assumption, a standard Kalman filter is used in the central tracker. The (assumed independent) sensor-level tracks are then simply treated as contacts, with the identity as an observation matrix. This approach, whose algorithm is given below, suffers from inconsistency.

Prediction

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{F}_k \hat{\boldsymbol{x}}_{k|k} + \boldsymbol{G}_k \boldsymbol{u}_k \tag{76}$$

$$\hat{\boldsymbol{P}}_{k+1|k} = \boldsymbol{F}_k \hat{\boldsymbol{P}}_{k|k} \boldsymbol{F}_k^T + \boldsymbol{Q} \tag{77}$$

Update

$$\hat{\boldsymbol{P}}^{-1}_{k+1|k+1} = \hat{\boldsymbol{P}}^{-1}_{k+1|k} + \hat{\boldsymbol{P}}^{-1}_{1_{(k+1)}} + \hat{\boldsymbol{P}}^{-1}_{2_{(k+1)}} \tag{78}$$

$$\hat{\boldsymbol{P}}^{-1}_{k+1|k+1} \hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{P}}^{-1}_{k+1|k} \hat{\boldsymbol{x}}_{k+1|k} + \hat{\boldsymbol{P}}^{-1}_{1_{(k+1)}} \hat{\boldsymbol{x}}_{1_{(k+1)}} + \hat{\boldsymbol{P}}^{-1}_{2_{(k+1)}} \hat{\boldsymbol{x}}_{2_{(k+1)}} \tag{79}$$

This filter overestimates its own performance, by underestimating the actual error covariance matrix. This inconsistency may lead the filter to instability, if the latter is applied for recursive estimation. This is why alternative methods are presented below.

3.1.3 Covariance intersection

The covariance intersection provides a method that allows performing the track[6] fusion without assuming independence. The "intersection" terminology is related to the geometric analogy based upon the covariance σ-contour plots.

[6] And correlated data, in general.

By using a Kalman-like fusion rule, the actual error covariance will always lie within the region defined by the intersection of the covariance of the fused sources, whatever is the degree of correlation between these sources [14].

Based on this observation, and in order to guarantee the consistency, the covariance intersection algorithm computes a covariance matrix that will always enclose the intersection region. The latter is in fact an upper bound for the actual covariance. In the case of two sensor-level tracks, $(\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{P}}_1)$ and $(\hat{\boldsymbol{x}}_2, \hat{\boldsymbol{P}}_2)$, the central fusion algorithm is given by.

Prediction

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{F}_k \hat{\boldsymbol{x}}_{k|k} + \boldsymbol{G}_k \boldsymbol{u}_k \tag{80}$$

$$\hat{\boldsymbol{P}}_{k+1|k} = \boldsymbol{F}_k \hat{\boldsymbol{P}}_{k|k} \boldsymbol{F}_k^T + \boldsymbol{Q} \tag{81}$$

Update

$$\hat{\boldsymbol{P}}^{-1}_{k+1|k+1} = \omega_0 \hat{\boldsymbol{P}}^{-1}_{k+1|k} + \omega_1 \hat{\boldsymbol{P}}^{-1}_{1_{(k+1)}} + \omega_2 \hat{\boldsymbol{P}}^{-1}_{2_{(k+1)}} \tag{82}$$

$$\hat{\boldsymbol{P}}^{-1}_{k+1|k+1} \hat{\boldsymbol{x}}_{k+1|k+1} = \omega_0 \hat{\boldsymbol{P}}^{-1}_{k+1|k} \hat{\boldsymbol{x}}_{k+1|k} + \omega_1 \hat{\boldsymbol{P}}^{-1}_{1_{(k+1)}} \hat{\boldsymbol{x}}_{1_{(k+1)}} + \omega_2 \hat{\boldsymbol{P}}^{-1}_{2_{(k+1)}} \hat{\boldsymbol{x}}_{2_{(k+1)}} \tag{83}$$

Subject to

$$\omega_0 + \omega_1 + \omega_2 = 1 \tag{84}$$

$$\omega_j \geq 0, \quad \forall j \tag{85}$$

The parameters ω_j provide additional degrees of freedom that allow one to perform a further optimization of (a some norm of) the combined covariance matrix.

Since the intersection region represents (in all directions) an upper limit for the actual covariance, and the covariance intersection method overestimates it, the latter results in a significant loose of performance. This is why the method is said sub-optimal. To avoid this large overestimation of the actual covariance matrix, we proposed in [15] a new filter called largest ellipsoid, which is briefly presented below.

3.1.4 Largest ellipsoid

As in the covariance intersection case, the design of the largest ellipsoid filter is based on the estimation of the intersection region. But instead of overestimating it, the proposed filter tries to slightly underestimate it. This will have no consequence on the consistency of the filter, because the intersection region is only an upper limit. Since the σ-contour defined by the actual covariance matrix is an ellipsoid, the largest ellipsoid filter computes

the largest ellipsoid contained within this intersection region. The matrix orientation problems may however render the computation of this intersection very difficult, and even impossible. This is why the following geometrical transformations are first applied to the covariance matrices $\hat{P}_{1_{(k+1)}}$ and $\hat{P}_{2_{(k+1)}}$.

$$T = \begin{bmatrix} 1 & 0 & .. & 0 \\ 0 & \sqrt{\frac{\lambda_{1_1}}{\lambda_{1_2}}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sqrt{\frac{\lambda_{1_1}}{\lambda_{1_n}}} \end{bmatrix} \begin{bmatrix} v_{1_1}^T \\ \vdots \\ v_{1_n}^T \end{bmatrix}$$

where λ_{1_j} and v_{1_j} are respectively the j^{th} eigenvalue and eigenvector of $\hat{P}_{1_{(k+1)}}$. The shape and the size of the intersection ellipsoid are then defined by the following matrix

$$\hat{E} = T^{-1} \begin{bmatrix} v_{2_1} & \dots & v_{2_n} \end{bmatrix} D_{min} \begin{bmatrix} v_{2_1}^T \\ \vdots \\ v_{2_n}^T \end{bmatrix} T^{-T} \tag{86}$$

which will serve as the error covariance matrix $\hat{P}_{k+1|k+1}$ for the central track. The matrix D_{min} is defined by

$$D_{min} = \begin{bmatrix} \min(\lambda_{1_1}, \lambda_{2_1}) & 0 & \dots & 0 \\ 0 & \min(\lambda_{1_1}, \lambda_{2_2}) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \min(\lambda_{1_1}, \lambda_{2_n}) \end{bmatrix} \tag{87}$$

where λ_{2_j} and v_{2_j} are respectively the j^{th} eigenvalue and eigenvector of the transformed matrix

$$P_2^{\dagger} = T \hat{P}_{2_{(k+1)}} T^T \tag{88}$$

Besides the computation of the largest ellipsoid matrix, a simple fusion algorithm is run. The resulting state vector estimate (79) is kept, while the covariance matrix (78) is dropped and replaced by (86). The pair (79)-(86) represents the updated central track, and will serve as the starting point for the next cycle in both of the simple fusion algorithm and the largest ellipsoid computation.

3.1.5 Weighted covariance

An estimate of the cross covariance matrix is computed recursively. This allows the correlation between the tracks to be taken into account, through the application of the general form of the Kalman filter given by the equations

(13)-(15). Since the Kalman gains W_1 and W_2 are computed at the contact arrival rate, the cross covariance matrix has to be computed at this same rate. An example of a recursive computation [16, 17], and the associated fusion rule, are given below.

Correlation

$$P_j^c = \left[I - W_{1_j} H_{1_j}\right] P_{j|j-1}^c \left[I - W_{2_j} H_{2_j}\right] \tag{89}$$

Update

$$P_E = \hat{P}_{1_{(k+1)}} - P_{k+1}^c + \hat{P}_{2_{(k+1)}} - (P_{k+1}^c)^T \tag{90}$$

$$W_{k+1} = \left[\hat{P}_{1_{(k+1)}} - P_{k+1}^c\right] P_E^{-1} \tag{91}$$

$$\hat{x}_{k+1} = \hat{x}_{1_{(k+1)}} + W_{k+1}\left[\hat{x}_{2_{(k+1)}} - \hat{x}_{1_{(k+1)}}\right] \tag{92}$$

$$\hat{P}_{(k+1)} = \hat{P}_{1_{(k+1)}} - \left[\hat{P}_{1_{(k+1)}} - P_{k+1}^c\right] P_E^{-1} \left[\hat{P}_{1_{(k+1)}} - (P_{k+1}^c)^T\right] \tag{93}$$

Since it requires the Kalman gains used to generate the sensor-level tracks, this method cannot use the prior information (predicted from the central track), and serves only to fuse synchronous sensor tracks. The fusion result is therefore not optimal in the minimum mean square error sense, but only in the maximum likelihood sense. Furthermore, if the sensor trackers' Kalman gains are not available, or cannot be sent the central tracker at the sensor trackers' update rate, the weighted covariance algorithm cannot be implemented.

3.1.6 Tracklet fusion

This method is based on the removal of the redundant information approach. In the case of a fusion without the process noise, this method is proven to be optimal, that is, it is equivalent to the contact-level fusion. Nevertheless, in the practice the presence of this noise does not affect much the performance of the method. Since the sensor-level trackers compute tracklets, and not complete tracks, the only available track is located at the central node[7]. The sensor-level tracks may be required in some situations, as for the track association problems in multi-target environments. To solve this problem, sensor-level trackers will have to maintain complete tracks besides the tracklets. This obviously will results in an increased computation load and communication bandwidth[8].

[7] If a feedback is used, the central node will also have to run a tracklet algorithm, which means that even the central track will no longer be available, unless additional computation is performed to also maintain a complete track.

[8] Since both of the tracklets and the tracks are sent to the central tracker.

The method is illustrated here with the example of two (sensor-level) trackers that send their tracks to a central tracker (see Figure 1). For more than two nodes, each pair of nodes that exchange information will have to run a similar algorithm to avoid redundancy. It is clear that, for a large number of nodes, the implementation may be difficult, and even impossible; if the data transfer paths are not known. This is why this method is only practical in the case of "small" networks with a known data flow.

