

Journal of Advances in Information Fusion

A semi-annual archival publication of the International Society of Information Fusion

Regular Papers

Page

A Generalized Framework for Multi-Criteria Classifiers with Automated Learning: Application on FLIR Ship Imagery 75
Khaled Jabeur, Institut Supérieur de Commerce et de Comptabilité de Bizerte (ISCCB), Tunisia
Adel Guitouni, Defence R&D Canada Valcartier, Canada

A Critical Look at the PMHT 93
David F. Crouse, Marco Guerriero, Peter Willett, University of Connecticut, USA

Multitarget Multisensor Tracking in the Presence of Wakes 117
Anders Rodningsby and Oddvar Hallingstad, University Graduate Center, Norway
Yaakov Bar-Shalom, University of Connecticut, USA
John Glattetre, Kongsberg Maritime, Norway

Track-to-Track Fusion Configurations and Association in a Sliding Window 146
Xin Tian, Yaakov Bar-Shalom, University of Connecticut, USA

*From the ISIF
 Vice President*

*One of our
 Goals is to Ensure
 the High Standards
 of this Flagship
 Publication...*

INTERNATIONAL SOCIETY OF INFORMATION FUSION

The International Society of Information Fusion (ISIF) is the premier professional society and global information resource for multidisciplinary approaches for theoretical and applied INFORMATION FUSION technologies. Technical areas of interest include target tracking, detection theory, applications for information fusion methods, image fusion, fusion systems architectures and management issues, classification, learning, data mining, Bayesian and reasoning methods.

JOURNAL OF ADVANCES IN INFORMATION FUSION: DECEMBER 2009

Editor-In-Chief	W. Dale Blair	Georgia Tech Research Institute, Atlanta, Georgia, USA; 404-407-7934; dale.blair@gtri.gatech.edu
Associate	Uwe Hanebeck	University of Karlsruhe, Germany; +49-721-608-3909; uwe.hanebeck@ieee.org
Administrative Editor	Robert Lynch	Naval Undersea Warfare Center, Newport, Rhode Island, USA; 401-832-8663; robert.s.lynch@navy.mil
Associate	Ruixin Niu	Syracuse University, Syracuse, New York, USA; 860-427-6809; rniu@ecs.syr.edu

EDITORS FOR TECHNICAL AREAS

Tracking	Mahendra Mallick	Georgia Tech Research Institute, Atlanta, Georgia, USA; 404-407-8711; mahendra.mallick@gtri.gatech.edu
Associate	Peter Willett	University of Connecticut, Storrs, Connecticut, USA; 860-486-2195; willett@enr.uconn.edu
Associate	Huimin Chen	University of New Orleans, New Orleans, Louisiana, USA; 504-280-1280; hchen2@uno.edu
Detection	Pramod Varshney	Syracuse University, Syracuse, New York, USA; 315-443-1060; varshney@syr.edu
Fusion Applications	Ben Slocumb	Numerica Corporation; Loveland, Colorado, USA; 970-461-2000; bjslocumb@numerica.us
Image Fusion	Lex Toet	TNO, Soesterberg, 3769de, Netherlands; +31346356237; lex.toet@tno.nl
Fusion Architectures and Management Issues	Chee Chong	BAE Systems, Los Altos, California, USA; 650-210-8822; chee.chong@baesystems.com
Classification, Learning, Data Mining	Müjdat Çetin	Sabancı University, Turkey; +90-216-483-9594; mçetin@sabancıuniv.edu
Associate	Pierre Valin	Defence R&D Canada Valcartier, Quebec, G3J 1X5, Canada; 418-844-4000 ext 4428; pierre.valin@drdc-rddc.gc.ca
Bayesian and Other Reasoning Methods	Shozo Mori	BAE Systems, Los Altos, California, USA; 650-210-8823; shozo.mori@baesystems.com
Associate	Jean Dezert	ONERA, Chatillon, 92320, France; +33146734990; jdezert@yahoo.com

Manuscripts are submitted at <http://jaif.msubmit.net>. If in doubt about the proper editorial area of a contribution, submit it under the unknown area.