3.1.6.1 *Tracklet computation*

For each sensor-level tracker, a tracklet is computed at the end of cycles corresponding to α contacts. The starting time and the end of each cycle are noted, respectively, l and k. Take note that

$$k = l + \alpha \tag{94}$$

where α can be seen as an over-sampling rate. Since the same treatment applies to both nodes, and to lighten the notation, the subscript corresponding to the node number will be dropped. Starting from the last sent tracklet at time instant l, *viz.* $(\boldsymbol{u}_l, \boldsymbol{U}_l)$, and under the zero process noise assumption, the predicted value (α steps ahead) of the state vector, without using the contacts, is given by

$$\hat{\boldsymbol{x}}_{k|l} = \boldsymbol{\Phi}_{l|k}\boldsymbol{u}_l \tag{95}$$

$$\hat{\boldsymbol{P}}_{k|l} = \boldsymbol{\Phi}_{l|k}\boldsymbol{U}_l\boldsymbol{\Phi}_{l|k}^T \tag{96}$$

where

$$\boldsymbol{\Phi}_{l|k} = \prod_{j=l}^{k-1} \boldsymbol{F}_j \tag{97}$$

During this same cycle, the filtered state that does use the contacts is given by

$$\hat{\boldsymbol{x}}_{k|k} = \boldsymbol{F}_{k-1}\hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{W}_k\left[\boldsymbol{z}_k - \boldsymbol{H}_k\boldsymbol{F}_{k-1}\hat{\boldsymbol{x}}_{k-1|k-1}\right] \tag{98}$$

$$= \left[\boldsymbol{I} - \boldsymbol{W}_k\boldsymbol{H}_k\right]\boldsymbol{F}_{k-1}\hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{K}_k\boldsymbol{z}_k \tag{99}$$

$$= \boldsymbol{\Gamma}_{l|k}\boldsymbol{u}_l + \boldsymbol{\varphi}(\boldsymbol{z}) \tag{100}$$

and

$$\hat{\boldsymbol{P}}_{k|k} = \left[\boldsymbol{I} - \boldsymbol{W}_k\boldsymbol{H}_k\right]\boldsymbol{F}_{k-1}\hat{\boldsymbol{P}}_{k-1|k-1}\boldsymbol{F}_{k-1}^T \tag{101}$$

$$= \boldsymbol{\Gamma}_{l|k}\boldsymbol{U}_l\boldsymbol{\Phi}_{l|k}^T \tag{102}$$

where

$$\Gamma_{l|k} = \prod_{j=l}^{k-1} \left[I - W_{j+1} H_{j+1} \right] F_j \tag{103}$$

The correlation between $\hat{x}_{k|k}$ and $\hat{x}_{k|l}$ is computed to allow for the removal of the redundant information. This correlation is given by

$$E\left[\hat{x}_{k|k}\hat{x}_{k|l}^T\right] = E\left[\hat{x}_{k|k}u_l^T \Phi_{l|k}^T\right] \tag{104}$$

$$= E\left[\hat{x}_k u_l^T\right] \Phi_{l|k}^T \tag{105}$$

$$= \Gamma_{l|k} U_l \Phi_{l|k}^T \tag{106}$$

$$= \hat{P}_{k|k} \tag{107}$$

The idea behind the tracklet fusion algorithm consists in generating a piece of information (u_k, U_k), called a tracklet that, unlike the complete track, is rendered independent of (u_l, U_l). The independence is obtained by combining the above-calculated predicted and filtered vectors as follows

$$u_k = \hat{x}_{k|l} + \kappa(\hat{x}_{k|k} - \hat{x}_{k|l}) \tag{108}$$

where the gain κ is chosen such that

$$E\left[u_k u_l^T\right] = 0 \tag{109}$$

From (95), this is equivalent to

$$E\left[u_k \hat{x}_{k|l}^T\right] = \hat{P}_{k|l} + \kappa\left[\hat{P}_{k|k} - \hat{P}_{k|l}\right] \tag{110}$$

$$= 0$$

An obvious choice for κ that guarantees a zero cross-correlation is given by

$$\kappa = \hat{P}_{k|l}\left[\hat{P}_{k|l} - \hat{P}_{k|k}\right]^{-1} \tag{111}$$

The resulting tracklet and its covariance matrix are then given by

$$u_k = \hat{x}_{k|l} + \hat{P}_{k|l}\left[\hat{P}_{k|l} - \hat{P}_{k|k}\right]^{-1}(\hat{x}_{k|k} - \hat{x}_{k|l}) \tag{112}$$

$$U_k = \hat{P}_{k|l}\left[\hat{P}_{k|l} - \hat{P}_k\right]^{-1}\hat{P}_{k|l} - \hat{P}_{k|l} \tag{113}$$

Because of their complete independence, the tracklets from the both sensor-level trackers are treated as contacts by the central node, with an identity observation matrix. The tracklet algorithm, whose principle is illustrated by Figure 2, can also be rewritten in the information space. This leads to the following algorithm.

Computation $(i = 1, 2)$

$$\Phi_{l|k} = \prod_{j=l}^{k-1} F_j \tag{114}$$

$$U_{i_k}^{-1} = \hat{P}_{i_{k|k}}^{-1} - \Phi_{l|k}^{-T} U_{i_l}^{-1} \Phi_{l|k}^{-1} \tag{115}$$

$$U_{i_k}^{-1} u_{i_k} = \hat{P}_{i_{k|k}}^{-1} \hat{x}_{i_{k|k}} - \Phi_{l|k}^{-T} U_{i_l}^{-1} u_{i_l} \tag{116}$$

Fusion

$$\hat{P}_{k|k}^{-1} = \Phi_{l|k}^{-T} \hat{P}_{l|l}^{-1} \Phi_{l|k}^{-1} + U_{1_k}^{-1} + U_{2_k}^{-1} \tag{117}$$

$$\hat{P}_{k|k}^{-1} \hat{x}_{k|k} = \Phi_{l|k}^{-T} \hat{P}_{l|l}^{-1} \hat{x}_{l|l} + U_{1_k}^{-1} u_{1_k} + U_{2_k}^{-1} u_{2_k} \tag{118}$$

Re-initialization

$$\hat{P}_{i_{l|l}} = U_{i_k} \tag{119}$$

$$\hat{x}_{i_{l|l}} = u_{i_k} \tag{120}$$

3.1.7 Information filter

As shown in Figure 3, the information filter method is very similar to the tracklet fusion approach (Figure 2). The information filter is however more general, since it does not assume the process noise to be null. This approach yields an optimal fusion of tracks and, in terms of performance, is equivalent to the contact-level fusion. Besides the process noise assumption, the fusion algorithm given below differs from the tracklet fusion in the re-initialization mechanism at the starting of each cycle. While the tracklet fusion re-initializes both of the predicted and the update estimates, the information filter re-initializes only the predicted one. Another difference lies in the information (u_l, U_l) used for this re-initialization. The tracklet uses the previously sent de-correlated tracklet, while the information filter re-initializes with the complete sensor-level track.

Besides its superiority over the tracklet fusion in terms of performance, the information filter has the advantage of maintaining complete tracks at the sensor-level, instead of tracklets that contain only partial information. Note that the information removal can be performed at either the sensor-level

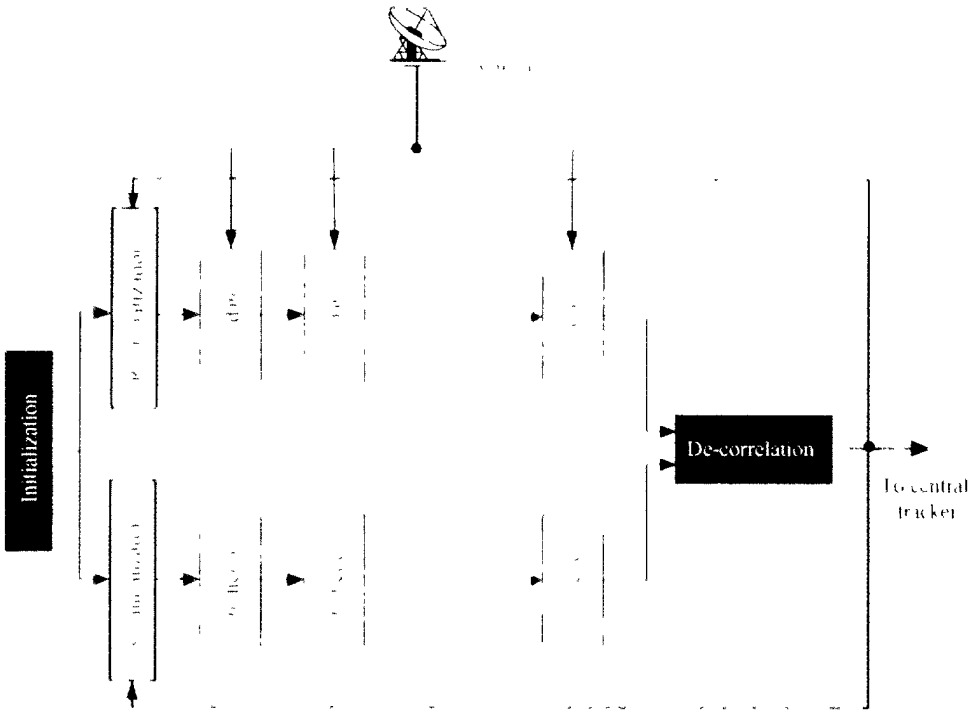

Figure 2: *Tracklet fusion algorithm*

trackers or the central tracker. If the removal is ignored, the method reduces to the simple fusion.

Redundancy removal ($i = 1, 2$)

$$\boldsymbol{U}_{i_k}^{-1} = \hat{\boldsymbol{P}}_{i_k}^{-1} - \hat{\boldsymbol{P}}_{i_{k|l}}^{-1} \tag{121}$$

$$\boldsymbol{U}_{i_k}^{-1}\boldsymbol{u}_{i_k} = \hat{\boldsymbol{P}}_{i_k}^{-1}\hat{\boldsymbol{x}}_{i_k} - \hat{\boldsymbol{P}}_{i_{k|l}}^{-1}\hat{\boldsymbol{x}}_{i_{k|l}} \tag{122}$$

Information fusion

$$\hat{\boldsymbol{P}}_k^{-1} = \hat{\boldsymbol{P}}_{k|l}^{-1} + \boldsymbol{U}_{1_k}^{-1} + \boldsymbol{U}_{2_k}^{-1} \tag{123}$$

$$\hat{\boldsymbol{P}}_k^{-1}\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{P}}_{k|l}^{-1}\hat{\boldsymbol{x}}_{k|l} + \boldsymbol{U}_{1_k}^{-1}\boldsymbol{u}_{1_k} + \boldsymbol{U}_{2_k}^{-1}\boldsymbol{u}_{2_k} \tag{124}$$

where

$$\hat{\boldsymbol{P}}_{j|j-1} = \boldsymbol{F}_j\hat{\boldsymbol{P}}_{j-1}\boldsymbol{F}_j^T + \boldsymbol{Q} \tag{125}$$

$$\hat{\boldsymbol{x}}_{j|j-1} = \boldsymbol{F}_j\hat{\boldsymbol{x}}_{j-1} \tag{126}$$

3.2 Benefits of track-level fusion

The track-level (or decentralized) fusion has the advantage of reduced communication requirements when compared with the centralized contact-level scheme [6, 8, 17]. The track-level fusion requires much less I/O bandwidth to transmit the data between the sensors and the central node. The local tracks are periodically transferred to the central processor rather than the copious measurement data. If necessary, the sensor-level tracks may also be communicated less frequently than the arrival of the sensor data (*e.g.*, the α over-sampling rate for the tracklet fusion and the information filter approaches). The approach also has the advantage of reduced computational loading (in any single processor). Certain computational advantages may result from the inherent parallel processing possible using this approach [6, 17].