INTERNATIONAL SOCIETY OF INFORMATION FUSION

Elisa Shahbazian, *President*

Stefano Coraluppi, *President-elect*

Uwe Hanebeck, *Secretary*

Chee Chong, *Treasurer*

Yaakov Bar-Shalom, *Vice President Publications*

Robert Lynch, *Vice President Communications*

Journal of Advances in Information Fusion (ISSN 1557-6418) is published semi-annually by the International Society of Information Fusion. The responsibility for the contents rests upon the authors and not upon ISIF, the Society, or its members. ISIF is a California Nonprofit Public Benefit Corporation at P.O. Box 4631, Mountain View, California 94040. **Copyright and Reprint Permissions:** Abstracting is permitted with credit to the source. For all other copying, reprint, or republication permissions, contact the Administrative Editor. Copyright© 2009 ISIF, Inc.

From the ISIF Vice President for Publications

December 2009



We want YOUR papers for JAIF!

The *Journal for Advances in Information Fusion* (JAIF), the flagship publication of ISIF, started in 2006 and this (eighth) issue completes its 4th year. While JAIF is a free-access and no-fees-for-authors on-line journal, hard copies of it were distributed as part of the registration package at several FUSION Conferences.

As indicated in a previous editorial, one of our goals is to ensure the high standards of this flagship publication, as appropriate for an archival publication that is worth of this designation. JAIF is fortunate to have as Founding Editor-in-Chief, W. Dale Blair, who with his extensive experience as Editor-in-Chief of the *IEEE Transactions on Aerospace and Electronic Systems*, managed to put together the publication mechanism and orchestrate, together with a dedicated Editorial Board and referees, a reviewing process that is both thorough and is completed (most of the time...) in a timely manner. While the review process can be sometimes painful (I am speaking from my own experience—when I was at the receiving end of harsh reviewer comments)), the final result is always a better paper that is worthy to be archived.

However, there is another part that is needed: YOU, the authors of good papers. Many of the papers presented at the annual FUSION conferences are suitable for submission to JAIF. We approached the candidates for the best paper awards and invited them to submit their papers to JAIF. Since all conference papers can use revisions and improvements, we suggested to the authors to put on their “best dress” for the journal submission. We would also like to invite other authors of papers at FUSION to consider submission to JAIF if they feel the paper is ready for an archival publication.

I realize how difficult and time consuming such a revision can be. I also realize that in non-academic jobs such an endeavor can have negative repercussions. Since in most cases this is not part of the “day job”, it requires spending time that otherwise could be used for family, recreation, or other purposes. Furthermore,

speaking from my own personal experience (as well as actual events that occurred to colleagues), there can be other consequences, like having a boss who could develop a negative attitude towards the author because of this. Nevertheless, in the long run the talents and perseverance will prevail, so such an endeavor is bound to bear fruits that that are not obvious when the author “sweats it out”.

I want to take this opportunity to thank the entire Editorial Board (all volunteers) for their work and solicit additional volunteers for both editorial work as well as reviewers.

Yaakov Bar-Shalom
Vice President for Publications
International Society for Information Fusion

A Generalized Framework for Multi-Criteria Classifiers with Automated Learning: Application on FLIR Ship Imagery

KHALED JABEUR
ADEL GUITOUNI

This paper reviews Multi-Criteria Classifiers (MCCs) or commonly multi-criteria classification methods. These methods have many advantages including flexibility, the integration of human judgments and prevention of black box syndrome. However, these advantages come with a price: large number of parameters to be setup. In particular, this paper focuses on Nominal Concordance/Discordance-based MCCs (NCD-MCCs). A generalized framework is proposed to synthesize the underlying computation algorithm for each MCC. In order to address MCCs disadvantages, an Automated Learning Method (ALM) based on Real-Coded Genetic Algorithm (RCGA) is proposed to infer these parameters. The empirical results of some MCCs are compared with those obtained by other classifiers (e.g. Bayes and Dempster-Shafer classifiers). A military dataset of 2545 Forward Looking Infra-Red (FLIR) images representing eight different classes of ships is therefore used to test the performance of these classifiers. In this paper, we argue the benefits of cross-fertilization of MCCs and information fusion algorithms.

Manuscript received October 2, 2007; revised March 6, 2009; released for publication March 26, 2009.

Refereeing of this contribution was handled by Dr. Pierre Valin.