In military applications, due to the distributed tracking capabilities, the decentralized fusion results in increased survivability when compared with centralized systems [5, 23]. Moreover, if one sensor becomes degraded, its observations will not affect the sensor-level tracks of the other sensors (*i.e.*, the good sensor-level tracks will not be corrupted by the bad data) [5, 23]. By checking the sensor-level tracks with the central tracks, one may be able to detect any errors in the sensors [23]. Then, when the sensor with poor data is finally recognized, the central-level tracks can be formed using only sensor-level tracks from non-degraded sensors [5]. The track-level fusion option allows also the fusion system designer to tune each sensor-specific estimation process to the nuances of that particular sensor's data and operating characteristics [4, 5, 23]. Data association for this architecture is relatively simple, since the process compares state vectors or identity declarations instead of raw data [9]. The approach also allows data from non-commensurate sensors to be combined. Finally, the track-level fusion

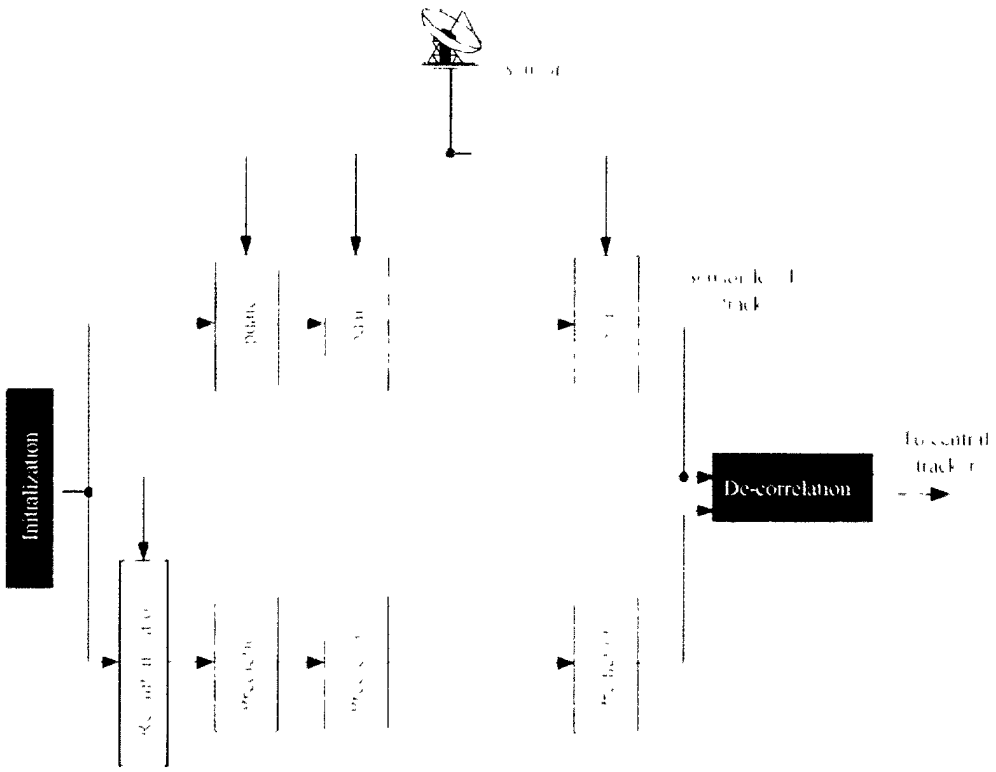

Figure 3: *Information filter algorithm*

architecture offers a systematic and natural upgrade path from the existing surveillance systems. This is achieved by using previously developed single-sensor trackers locally, followed by the combination of their local tracks to form global tracks.

3.3 Drawbacks of track-level fusion

A major disadvantage of the track-level fusion is the inherent poor reaction time. A target track is reported to the Commanding Officer / Operator only after one of the single sensors has promoted its internal track to the firm status. Moreover, the track-level fusion potentially provides a significant information loss compared with the contact-level fusion, since sensor data are represented via state vectors or identity declarations. The information processing and analysis for each sensor may result in a local optimization rather than a global optimized solution [9, 10].

Less accurate tracking and correlation are to be expected if sensor-level tracks are maintained. For example, there will be a higher probability of false correlation for the sensor-level tracks in areas affected by clutter because the gate sizes will be larger due to less frequent track updates [23].

The major difficulty associated with the use of the decentralized data fusion arises during the formation of global tracks from sensor-level tracks [5]. There is a requirement to correlate and fuse track data, such as state vectors, which are not statistically independent due to potential target maneuvers (representing common error sources). It is necessary in the track processing algorithms to account explicitly for such dependence [9]. If track fusion occurs without taking into consideration the statistical interdependence, then there will be a reduction in the random error but the mean error due to target maneuver will not be averaged out [5]. However, these issues are of importance only when the tracks being combined are of comparable accuracy. When considering radar, IR, and ESM sensors the track fusion process essentially will be a merging of IR angle, radar range, and ESM ID. The effect of the common error source of target maneuver will not be a major issue for this set of sensors. Finally, the track-level fusion architecture can suffer from a poor false track suppression capability since poor track-to-track correlation (*e.g.*, for a maneuvering target) will inevitably lead to multiple tracks on the same target.

4. Performance comparison of various trackers

The methods presented are illustrated and compared using the target's tracking example, whose fusion network is shown in Figure 1. One is interested here in providing an "improved" estimate of the state vector, such that the target's kinematics can be depicted as correctly as possible. This tracking problem presents both of the independent and correlated source fusion problems. But since the independent source case admits the standard Kalman filter as an optimal solution, the emphasis will be put on the more challenging correlated case, *i.e.*, the track-level fusion.

The tracked target is assumed to be moving in a 2D space, where the acceleration acts as an input[9]. The state to be estimated is therefore composed of the target's coordinates, *viz.*, the position and the linear velocity. With the following state variable notation

$$x = \begin{bmatrix} p_x \\ p_y \\ v_x \\ v_y \end{bmatrix} \quad \text{and} \quad v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{127}$$

the equations of such a target can be expressed as

$$x_{k+1} = F_k x_k + \Gamma v_k \tag{128}$$

where v is vector of random variables that reflects the unforeseeable variation of the acceleration in both directions and the state transition matrix is given by

$$F_k = \begin{bmatrix} 1 & 0 & h & 0 \\ 0 & 1 & 0 & h \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{129}$$

h is the time increment. The matrix Γ in (128) depends on the model used to represent the discrete-time nature of the process noise v. Two examples are given below. The first one

$$\Gamma = \begin{bmatrix} h^2/2 & 0 \\ 0 & h^2/2 \\ h & 0 \\ 0 & h \end{bmatrix} \tag{130}$$

[9]Since the acceleration may change unforeseeably, it is often modeled as a random variable, and so will be as such.

reprensents to the pulse model, while the following one

$$\Gamma = \begin{bmatrix} \frac{\sqrt[3]{h^2}}{2\sqrt{3}} & \frac{\sqrt[3]{h^2}}{2} & 0 & 0 \\ 0 & 0 & \frac{\sqrt[3]{h^2}}{2\sqrt{3}} & \frac{\sqrt[3]{h^2}}{2} \\ 0 & \sqrt{h} & 0 & 0 \\ 0 & 0 & 0 & \sqrt{h} \end{bmatrix} \tag{131}$$

is used by the Brownian motion modelling. Note that the latter requires four random variables instead of two. If the process noises are assumed to have the same standard deviation δ_v, the pulse model results in the following process noise covariance matrix

$$Q = \delta_v^2 \Gamma\Gamma^T \tag{132}$$

$$= \delta_v^2 \begin{bmatrix} \frac{h^4}{4} & 0 & \frac{h^3}{2} & 0 \\ 0 & \frac{h^4}{4} & 0 & \frac{h^3}{2} \\ \frac{h^3}{2} & 0 & h^2 & 0 \\ 0 & \frac{h^3}{2} & 0 & h^2 \end{bmatrix} \tag{133}$$

while the covariance matrix for the brownian motion is given by

$$Q = \delta_v^2 \Gamma\Gamma^T \tag{134}$$

$$= \delta_v^2 \begin{bmatrix} \frac{h^3}{3} & 0 & \frac{h^2}{2} & 0 \\ 0 & \frac{h^3}{3} & 0 & \frac{h^2}{2} \\ \frac{h^2}{2} & 0 & h & 0 \\ 0 & \frac{h^2}{2} & 0 & h \end{bmatrix} \tag{135}$$

4.1 Sensor-level tracker

The contact fusion problem (at the sensor level) is first considered. This corresponds to the case of actually independent sources, since the measurement and the estimation noise at the sensor level are not correlated. The target's position (in both directions) is the only measured variable. This can be represented by the following observation equation

$$\boldsymbol{z}_{i_{k+1}} = \boldsymbol{H}_i \boldsymbol{x}_{k+1} \tag{136}$$

where $k+1$ is the observation time, i is the sensor number $(= 1, 2)$ and $\boldsymbol{H}_i$ is the observation matrix that is given, for both sensors, by

$$\boldsymbol{H}_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{137}$$

4.2 Central tracker

The central tracker receives and fuses the tracks from the two sensor-level trackers, which, for simplicity, are assumed time aligned. Since it deals with correlated sources, the track-level fusion problem is more challenging than the contact-level one. Different methods are compared, and in order to lighten the graphics, each method is separately plotted versus the centralized contact-level fusion approach. Due to its proven optimality, the latter is taken as a reference, in the performance evaluation. Nevertheless, the performance is not the unique comparison criterion. The consistency is another important property, which helps in choosing the appropriate approach for a given situation.

4.3 Results

To compare the different algorithms, the traces of the actual and the estimated error covariance matrix of the central tracker are plotted. All the plotted traces are normalized with respect to the trace of the estimated error covariance matrix yielded by the centralized contact-level fusion. The results presented are obtain via 200 Monte-Carlo runs. The used covariance matrix corresponds to the pulse model, with $\delta_v = 12$ and $h = .1$. The sensor noise standard deviations are given, respectively, by $\delta_{v_x} = 5$ and $\delta_{v_y} = 4$ for the first sensor, and $\delta_{v_x} = 6$ and $\delta_{v_y} = 3$, for the second one. Note that the sensors are not assumed identical, and the positions in each direction (x or y) are measured with a different accuracy. The over-sampling rate α has been fixed to 5.

Furthermore, to obtain a comparison means between the presented methods, two parameters are defined (see Table 1). The first one ϱ_c corresponds, for each considered method, to the ratio of the trace of the estimated covariance matrix to the trace of the actual one. This gives a direct measure of the consistency of the fusion algorithms. Note that a method is inconsistent, if $\varrho_c \leq 1$. The second parameter ϱ_p, which is ≥ 1, deals with the performance. It is defined by the ratio of the trace of the actual covariance matrix (yielded by each method) to the actual covariance matrix yielded by the centralized contact fusion. Note that the more ϱ_p is close to 1, the more the corresponding method is performant.

4.3.1 Best track

The best track approach consists in choosing, between the two tracks, the one that yields the highest performance, according to some criterion. But since the information contained in the other one is dropped, this approach results, as shown in Figure 4, in a loose of performance ($\varrho_p = 1.32$). The best track method has however the advantage of always leading to a consistent estimate ($\varrho_c = 1.02$). The consistency is guaranteed because the central track is always equal to one of the two sensor-level tracks, which are both consistent.