Authors' addresses: kK. Jabeur, Institut Supérieur de Commerce et de Comptabilité de Bizerte (ISCCB), 7021 Zarzouna Bizerte—Tunisia, E-mail: (khaled.jabeur@fsa.ulaval.ca); A. Guitouni, Defence R&D Canada Valcartier, 2459, Pie-XI Blvd North, Val-Bélair, QC, Canada, G3J 1X5, E-mail: (adel.guitouni@drdc-rddc.gc.ca).

1557-6418/09/\$17.00 © 2009 JAIF

1. INTRODUCTION

Supervised classification often consists in assigning a set of entities (e.g. alternatives, images, projects, subjects) into pre-defined and homogeneous categories. Categories are known a priori either by defining profiles limit between them or by a set of typical profiles (reference prototypes or elements) for each category. Ordinal Classification (or Sorting) usually refers to an order relationship between the categories, and nominal classification otherwise. Recently, a variety of classification methods—based on Artificial Intelligence (AI) and Operations Research (OR) techniques—have been proposed to solve classification problems [41]. Neural Networks (NN), Machine Learning (ML), Rough Sets (RS), Fuzzy Sets (FS) and Multi-Criteria Decision Analysis (MCDA) were used for the development and the validation of these methods. This paper focuses on classification methods based on MCDA methodology.

In this paper we use Multi-Criteria Classifiers (MCCs) to designate supervised classification methods based on MCDA methodology. The most MCCs are based on either outranking or multi-attribute utility approaches. Roy and Moscarola [35], Masaglia and Ostanello [24], Yu [42], Perny [31], Belacel [3] and Henriot [15] have proposed MCCs based on the outranking approach, while M.H.DIS (Multi-group Hierarchical DIScrimination) method [40] and UTADIS (UTilités Additives DIScriminantes) method and its variants ([21], [39], [10]) are typical methods based on multi-attribute utility theory. This paper focuses essentially on outranking-based nominal MCCs where there is no order relationship between the categories. These MCCs are based on concordance/discordance concepts.

Limitation of outranking-based methods is due to the large number of parameters (e.g. discrimination thresholds, weights, reference alternatives, etc.) required. In MCDA context, these parameters are generally elicited using interactive approaches from the decision-maker to articulate his relational preference system: it's the Direct Elicitation Approach (DEA). However, it is difficult for the decision-maker to provide such information in a coherent way when the number of these parameters is considerable. Indirect Elicitation Approach (IEA) or Automatic Learning Methods (ALMs) might be the solution to elicit automatically the values of these parameters based on a training set of pre-assigned examples. These two elicitation approaches will be discussed in Section 3.

This paper makes three main contributions. First, we propose a generalized framework for Nominal Concordance/Discordance-based MCC (NCD-based MCCs). Second, we develop an ALM based on Real-Coded Genetic Algorithm (RCGA) to estimate the parameters of NCD-based MCCs. Then we illustrate and assess the performance of the proposed approach on selected NCD-based MCCs. Even if the purpose of the comparison might be seen limited, we present exper-