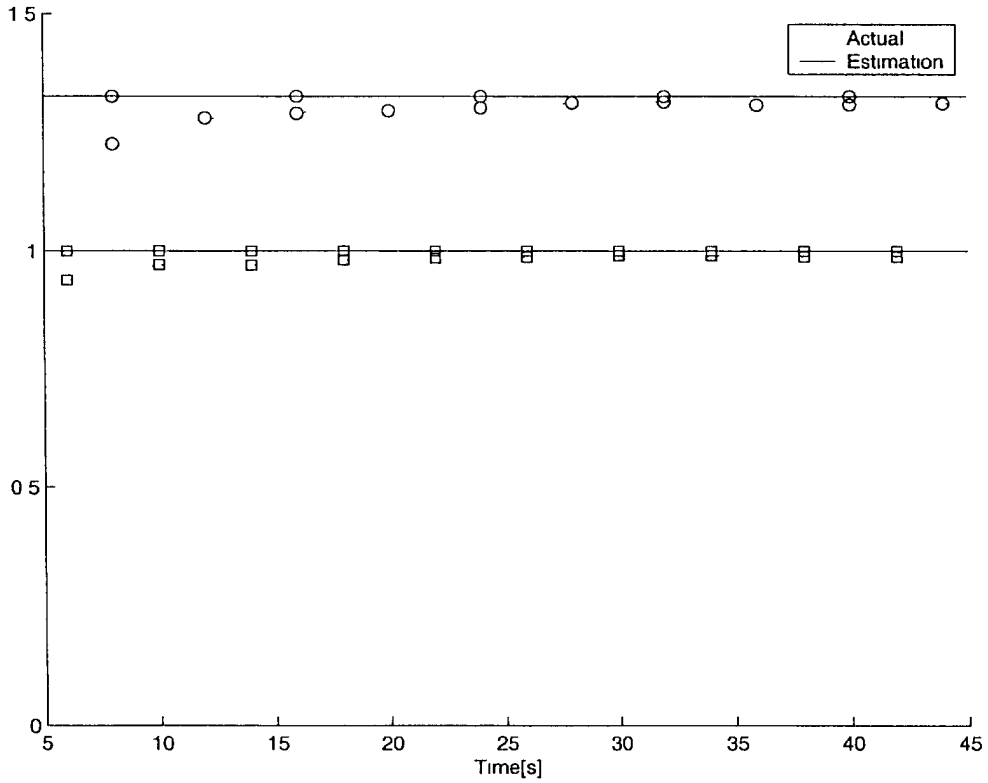

Figure 4: *Best track fusion (◯) vs. Contact fusion (□)*

4.3.2 Simple fusion

The simple fusion assumes the independence between the sensor-level tracks, and between each of the sensor-level tracks and the central track. It therefore uses a standard Kalman filter to fuse them, which obviously leads to inconsistency. This result is clearly shown in Figure 5, where the fusion algorithm underestimates the actual error covariance matrix ($\varrho_c = 0.35$).

4.3.3 Covariance intersection

As shown in Figure 6, the covariance intersection allows avoiding the inconsistency of the simple fusion. The actual covariance matrix is indeed smaller than the assumed one ($\varrho_c = 1.12$). This consistency is however obtained at the expense of a significant loose of performance ($\varrho_p = 1.20$).

4.3.4 Largest ellipsoid

The results given in Figures 7 & 8 are conclusive, since, while the estimate is still consistent ($\varrho_c = 1.10$), the assumed covariance, is much closer the centralized contact-level fusion one, than that obtained by the covariance intersection. Indeed, the actual error is only 10% larger than the one obtained with the centralized contact-level fusion, while it is 20% in the case of the covariance intersection. The performance ratio is given by $\varrho_p = 1.10$.

Even if they both result in a loose of performance, both of the covariance intersection and the largest ellipsoid have the advantage of being independent of the network fusion architecture. This makes them more suitable in decentralized fusion problems, which deal with arbitrary topologies. This is not the case of the three remaining methods, which are less general. Nevertheless, when they can be implemented, these methods guarantee better performance.

4.3.5 Weighted covariance

In the case of Figure 9, the estimate yielded by the weighted covariance is not consistent, since the filter underestimates the actual error ($\varrho_c = 0.88$). This can be attributed to that the obtained estimated cross covariance is smaller than the actual one. The increase resulting from its use in the Kalman filter is therefore not sufficient to ensure consistency.

4.3.6 Tracklet fusion & information filter

From either the consistency or performance viewpoint, the tracklet fusion (Figure 10) and the information filter (Figure 11), represent the most interesting solutions (see Table 1). The tracklet fusion presents however no