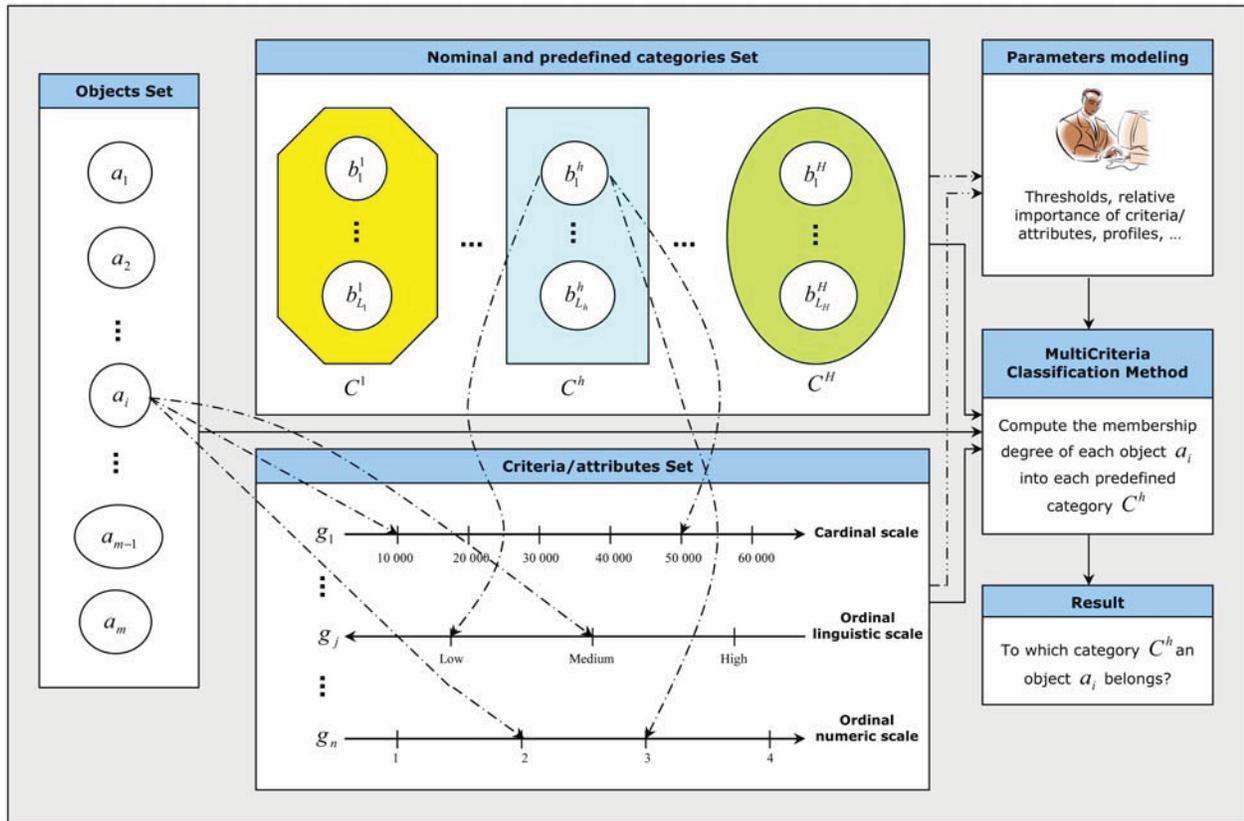


Fig. 1. NCD-based MCCs concepts.

imental results by comparing NCD-based MCCs with other classifiers such as Bayes and Dempster-Shafer classifiers. A dataset of 2545 Forward Looking Infra-Red (FLIR) images representing eight different classes of ships is used for the empirical validation.

This paper is organized as follows. Section 2 presents a generalized framework for NCD-based MCCs. Section 3 proposes an ALM based on RCGA to infer the parameters of the NCD-based MCCs. Section 4 presents a brief description of the dataset used to test the performance of the different classifiers. In Section 5, computational results of some NCD-based MCCs are presented and compared with those obtained by other classifiers. Finally, conclusions, discussions and future works are presented in Section 6.

2. A GENERALIZED FRAMEWORK FOR NCD-BASED MCCs

In our opinion, there are at least three major advantages which distinguish MCCs from the other classifiers:

1. MCCs are designed to incorporate objective and subjective information and deal “correctly” with quantitative and qualitative data. In fact, it is possible to take into account human judgments and compute information obtained on conflicting and heterogeneous dimensions [10]. Therefore, these methods are essential when it is important to explicitly integrate

human judgments (decision-maker’s preferences), to consider many conflicting criteria, and to deal with data obtained on heterogeneous measurement scales (see Fig. 1);

2. MCCs allow pairwise comparisons between the objects to be assigned and the profiles (or reference elements). The pairwise comparison might be seen as a projection isomorphism for each pair of alternatives from the attributes spaces to the preferences spaces. The result of the comparison is a valued function between each pair of alternatives/elements. The aggregation and exploitation of these valued functions avoid computing distance measures obtained on heterogeneous measurement scales (as in K-NN classifier) and allow handling qualitative and/or quantitative information;
3. MCCs avoid the black box situation: it is easy to explain the classification result in natural language.