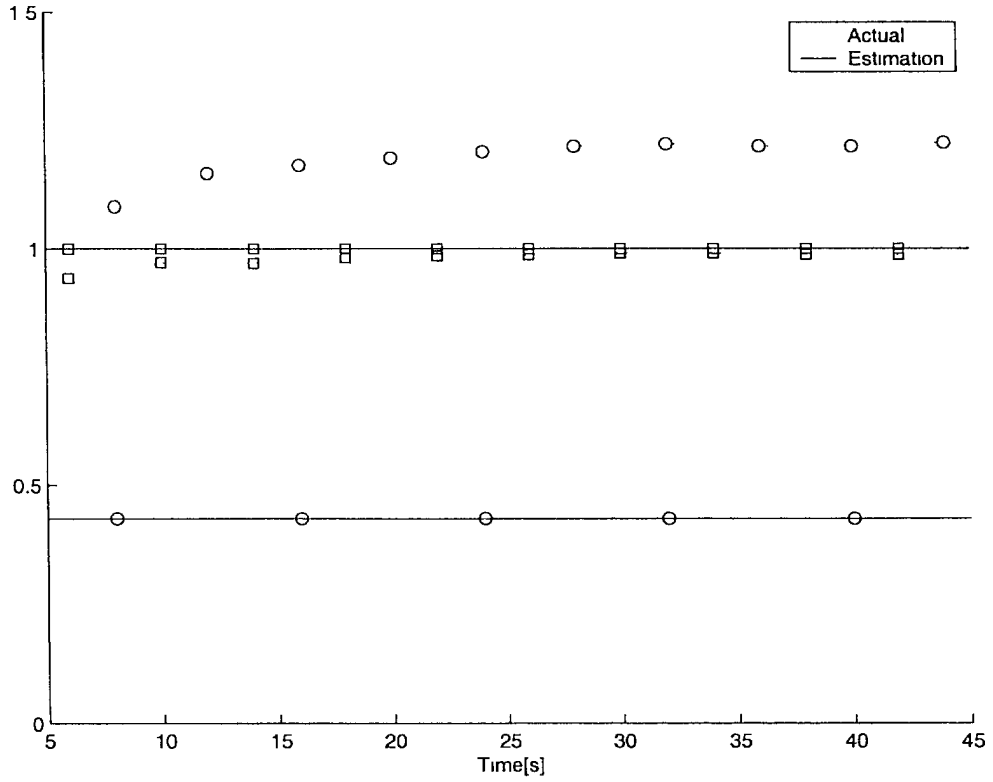

Figure 5: *Simple fusion (○) vs. Contact fusion (□)*

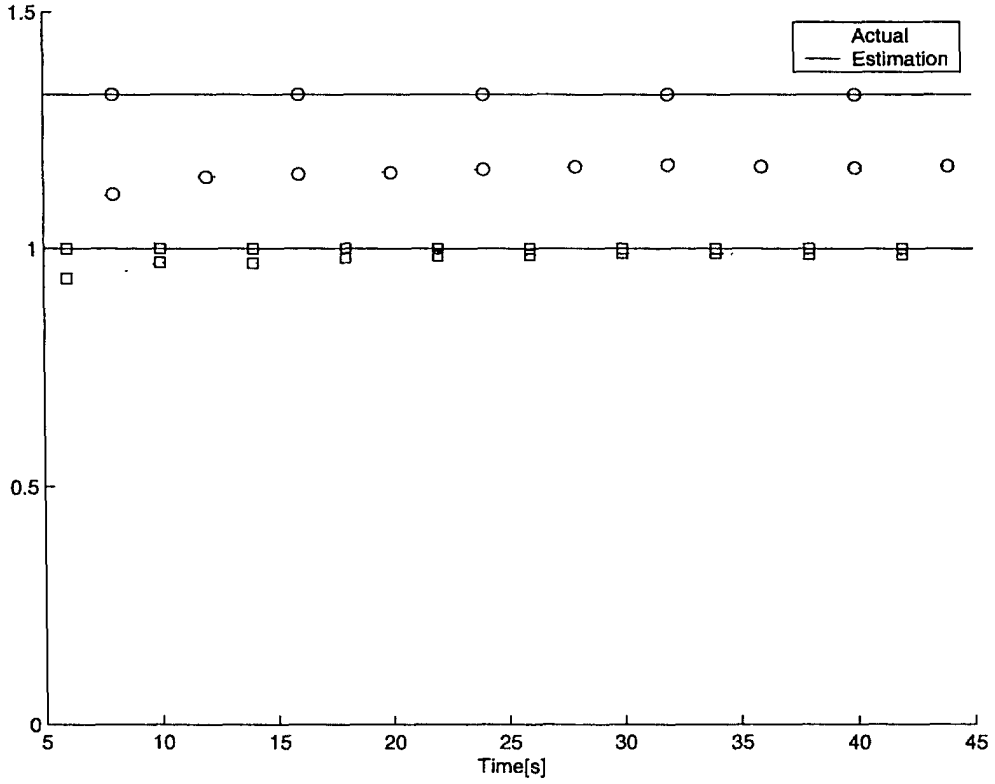

Figure 6: *Covariance intersection fusion (◯) vs. Contact fusion (□)*

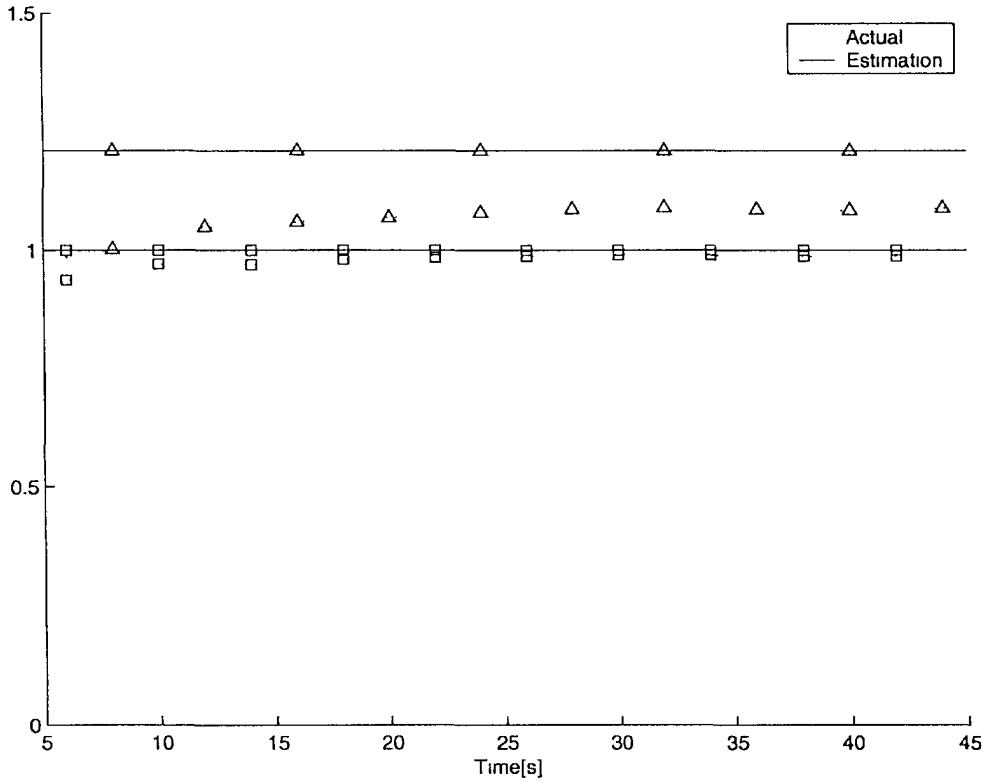

Figure 7: *Largest ellipsoid fusion (△) vs Contact fusion (□)*

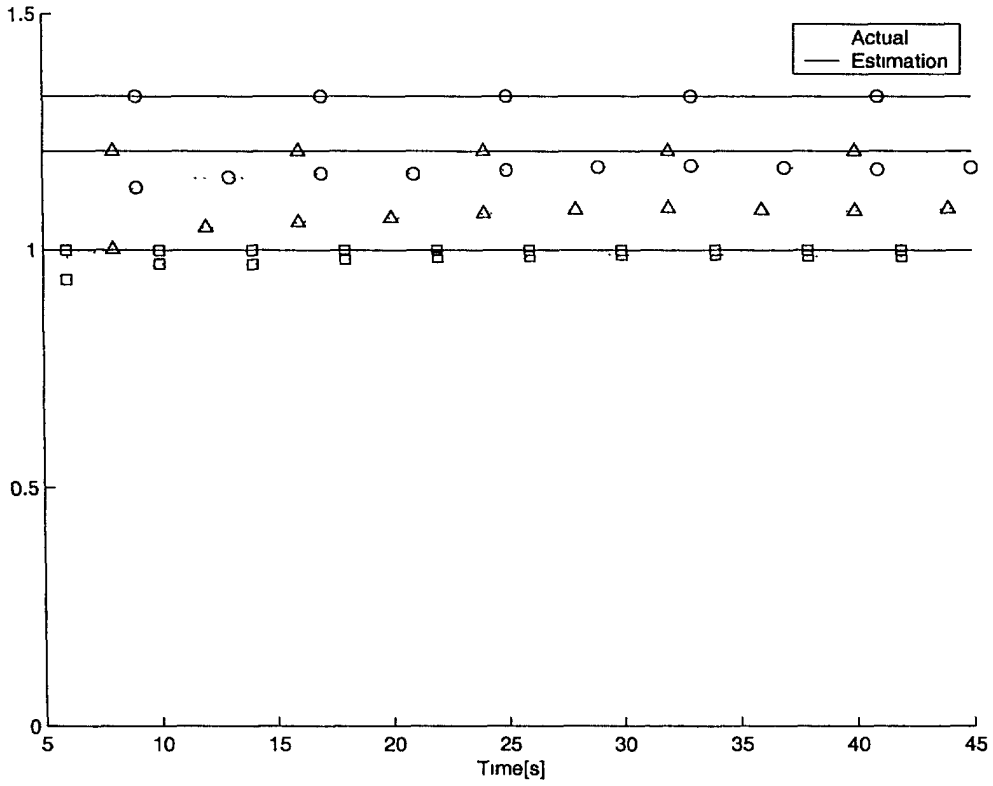

Figure 8: *Largest ellipsoid fusion (△) vs Covariance Intersection (◯) and Contact fusion (□)*

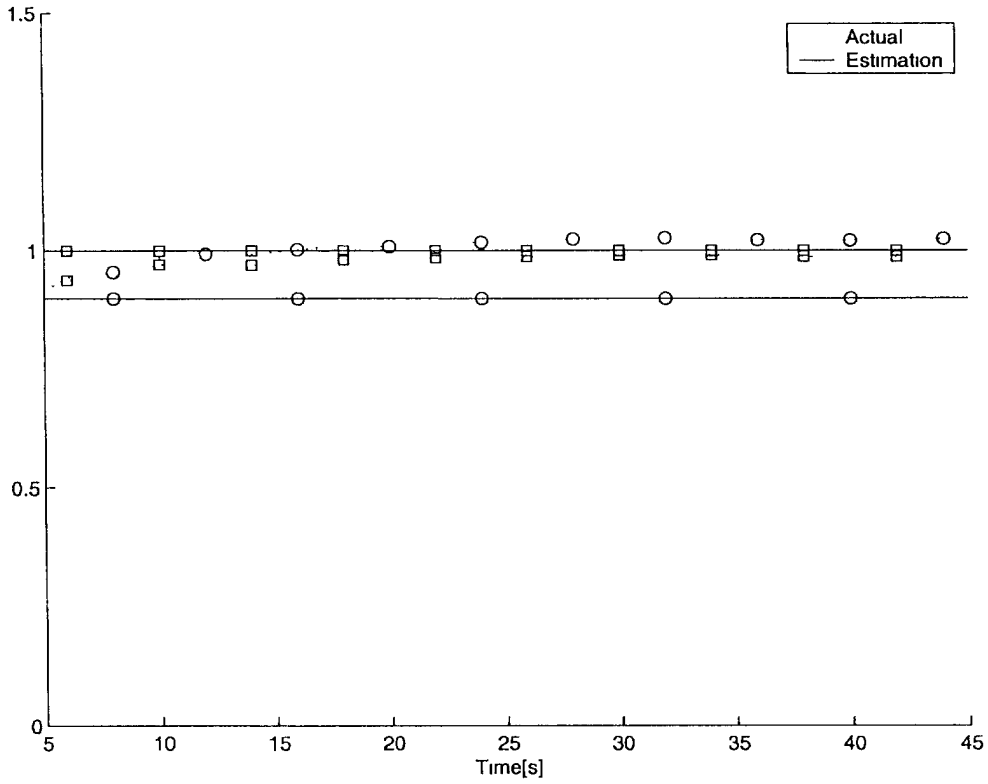

Figure 9: *Weighted covariance fusion (◯) vs. Contact fusion (□)*

real advantage over the information filter. They both have the same computation/communication requirements, while the information filter is

- **More accurate** : *i.e.*, very close to the centralized contact-level fusion.
- **Easier to implement** : does not require the modification of the existing local trackers for its implementation.
- **Maintains full tracks** : instead of tracklets. Full tracks can be used in the other fusion operations, as the track association, which is not the case of the tracklets.

4.4 Comparison

The comparison results are summarized in Table 1. It is clear that, when the network topology allows for its implementation[10], the information filter is the method to be considered, since it guarantees both the consistency and the minimum mean square error optimality. When the information filter cannot be implemented, neither of the less efficient weighted covariance and tracklet fusion can. These methods cannot therefore be of any help. In this case, the only remaining alternatives are the covariance intersection and the largest ellipsoid approaches. Since the latter is more efficient, it represents a good alternative that does not suffer from the applicability limitation.

[10]That is, is the case of small networks, with a known information flow.

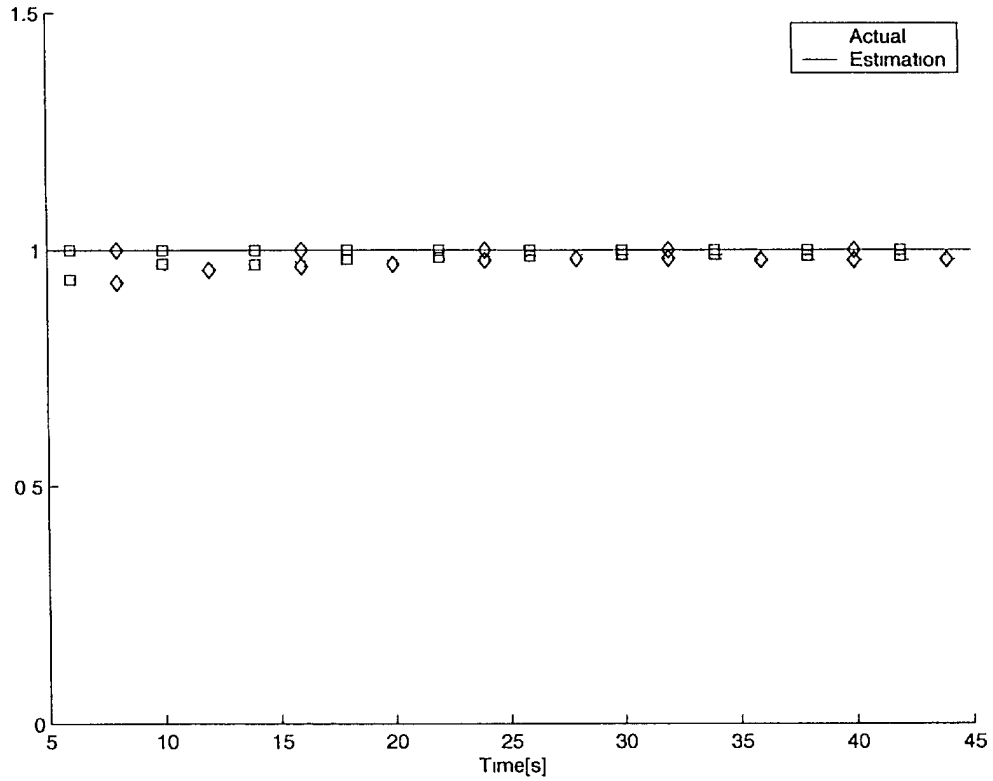

Figure 10: *Tracklet fusion (◊) vs Contact fusion (□)*

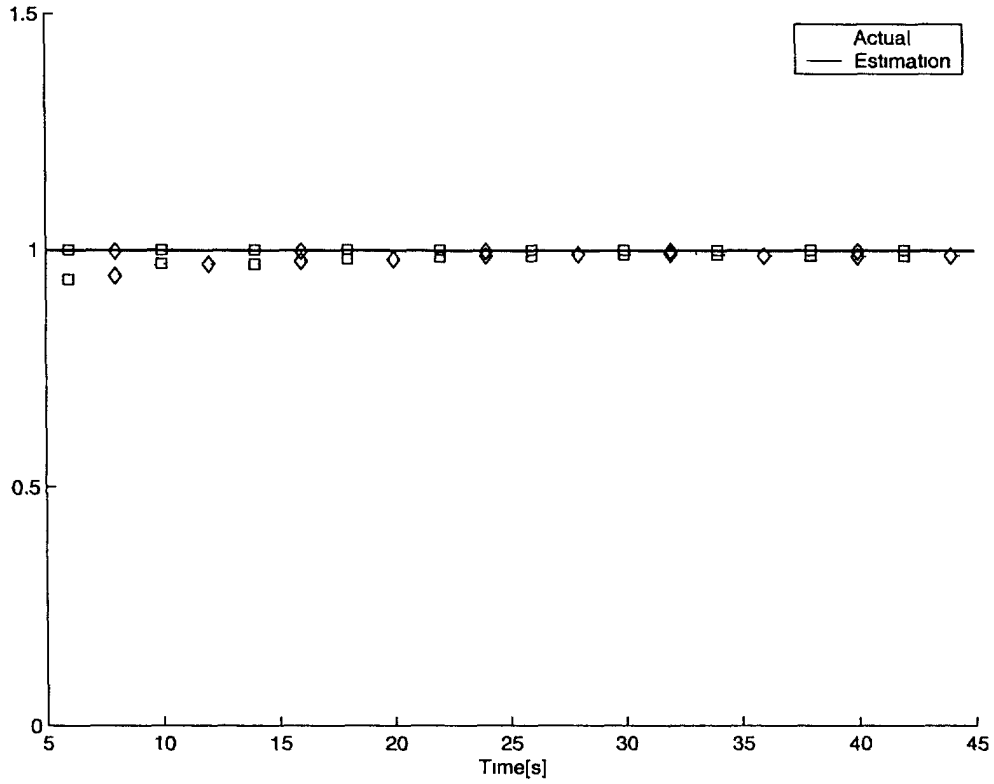

Figure 11: *Information filter fusion (◊) vs. Contact fusion (□)*

Method	ϱ_p	ϱ_c
Contact fusion	1.00	1.01
Best track	1.32	1.02
Simple fusion	1.25	0.35
Covariance intersection	1.20	1.12
Largest ellipsoid	1.10	1.10
Weighted covariance	1.01	0.88
Information filter	1.00	1.02
Tracklet fusion	1.01	1.02

Table 1: *Performance/Consistency comparison of the different track fusion methods*

5. Conclusion

Various trade-offs are generally required for the selection of the data fusion architecture, since each approach has benefits and disadvantages. As discussed in the previous chapters, this design choice affects the quality of the fused product and the nature of the algorithms or techniques that may be used. Hence, besides its high vulnerability, due to its centralized nature, contact-level fusion presents the disadvantage of being very demanding in terms of communication bandwidth and central computation capacity. This is why the track-level fusion may represent a good alternative, and the only possible solution, if the sensor contacts are not available at all. Nevertheless, the correlation between the sensor-level tracks, and between each sensor-level track and the central track, is at the origin of the inconsistency problem. This results in an underestimation of the actual error covariance matrix, if a standard Kalman filter is used for the track-level fusion. Different approaches are presented to handle this inconsistency, and result all in an increased estimated error covariance matrix. The amount of this increase, and the way it is computed, differs, however, from one method to another, leading to a different performance and conditions of applicability.