The advantages or strengths of MCCs could also be seen as weaknesses if subjective human judgments are to be avoided and the information is not heterogeneous and highly correlated. Moreover, these methods require a quite large set of parameters to be determined, which could also be seen as a strength (offering many degrees of freedom). To overcome these limitations, an IEA may be used to infer automatically these parameters. To our knowledge, there exist in the literature four NCD-based MCCs: (i) TRI-NOMFC classifier [23], (ii) PROAFTN

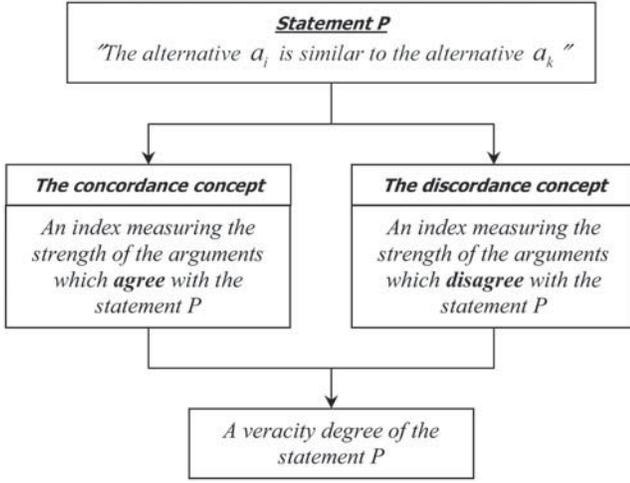


Fig. 2. Indifference relation principle.

classifier [3], (iii) PIP and K-PIP classifiers [15] and (iv) FBI classifier [31].¹ The concordance and discordance concepts were introduced by Bernard Roy [32] when he defined the outranking relation in ELECTRE I method. Indeed, according to many authors ([7], [33], [38]), an outranking relationship can be defined as follows: *an alternative a_i outranks an alternative a_k if and only if there are enough arguments to decide that a_i is at least as good as a_k (concordance concept) while there is no essential reason to refute that statement (discordance concept)*. Recently, Perny [31], Belacel [3] and Henriet [15] extended the above definition by developing an indifference relation measuring the similarity between two alternatives a_i and a_k . The main idea of this relation is to compare the strength of the arguments supporting a proposition to the strength of the arguments opposing the same proposition (see Fig. 2). In fact, it's on the basis of this relation that NCD-based MCCs assign an object (project, alternative, image, etc.) into a predefined category. Hence, according to the NCD-based MCCs, assigning an object consists in computing its membership degree into a predefined category. Membership degree is an aggregate of the evaluation of the indifference between the object to be assigned and each prototype characterizing each category. Each indifference evaluation considers parameters such as discrimination thresholds and criteria/attributes weights.

Let's consider the following notations:

- Let $A = \{a_i\}_{i=1\dots m}$ be a set of m objects to be assigned;
- Let $C = \{C^h\}_{h=1,2,\dots,H}$ be a set of H nominal and predefined categories or classes. Each category C^h is characterized by a set of profiles or reference objects $B^h = \{b_k^h\}_{k=1\dots L_h}$. The set of all profiles is noted by $B = \bigcup_{h=1}^H B^h$;

¹Figueira *et al.* [11] have recently proposed an extension of PROMETHEE method for classification purposes. This method will not be considered in this paper because, according to these authors, some improvements must be done to finalize it.

—Let $F = \{g_j\}_{j=1\dots n}$ be a set of n criteria. We assume that the criteria are to be maximized (transforming a minimization to a maximization is a straight forward). To each criterion g_j , we assign a weight w_j^h ($j = 1\dots n$) which expresses its relative importance in the category C^h ;

—Each object a_i (respectively profile b_k^h) is evaluated on all criteria by the vector: $\mathbf{a}_i = (g_1(a_i), g_2(a_i), \dots, g_n(a_i))$ (respectively by $\mathbf{b}_k^h = (g_1(b_k^h), g_2(b_k^h), \dots, g_n(b_k^h))$).

Most NCD-based MCCs compute for each object a_i a fuzzy number called $\mu(a_i, C^h) \in [0, 1]$, which measures the membership degree of a_i to a given category or class C^h . Hence, if $\mu(a_i, C^h) = 1$ we say that a_i belongs perfectly to the category C^h . However, when $\mu(a_i, C^h) = 0$, we say that a_i has no common ground with the category C^h . Fig. 1 presents a functioning schema of NCD-based MCCs. Fig. 3 shows a generalized framework used by NCD-based MCCs to compute $\mu(a_i, C^h)$.

The membership degree of an object a_i to a category C^h is computed using the concordance and discordance concepts. A local concordance $C_j(a_i, b_k^h)$ and discordance $D_j(a_i, b_k^h)$ indices are computed for each object a_i to be assigned, for each criterion g_j ($j = 1\dots n$) and for each profile b_k^h ($k = 1\dots L_h$) characteristic of C^h .² For instance, PROAFTN method [3] proposes linear functions—similar to those used in ELECTRE III method [34]—for the local concordance and discordance indices (see Fig. 4). In TRINOMFC method [23], only local concordance indices are computed using criteria functions similar to those used in PROMÉTHÉE method [6]. According to [23], it's not appropriate to consider discordance concept when dealing with nominal classification. The computation of local concordance and discordance indices of all NCD-based MCCs are summarised in Tables I and II (see pages 80 and 81).

The computation of local concordance and discordance indices is based on the following types of thresholds: indifference, preference and veto thresholds. The indifference (q_j) and the preference (p_j) thresholds are used to nuance the distinction between two objects into weak and strong preference relationships. The veto threshold (v_j) represents the limit of the tolerance for partial compensation between evaluations. In other word, if the evaluation of a_k is at least v_j different than the evaluation of a_i on criterion g_j , then we may refuse/confirm some propositions about a_i and a_k without regarding their evaluations on the other criteria. Note that these thresholds are established: i) for each criterion in FBI classifier, ii) for each criterion and category in PIP, K-PIP and TRI-NOMFC classifiers and iii) for each criterion, category and profile in PROAFTN classifier. The aggregation operators η_C and η_D are used in order to compute respectively the global concordance and discordance indices (see Ta-

²In PROAFTN classifier [3] each profile b_k^h is defined, for each criterion g_j , by an interval $[S_j^1(b_k^h), S_j^2(b_k^h)]$ which is an exception with regards the other MCCs.