Due to its minimum mean square error optimality and its consistency, the information filter is the method to be considered, when the fusion architecture allows for its implementation. This filter guarantees the consistency by avoiding double counting of information in the fusion process. The information filter is applicable where the tracklet algorithm is, and is more efficient. The weighted covariance is not optimal in the minimum mean square error sense, but only in the maximum likelihood sense. Furthermore, its implementation is not always possible. It does not therefore present any advantage over the two previous ones. When the applicability is the major concern, the covariance intersection offers a robust alternative to the simple fusion. Nevertheless, to avoid underestimating the actual error covariance matrix, this method overestimates it too much. This results in a significant loose in performance. A good alternative to the covariance intersection is given by the largest ellipsoid method. The latter inherits the advantage of the covariance intersection, that is the large applicability and consistency, and avoids its drawback, *viz.*, the lack of performance. As the covariance intersection, this method is also based on the estimation of the intersection region, but provides a tighter estimate.

References (U)

1. Roy, J.M.J., Bossé, É., and Des Groseilliers, L. (1995). State of the Art in Local Area Sensor Data Fusion for Naval Command and Control Afloat. DREV TR 9410. UNCLASSIFIED.

2. Roy, J.M.J., Bossé, É., and Dion, D. (1995). CASE_ATTI: An Algorithm-Level Testbed for Multi-Sensor Data Fusion. DREV TR 9411. UNCLASSIFIED.

3. Roy, J., Duclos-Hindié, N., and Bossé, É (1999). An Algorithm-Level Test Bed For Level-One Data Fusion Studies. In *Proceedings of International Conference on Data Fusion - EuroFusion 99*, Stratford-upon-Avon, UK.

4. Waltz, E. and Llinas, J. (1990). Multi-sensor Data Fusion, Boston: Artech House.

5. Bar-Shalom, Y. (Editor) (1992). Multitarget-Multisensor Tracking : Applications and Advances, Artech House.

6. Bar-Shalom, Y. and Li, X.-R. (1995). Multitarget-Multisensor Tracking: Principles and Techniques, Storrs, CT: Yaakov Bar-Shalom.

7. Roy, J. and Bossé, É (1995). Definition of a Performance Evaluation Methology for Sensor Data Fusion Systems. DREV TR 9423. UNCLASSIFIED.

8. Roy, J., Bossé, É, and Duclos-Hindié, N. (1998). Performance Comparison of Contact-level and Track-level Sensor Fusion Architectures. *Otp. Eng.*, **37**(2), 434–440.

9. Hall, D. (1992). Mathematical Techniques in Multisensor Data Fusion, Norwod, MA: Artech House.

10. Antony, R.T. (1995). Principles of Data Fusion Automation, Norwood, MA: Artech House.

11. Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME, Journal of Basic Engineering*, **82**, 34–45.

12. Maybeck, P. (1979). Stochastic Models, Estimation and Control, Vol. 1. Academic Press.

13. Bar-Shalom, Y. and Fortman, T. (1988). Tracking and Data Association, Vol. 179 of *Mathematics in Science and Engineering*. Academic Press.

14. Uhlmann, J. K. (1996). General Data Fusion for Estimates with Unknown Cross Covariances. In *Proceedings of the SPIE AeroSense Conference*, pp. 165–173.

15. Benaskeur, A. and Roy, J. (2001). A Consistent Filter for Robust Decentralized Data Fusion. DRDC - Valcartier TR 2001-223. UNCLASSIFIED.

16. Bar-Shalom, Y. (1981). Development of Track-to-Track Correlation Problem. *IEEE Transactions on Automatic Control*, **2**(26), 571–572.

17. Bar-Shalom, Y. and Campo, L. (1986). The Effect of Common Noise on the Two-Sensor Fused-Track Covariance. *IEEE Transactions on Aerospace and Electronic Systems*, **6**(22), 803–805.

18. Simukai, U. (1995). Network Management in Decentralized Sensing Systems. Ph.D. thesis. University of Oxford.

19. Chee-Yee, C. (1998). Distributed Architecture for Data Fusion. *In Proceedings of the International Conference on Multisource-Multisensor Information Fusion*, pp. 84–91.

20. Miller, M. D., Drummond, O. E., and Perrella, A. J. (1998). Tracklets and Covariance Truncation Options for Threater Missile Tracking. *In Proceedings of the International Conference on Multisource-Multisensor Information Fusion*, pp. 165–173.

21. Chee-Yee, C., Mori, S., and Chang, K. (1985). Information Fusion in Distributed Sensor Networks. *In American Control Conference*, pp. 830–835.

22. Willner, D., Chang, C. B., and Dunn, K. P. (1976). Kalman Filter Algorithms For a Multisensor System. In *Proceedings of the 1976 IEEE Conference on Decision and Control*, pp. 570–574.

23. Blackman, S. (1986). Multi-Target Tracking with Radar Applications, Norwod, MA: Artech House.

24. Kaminski, P., Bryson, A., and Schmidt, S. (1971). Discrete Square-Root Filtering : A Survey of Current Techniques. *IEEE Transactions on Automatic Control*, **16**, 727–735.

25. Harter, H. (1972). The Method of Least-Squares and Some Alternatives. Aerospace Res. Lab., Air Force Systems Command, Wright-Patterson AFB, Ohio. (Technical Report ARL 72-0129).

Annex A
Implementation of Kalman filter

The Kalman filter is usually implemented as a recursive algorithm. The procedure starts from an initial estimate of the state variables, and recursively predicts and adjusts this estimate with each new measurement. From a practical standpoint, this recursiveness makes the Kalman filter very suitable for the on-computer implementation. No storage of the previous data is required, which allows real-time operating. This recursive nature has much helped in making the filter very popular.

A.1 Recursive algorithm

The standard form of the recursive discrete-time Kalman filter, can be summarized in the following steps.

1. **Initialization** — The filter is supplied with initial information, including the estimate of the initial state

$$\hat{\boldsymbol{x}}_{0|0} = E\left[\boldsymbol{x}_0\right]$$

and the measurement error covariance

$$\hat{\boldsymbol{P}}_{0|0} = \boldsymbol{R}$$

Note that different initializations might be used.

2. **Prediction of the state** — The most recent available *a posteriori* estimate of the state, namely $\hat{\boldsymbol{x}}_{k|k}$, is injected in the system model to yield the new *a priori* estimate $\hat{\boldsymbol{x}}_{k+1|k}$ of the state at the time instant $k+1$.

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{F}_k \hat{\boldsymbol{x}}_{k|k} + \boldsymbol{G}_k \boldsymbol{u}_k \tag{A.1}$$

3. **Prediction of the covariance matrix** — A prediction of the estimation error covariance matrix is performed, at the time instance $k+1$. Its computation is based upon the most recent available *a posteriori* value $\hat{\boldsymbol{P}}_{k|k}$.

$$\hat{\boldsymbol{P}}_{k+1|k} = \boldsymbol{F}_k \hat{\boldsymbol{P}}_{k|k} \boldsymbol{F}_k^T + \boldsymbol{Q}^T$$

4. **Kalman Gain matrix** — The obtained so far (predicted) values are used to calculate the new Kalman gain matrix.

$$\boldsymbol{W}_k = \hat{\boldsymbol{P}}_{k+1|k} \boldsymbol{H}_{k+1}^T (\boldsymbol{H}_{k+1} \hat{\boldsymbol{P}}_{k+1|k} \boldsymbol{H}_{k+1}^T + \boldsymbol{R})^{-1}$$

Note that this gain is needed only for the direct form implementation.

5. **Update of the Covariance Matrix** — which gives, in both direct and inverse forms, a confidence measure in the estimate.

- Direct form : The Kalman matrix gain W_{k+1} serves to update the estimation of the covariance matrix.

$$\hat{P}_{k+1|k+1} = (I - W_{k+1}H_{k+1})\hat{P}_{k+1|k}$$

- Inverse form : The update is performed without the need of W_{k+1}.

$$\hat{P}^{-1}_{k+1|k+1} = \hat{P}^{-1}_{k+1|k} + H^T_{k+1}R^{-1}H_{k+1}$$

6. **Update of the state estimate**
   - Direct form : The error[11] that reflects the discrepancy between the predicted measurement based upon the *a priori* estimate $\hat{x}_{k+1|k}$ and the newly available measurement z_{k+1} is determined and multiplied by the gain matrix W_{k+1} to compute the *a posteriori* estimate at time instant $k+1$.

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + W_{k+1}(z_{k+1} - H_{k+1}\hat{x}_{k+1|k})$$

   - Inverse form : the residual is not computed, and the update rule uses a weighted sum of the *a priori* estimate and the new measurement.

$$\hat{x}_{k+1|k+1} = \hat{P}_{k+1|k+1}\left[\hat{P}^{-1}_{k+1|k}\hat{x}_{k+1|k} + H^T_{k+1}R^{-1}z_{k+1}\right]$$

7. At time $k+2$, the cycle starts again at step 2. This cycle will go on at the sampling rate.

Note that the system noise v is not used in the state prediction equation (A.1). This is due to the fact that the best prediction one can make for a white noise is its mean, which happens to be equal to zero here.

A.2 Array algorithm (Kalman filter)

Even though the above-mentioned implementation does not show it explicitly, the Kalman filter can be expressed as a recursive algorithm, which propagates some Riccati equation, whose solution is the covariance matrix $\hat{P}_{k+1|k+1}$. A sufficient condition to the convergence of this recursion is given by the positivity of the matrix $\hat{P}_{k+1|k+1}$. In this algorithm, due, among others, to the unknown modeling/measurement errors, this positivity cannot always be guaranteed.

An interesting solution to this problem is given by the so-called square-root array algorithm [24, 25]. Instead of propagating the covariance itself, this solution propagates its square-root matrix. Hence, squaring the resulting matrix will ensure the positivity of the covariance. The principle of the algorithm is shown below.