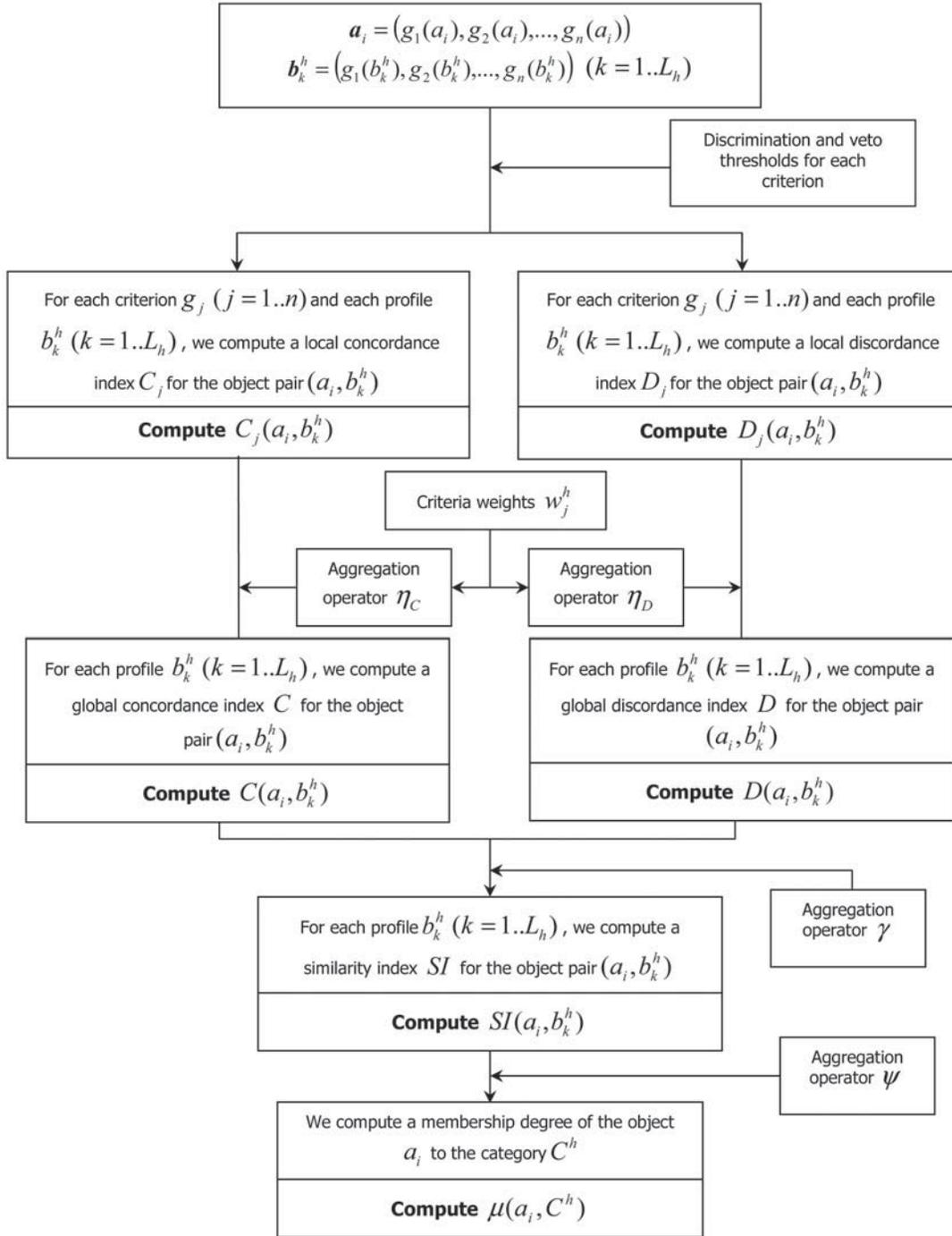


Fig. 3. Generalized framework of NCD-based MCCs to compute $\mu(a_i, C^h)$.

bles I and II). For all NCD-based MCCs, the operator η_C (respectively η_D) is often equal to the weighted sum (respectively weighted geometric mean). The computation of these global indices takes into account the criteria weights. For PROAFTN, TRI-NOMFC, PIP and K-PIP classifiers these weights are determined for each criterion and category whereas for FBI classifier these weights are computed only for each criterion.

The similarity index SI for each pair of objects (a_i, b_k^h) is computed as shown in Table III (see page 81). In general, two kinds of aggregation operator

γ are used to combine the quantities $C(a_i, b_k^h)$ and $(1 - D(a_i, b_k^h))$: the *Product* and the *Minimum*. Since for TRI-NOMFC classifier there is no discordance, the global concordance index is equal to the similarity index; i.e. $C(a_i, b_k^h) = SI(a_i, b_k^h)$. The aggregation operator ψ computes the membership degree $\mu(a_i, C^h)$ of a_i to C^h as shown in Table IV (see page 82). Finally, based on these membership degrees $\mu(a_i, C^h)$, Table V (see page 82) presents the decision rules used by NCD-based MCCs to assign an object a_i to a category C^h .

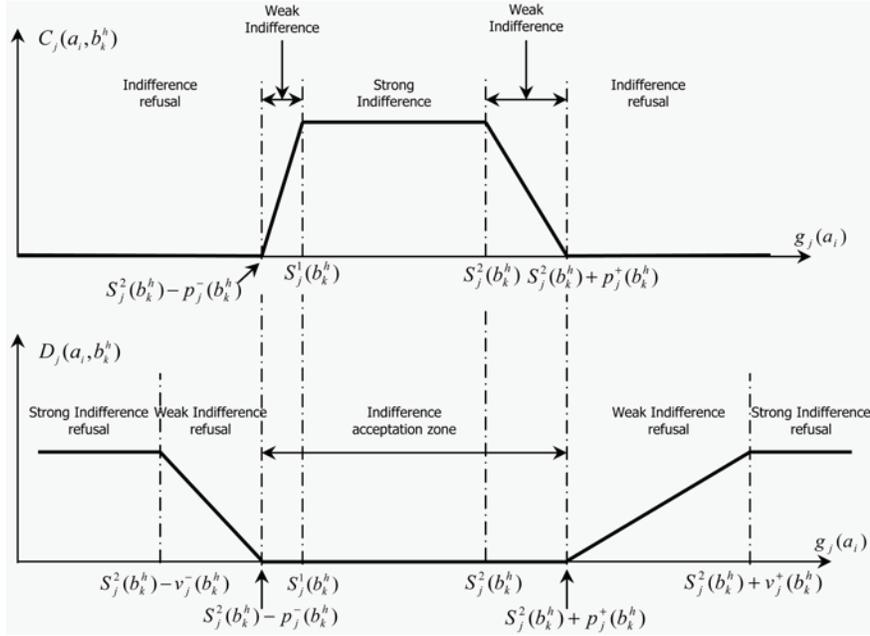


Fig. 4. Representation of local concordance/discordance functions for PROAFTN classifier.