[11] Often called *the measurement innovation* or *the residual.*

A.2.1 Time update recursion

The error covariance prediction equation is given by

$$\begin{aligned}\hat{P}_{k+1|k} &= F_k \hat{P}_{k|k} F_k^T + Q \\ &= \left[F_k \hat{P}_{k|k}^{1/2} \quad Q^{1/2}\right] \left[F_k \hat{P}_{k|k}^{1/2} \quad Q^{1/2}\right]^T \\ &= \left[F_k \hat{P}_{k|k}^{1/2} \quad Q^{1/2}\right] \Theta\Theta^T \left[F_k \hat{P}_{k|k}^{1/2} \quad Q^{1/2}\right]^T\end{aligned}$$

where Θ is any unitary matrix, chosen such that

$$\Theta\Theta^T = I$$

and

$$\left[F_k \hat{P}_{k|k}^{1/2} \quad Q^{1/2}\right] \Theta = \left[X \quad 0\right]$$

This means that Θ has as a main role the triangularization of the matrix

$$\left[F_k \hat{P}_{k|k}^{1/2} \quad Q^{1/2}\right]$$

These conditions will ensure that

$$\begin{aligned}XX^T &= \left[X \quad 0\right] \left[X \quad 0\right]^T \\ &= \left[F_k \hat{P}_{k|k}^{1/2} \quad Q^{1/2}\right] \Theta\Theta^T \left[F_k \hat{P}_{k|k}^{1/2} \quad Q^{1/2}\right]^T \\ &= F_k \hat{P}_{k|k} F_k^T + Q\end{aligned}$$

which will provide the square-root (matrix) of the covariance matrix, that is,

$$X = \hat{P}_{k+1|k}^{1/2}$$

This result gives the following recursion for updating this square root, for the *a priori* covariance matrix.

$$\left[F_k \hat{P}_{k|k}^{1/2} \quad Q^{1/2}\right] \Theta = \left[\hat{P}_{k+1|k}^{1/2} \quad 0\right]$$

A.2.2 State update recursion

The equations, for the calculation of the Kalman gain and the update of the error covariance matrix, are given below.

$$\begin{aligned}W_{k+1} &= \hat{P}_{k+1|k} H_{k+1}^T (H_{k+1} \hat{P}_{k+1|k} H_{k+1}^T + R)^{-1} \\ \hat{P}_{k+1|k+1} &= \hat{P}_{k+1|k} - W_{k+1} H_{k+1} \hat{P}_{k+1|k} \\ &= \hat{P}_{k+1|k} - \hat{P}_{k+1|k} H_{k+1}^T (H_{k+1} \hat{P}_{k+1|k} H_{k+1}^T + R)^{-1} H_{k+1} \hat{P}_{k+1|k}\end{aligned}$$

We define the matrix

$$\Gamma = \begin{bmatrix} R^{1/2} & H_{k+1}\hat{P}_{k+1|k}^{1/2} \\ 0 & \hat{P}_{k+1|k}^{1/2} \end{bmatrix}$$

and suppose that we know a matrix Θ such that $\Theta\Theta^T = I$ and

$$\Gamma\Theta = \begin{bmatrix} X & 0 \\ Y & Z \end{bmatrix}$$

By squaring, we will obtain

$$\begin{aligned} \Gamma\Theta\Theta^T\Gamma^T &= \Gamma\Gamma^T \\ &= \begin{bmatrix} XX^T & XY^T \\ YX^T & YY^T + ZZ^T \end{bmatrix} \\ &= \begin{bmatrix} R + H_{k+1}\hat{P}_{k+1|k}H_{k+1}^T & H_{k+1}\hat{P}_{k+1|k} \\ \hat{P}_{k+1|k}H_{k+1}^T & \hat{P}_{k+1|k} \end{bmatrix} \end{aligned}$$

A simple identification of the terms will yield

$$\begin{aligned} X &= \left(R + H_{k+1}\hat{P}_{k+1|k}H_{k+1}^T\right)^{1/2} \\ Y &= \hat{P}_{k+1|k}H_{k+1}^T\left(R + H_{k+1}\hat{P}_{k+1|k}H_{k+1}^T\right)^{-T/2} \\ Z &= \left(\hat{P}_{k+1|k} - YY^T\right)^{1/2} \\ &= \hat{P}_{k+1|k+1}^{1/2} \end{aligned}$$

With these variables, the Kalman gain can be rewritten as

$$W_{k+1} = YX^{-1}$$

This notation allows us to express the recursion as

$$\begin{aligned} \begin{bmatrix} R^{1/2} & H_{k+1}\hat{P}_{k+1|k}^{1/2} \\ 0 & \hat{P}_{k+1|k}^{1/2} \end{bmatrix}\Theta &= \begin{bmatrix} \left(R + H_{k+1}\hat{P}_{k+1|k}H_{k+1}^T\right)^{1/2} & 0 \\ \hat{P}_{k+1|k}H_{k+1}^T\left(R + H_{k+1}\hat{P}_{k+1|k}H_{k+1}^T\right)^{-T/2} & \hat{P}_{k+1|k+1}^{1/2} \end{bmatrix} \\ &= \begin{bmatrix} X & 0 \\ W_{k+1}X^{-1} & \hat{P}_{k+1|k+1}^{1/2} \end{bmatrix} \end{aligned}$$

which provides all the necessary information for the implementation of the Kalman filter. Note that only the square root of the covariance matrix is propagated.

A.3 Array algorithm (covariance intersection)

With a small modification, the same array algorithm can be used to implement the covariance intersection filter. The details of this modification are given below.

A.3.1 Time update recursion

Identical the Kalman filter case.

A.3.2 State update recursion

The state update equation is given by

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{P}}_{k+1|k+1}\left[\omega\hat{\boldsymbol{P}}^{-1}_{k+1|k}\hat{\boldsymbol{x}}_{k+1|k} + (1-\omega)\boldsymbol{H}^T_{k+1}\boldsymbol{R}^{-1}\boldsymbol{z}_{k+1}\right] \tag{A.2}$$

where the error covariance matrix is calculated by

$$\hat{\boldsymbol{P}}^{-1}_{k+1|k+1} = \omega\hat{\boldsymbol{P}}^{-1}_{k+1|k} + (1-\omega)\boldsymbol{H}^T_{k+1}\boldsymbol{R}^{-1}\boldsymbol{H}_{k+1} \tag{A.3}$$

Using the matrix inversion lemma (Eq. 41, p. 11), this covariance can be expressed as

$$\hat{\boldsymbol{P}}_{k+1|k+1} = \frac{\hat{\boldsymbol{P}}_{k+1|k}}{\omega} - \frac{\hat{\boldsymbol{P}}_{k+1|k}}{\omega}\boldsymbol{H}^T_{k+1}\left[\omega\boldsymbol{R} + (1-\omega)\boldsymbol{H}_{k+1}\hat{\boldsymbol{P}}_{k+1|k}\boldsymbol{H}^T_{k+1}\right]^{-1}\boldsymbol{H}_{k+1}\frac{\hat{\boldsymbol{P}}_{k+1|k}}{\omega}$$

Using (A.3), the update rule (A.2) can be rewritten as

$$\begin{aligned}\hat{\boldsymbol{x}}_{k+1|k+1} &= \hat{\boldsymbol{x}}_{k+1|k} + (1-\omega)\hat{\boldsymbol{P}}_k\boldsymbol{H}^T_{k+1}\boldsymbol{R}^{-1}\left(\boldsymbol{z}_{k+1} - \boldsymbol{H}_{k+1}\hat{\boldsymbol{x}}_{k+1|k}\right)\\ &= \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{W}_{k+1}\left(\boldsymbol{z}_{k+1} - \boldsymbol{H}_{k+1}\hat{\boldsymbol{x}}_{k+1|k}\right)\end{aligned}$$

where the covariance intersection gain (which is equivalent to the Kalman gain) is given by

$$\begin{aligned}\boldsymbol{W}_{k+1} &= (1-\omega)\hat{\boldsymbol{P}}_{k+1|k+1}\boldsymbol{H}^T_{k+1}\boldsymbol{R}^{-1}\\ &= (1-\omega)\left[\omega\hat{\boldsymbol{P}}^{-1}_{k+1|k} + (1-\omega)\boldsymbol{H}^T_{k+1}\boldsymbol{R}^{-1}\boldsymbol{H}_{k+1}\right]^{-1}\boldsymbol{H}^T_{k+1}\boldsymbol{R}^{-1}\\ &= (1-\omega)\hat{\boldsymbol{P}}_{k+1|k}\boldsymbol{H}^T_{k+1}\left[\omega\boldsymbol{R} + (1-\omega)\boldsymbol{H}_{k+1}\hat{\boldsymbol{P}}_{k+1|k}\boldsymbol{H}^T_{k+1}\right]^{-1}\end{aligned}$$

As for the Kalman filter, we define the matrix

$$\boldsymbol{\Gamma} = \begin{bmatrix}(\omega\boldsymbol{R})^{1/2} & \boldsymbol{H}_{k+1}\left[(1-\omega)\hat{\boldsymbol{P}}_{k+1|k}\right]^{1/2}\\ \boldsymbol{0} & \left[\dfrac{\hat{\boldsymbol{P}}_{k+1|k}}{\omega}\right]^{1/2}\end{bmatrix}$$

and suppose, also as in the Kalman filter case, that a matrix Θ exists such that $\Theta\Theta^T = I$ and

$$\Gamma\Theta = \begin{bmatrix} X & 0 \\ Y & Z \end{bmatrix}$$

By squaring, we will obtain

$$\begin{aligned}
\Gamma\Theta\Theta^T\Gamma^T &= \Gamma\Gamma^T \\
&= \begin{bmatrix} XX^T & XY^T \\ YX^T & YY^T + ZZ^T \end{bmatrix} \\
&= \begin{bmatrix} \omega R + (1-\omega) H_{k+1}\hat{P}_{k+1|k}H_{k+1}^T & \sqrt{\frac{1-\omega}{\omega}} H_{k+1}\hat{P}_{k+1|k} \\ \sqrt{\frac{1-\omega}{\omega}}\hat{P}_{k+1|k}H_{k+1}^T & \frac{\hat{P}_{k+1|k}}{\omega} \end{bmatrix}
\end{aligned}$$

and by identification,

$$\begin{aligned}
X &= \left[\omega R + (1-\omega) H_{k+1}\hat{P}_{k+1|k}H_{k+1}^T\right]^{1/2} \\
Y &= \sqrt{\frac{1-\omega}{\omega}}\hat{P}_{k+1|k}H_{k+1}^T\left[R + H_{k+1}\hat{P}_{k+1|k}H_{k+1}^T\right]^{-T/2} \\
Z &= \left[\hat{P}_{k+1|k} - YY^T\right]^{1/2} \\
&= \hat{P}_{k+1|k+1}^{1/2}
\end{aligned}$$

The gain W_{k+1} can be rewritten as

$$W_{k+1} = \sqrt{\omega(1-\omega)}YX^{-1}$$

This gives as a final recursion

$$\begin{bmatrix} (\omega R)^{1/2} & H_{k+1}\left[(1-\omega)\hat{P}_{k+1|k}\right]^{1/2} \\ 0 & \left[\frac{\hat{P}_{k+1|k}}{\omega}\right]^{1/2} \end{bmatrix} \Theta = \begin{bmatrix} X & 0 \\ \frac{1}{\sqrt{\omega(1-\omega)}}W_{k+1}X^{-1} & \hat{P}_{k+1|k+1}^{1/2} \end{bmatrix}$$

Annex B
Bayesian interpretation of Kalman filter

There are many ways to interpret the Kalman filter method. Kalman's original paper [11] posed the problem in terms of error (trace) minimization. The Bayesian interpretation of the Kalman filter did not become popular until the early 1970s. From the Bayesian viewpoint, the Kalman filter propagates the conditional probability density of the desired state, conditioned on the knowledge of the measurement data. The Kalman filter maintains the first two moments of the state distribution[12].

$$E\left[\boldsymbol{x}_{k+1}\right] = \hat{\boldsymbol{x}}_{k+1|k+1}$$

$$E\left[(\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k+1})(\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k+1})^T\right] = \hat{\boldsymbol{P}}_{k+1|k+1}$$

If the Gaussian and whiteness conditions, of the noises $\boldsymbol{w}$ and $\boldsymbol{v}$ are satisfied, the state will be normally distributed, with as a mean the *a posteriori* estimate and a variance, the estimation covariance matrix, that is

$$\boldsymbol{f}_x(\boldsymbol{x}_{k+1}|\boldsymbol{z}_{k+1}) = \mathcal{N}(\hat{\boldsymbol{x}}_{k+1|k+1}, \hat{\boldsymbol{P}}_{k+1|k+1})$$

The *a priori* probability is given by

$$\begin{aligned}\boldsymbol{f}_x &= \mathcal{N}(\hat{\boldsymbol{x}}_{k+1|k}, \hat{\boldsymbol{P}}_{k+1|k}) \\ &= \frac{1}{(2\pi)^{n/2}\left|\hat{\boldsymbol{P}}_{k+1|k}\right|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k+1|k})^T \hat{\boldsymbol{P}}^{-1}_{k+1|k}(\boldsymbol{x}_{k+1} - \hat{\boldsymbol{x}}_{k+1|k})\right]\end{aligned}$$

The statistics of the available data, *i.e.* the measurement, can be expressed as

$$\begin{aligned}\boldsymbol{f}_{z|x} &= \mathcal{N}(\boldsymbol{H}_{k+1}\boldsymbol{x}_{k+1}, \boldsymbol{R}) \\ &= \frac{1}{(2\pi)^{n/2}\left|\boldsymbol{R}\right|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{z}_{k+1} - \boldsymbol{H}_{k+1}\boldsymbol{x}_{k+1})^T \boldsymbol{R}^{-1}(\boldsymbol{z}_{k+1} - \boldsymbol{H}_{k+1}\boldsymbol{x}_{k+1})\right]\end{aligned}$$

Given these two probability functions, the estimation problem consists then in maximizing the *a posteriori* estimation probability, whose expression is given by the Bayes's rule

$$\boldsymbol{f}_{x|z} = \frac{\boldsymbol{f}_{z|x}\boldsymbol{f}_x}{\boldsymbol{f}_z}$$

Under these conditions, the *a posteriori* estimate of the state $\boldsymbol{x}$ can be expressed as

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \arg\max_x\left[\frac{\boldsymbol{f}_{z|x}\boldsymbol{f}_x}{\boldsymbol{f}_z}\right] = \arg\max_x\left[\boldsymbol{f}_{z|x}\boldsymbol{f}_x\right]$$

[12] A distribution function $\boldsymbol{f}_x$ of a random variable $\boldsymbol{X}$ is a mathematical relation that gives for each number $\boldsymbol{x}$, the probability of $\boldsymbol{X} \leq \boldsymbol{x}$.

since f_z is independent of x. The same estimate could be obtained by maximizing the logarithm of the *a posteriori* probability function.

$$\begin{aligned}\hat{x}_{k+1|k+1} &= \arg\max_x \ln\left[f_{z|x} f_x\right] \\ &= \arg\max_x \left\{ -\frac{1}{2}\left[\ln((2\pi)^n |\hat{P}_{k+1|k}|) + \ln((2\pi)^n |R|) \right.\right. \\ &\quad + (x_{k+1} - \hat{x}_{k+1|k})^T \hat{P}_{k+1|k}^{-1} (x_{k+1} - \hat{x}_{k+1|k}) \\ &\quad \left.\left. + (z_{k+1} - H_{k+1} x_{k+1})^T R^{-1} (z_{k+1} - H_{k+1} x_{k+1}) \right] \right\}\end{aligned}$$

Computing the negative gradient and setting it equal to zero yields

$$\begin{aligned}0 &= -\frac{\partial}{\partial x} \ln\left(f_{z|x} f_x\right)\Big|_{x=\hat{x}_{k+1|k+1}} \\ &= \hat{P}_{k+1|k}^{-1}(\hat{x}_{k+1|k+1} - \hat{x}_{k+1|k}) - H_{k+1}^T R^{-1}(z_{k+1} - H_{k+1}\hat{x}_{k+1|k+1}) \\ &= \left[\hat{P}_{k+1|k}^{-1} + H_{k+1}^T R^{-1} H_{k+1}\right](\hat{x}_{k+1|k+1} - \hat{x}_{k+1|k}) - H_{k+1}^T R^{-1}(z_{k+1} - H_{k+1}\hat{x}_{k+1|k})\end{aligned}$$

This gives

$$\begin{aligned}\hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + \left[\hat{P}_{k+1|k}^{-1} + H_{k+1}^T R^{-1} H_{k+1}\right]^{-1} H_{k+1}^T R^{-1}(z_{k+1} - H_{k+1}\hat{x}_{k+1|k}) \\ &= \hat{x}_{k+1|k} + W_{k+1}(z_{k+1} - H_{k+1}\hat{x}_{k+1|k})\end{aligned}$$

which is identical to the equation (50, p. 12), obtained within the trace minimization framework.

Distribution List

INTERNAL DISTRIBUTION

TR 2001-224

1 - Director General

1 - Deputy Director General

3 - Document Library

1 - Head/DSS

1 - Head/IKM

1 - Head/SOS

1 - A. Benaskeur (author)

1 - J.M.J. Roy

1 - M. Blanchette

1 - S. Paradis

1 - M. Bélanger

1 - A. Guitouni

1 - I. Abi-Zeid

1 - J. Berger

1 - A. Boukhtouta

1 - R. Breton

1 - A.C. Bourry-Brisset

1 - M. Allouche

1 - H.Irandoust

1 - M. Gagnon

1 - P. Maupin

1 - C. Daigle

INTERNAL DISTRIBUTION (cont'd)

TR 2001-224

1 - A.L. Jousselme

1 - LCol C. Beaumont

1 - LCdr D. Morrissey

1 - Lt(N) E. Woodliffe

1 - Maj. G. Clairoux

1 - Maj. M. Gareau

1 - R. Charpentier

1 - D. Gouin

1 - G. Thibault

1 - J. Bédard

1 - J. Dumas

EXTERNAL DISTRIBUTION

TR 2001-224

1 - DRD KIM

1 - DRD KIM (unbound copy)

1 - DRDC

3 - DGOR

4 - DGJFD

2 - D ST CCIS

2 - D ST L

2 - D ST A

2 - D ST M

2 - D ST HP

2 - D MRS

1 - D MRS 6

1 - D MRS 6-2

2 - D DCEI

1 - D LCSPM

2 - DAR

1 - DAR-4

1 - DAR-3

2 - DMSS

2 - DMSS - 6

2 - DMSS - 8

1 - DSTA - 3 (3d)

1 - DSTM - 2 (1b)

EXTERNAL DISTRIBUTION (cont'd)

TR 2001-224

1 - DSTM - 3 (1a)

1 - DSTM - 5 (1c)

1 - DSTL - 2

2 - DLR

1 - ADM (IM)

1 - CFEC

7 - DRDC - Toronto:
Attn: R. Pigeau
K. Hendy
J. Baranski
F. Lichacz

4 - DRDC - Atlantic:
Attn: J.S. Kennedy
B. McArthur
D. Peters
LCdr B. MacLennan

2 - DRDC - Ottawa:
Attn: M. Rey
B. Bridgewater
P. Yansouni
C. McMillan

1 - PMO MHP

1 - PMO CADRE

1 - PMO AURORA

1 - Canadian Forces Command and Staff College Toronto
Attn: Commanding Officer

1 - CF Maritime Warfare School CFB Halifax
Halifax, Nova Scotia
Attn: Commanding Officer

SANS CLASSIFICATION
COTE DE SÉCURITÉ DE LA FORMULE
(plus haut niveau du titre, du résumé ou des mots-clefs)

FICHE DE CONTRÔLE DU DOCUMENT

1. PROVENANCE (le nom et l'adresse)	2. COTE DE SÉCURITÉ (y compris les notices d'avertissement, s'il y a lieu)
Abder Rezak Benaskeur **Centre de Recherche pour la Défense Valcartier** **2495, boul. Pie-XI Nord** **Val-Bélair Qc, G3J 1X5 Canada**	

3. TITRE (Indiquer la cote de sécurité au moyen de l'abréviation (S, C, R ou U) mise entre parentheses, immédiatement après le titre.)

A comparative study of data fusion algorithms for target tracking (U)

4. AUTEURS (Nom de famille, prénom et initiales. Indiquer les grades militaires, ex : Bleau, Maj. Louis E.)

Abder Rezak Benaskeur et Jean Roy

5 DATE DE PUBLICATION DU DOCUMENT (mois et année)	6a. NOMBRE DE PAGES	6b NOMBRE DE REFERENCES
10/2001	**70**	**25**

7. DESCRIPTION DU DOCUMENT (La catégorie du document, par exemple rapport, note technique ou mémorandum. Indiquer les dates lorsque le rapport couvre une période definie)

Rapport Technique

8. PARRAIN (le nom et l'adresse)

9a. NUMÉRO DU PROJET OU DE LA SUBVENTION (Spécifier si c'est un projet ou une subvention)	9b. NUMÉRO DE CONTRAT
11ba18	

10a. NUMÉRO DU DOCUMENT DE L'ORGANISME EXPÉDITEUR	10b. AUTRES NUMÉROS DU DOCUMENT
	N/A

11. ACCÈS AU DOCUMENT (Toutes les restrictions concernant une diffusion plus ample du document, autres que celles inhérentes à la cote de sécurité)

- ☒ Diffusion illimitée
- ☐ Diffusion limitée aux entrepreneurs des pays suivants (spécifier)
- ☐ Diffusion limitée aux entrepreneurs canadiens (avec une justification)
- ☐ Diffusion limitée aux organismes gouvernementaux (avec une justification)
- ☐ Diffusion limitée aux ministères de la Défense
- ☐ Autres

12. ANNONCE DU DOCUMENT (Toutes les restrictions à l'annonce bibliographique de ce document. Cela correspond, en principe, aux données d'accès au document (11). Lorsqu'une diffusion supplémentaire (à d'autres organismes que ceux précisés à la case 11) est possible, on pourra élargir le cercle de diffusion de l'annonce.)

SANS CLASSIFICATION
COTE DE LA SÉCURITÉ DE LA FORMULE
(plus haut niveau du titre, du résumé ou des mots-clefs)

SANS CLASSIFICATION

COTE DE LA SÉCURITÉ DE LA FORMULE
(plus haut niveau du titre, du résumé ou des mots-clefs)

13. SOMMAIRE (Un résumé clair et concis du document Les renseignements peuvent aussi figurer ailleurs dans le document. Il est souhaitable que le sommaire des documents classifies soit non classifié. Il faut inscrire au commencement de chaque paragraphe du sommaire la cote de sécurité applicable aux renseignements qui s'y trouvent, à moins que le document lui-même soit non classifie. Se servir des lettres suivantes: (S), (C), (R) ou (U). Il n'est pas nécessaire de fournir ici des sommaires dans les deux langues officielles à moins que le document soit bilingue.)

14. MOTS-CLÉS, DESCRIPTEURS OU RENSEIGNEMENTS SPÉCIAUX (Expressions ou mots significatifs du point de vue technique, qui caractérisent un document et peuvent aider à le cataloguer. Il faut choisir des termes qui n'exigent pas de cote de sécurité Des renseignements tels que le modèle de l'équipement, la marque de fabrique, le nom de code du projet militaire, la situation géographique, peuvent servir de mots-cles. Si possible, on doit choisir des mots-clés d'un thésaurus, par exemple le "Thesaurus of Engineering and Scientific Terms (TESTS)". Nommer ce thésaurus. Si l'on ne peut pas trouver de termes non classifies, il faut indiquer la classification de chaque terme comme on le fait avec le titre)

F 518346

CA021607

SANS CLASSIFICATION

COTE DE SÉCURITÉ DE LA FORMULE
(plus haut niveau du titre, du résumé ou des mots-clefs)