Three interesting elements should be considered from Tables IV and V. The first one is that K-PIP classifier constitutes a generalization of the fuzzy K-NN algorithm [22] where neighbouring is defined by the similarity index SI . Thus, according to this classifier an object a_i has likelihood to be assigned to category C^h if and only if at least one of its profiles b_k^h ($k = 1 \dots L_h$) belongs to the set of the K more similar profiles to a_i in $B = \bigcup_{h=1}^H B^h$. Secondly, TRI-NOMFC classifier [23] has introduced weight coefficients $w_k^h(b_k^h)$ for each profile. Indeed, we believe that the idea of assigning weights to profiles is very interesting since it is possible that, in the same category, a profile $b_{k_1}^h$ is more representative of the category C^h than another profile $b_{k_2}^h$. Finally, the third element is related to the decision rule used in TRI-NOMFC classifier [23] to assign an object a_i to a category C^h . In fact, the advantage of this rule, also known as Hurwitz rule, is to combine an optimistic (*MaxiMax*) and a pessimistic (*MaxiMin*) behaviour in order to provide a more nuanced behaviour in which the optimism level is controlled by the parameter $\alpha \in [0, 1]$; When $\alpha = 1$ (respectively $\alpha = 0$) Hurwitz rule is equivalent to the optimistic rule (*MaxiMax*) (respectively to the pessimistic rule (*MaxiMin*)).

3. ELICITATION OF NCD-BASED MCCs PARAMETERS

According to [41], any multiple criteria classification methodology faces two issues: (i) The specification of the aggregation model to be used, and (ii) the assessment of the parameters of the model. In the above section the first issue is discussed. In this section, we propose first a mathematical model that provides the optimal parameters values of the aggregation model.

Since, as it will be shown later, the proposed mathematical model can not be solved using classical optimization methods, an Automatic Learning Method (ALM) based on Real-Coded Genetic Algorithm (RCGA) is then proposed to approximate the optimal solution of this model and consequently to infer the parameters values of NCD-based MCCs.

3.1. A Mathematical Model to Provide the Optimal Values of NCD-Based MCC Parameters

It is essential to estimate the best values for aggregation model's parameters (e.g. criteria weights, substitution ratios, indifference, preference and veto thresholds). In MCDA literature, two approaches are proposed to elicit the parameters of MCCs: the Direct Elicitation Approach (DEA) and the Indirect Elicitation Approach (IEA). In the first approach, through an interactive questioning, the decision-maker provides the values of these parameters. The aim of this interaction is to ensure that the provided parameters values represent properly the decision-maker judgments and preference system (value or expertise). However, in many other decision-making situations, the determination of the values of these parameters represents a difficult task due to many reasons such as the size of the problem (i.e. high number of parameters), the imprecise nature of the data, the confusing meaning of the parameters, the analyst ability to perform efficiently the elicitation process, etc. Thus, the DEA is often time-consuming and consequently it may discourage the decision-maker from participating. To overcome the drawbacks of the DEA, the IEA employs ALMs to infer automatically the values of these parameters based on examples (or prototypes) known as training objects (part of a training set). In MCDA literature, this second approach is called Preference De-

TABLE I
Local and Global Concordance Indices

Method	Formula
PIP and K-PIP classifiers [15]	$C(a_i, b_k^h) = \sum_{j=1}^n w_j^h \times C_j(a_i, b_k^h) \text{ or } C(a_i, b_k^h) = \text{Min } C_j(a_i, b_k^h), \text{ where}$ $C_j(a_i, b_k^h) = \begin{cases} 0 & \text{if } g_j(a_i) - g_j(b_k^h) \leq -p_j^h \\ \frac{1}{2} + \frac{1}{2} \sin\left(\frac{\pi}{p_j^h - q_j^h}\right) \left(g_j(a_i) - g_j(b_k^h) + \frac{p_j^h + q_j^h}{2}\right) & \text{if } -p_j^h \leq g_j(a_i) - g_j(b_k^h) \leq -q_j^h \\ 1 & \text{if } g_j(a_i) - g_j(b_k^h) \leq q_j^h \\ \frac{1}{2} - \frac{1}{2} \sin\left(\frac{\pi}{p_j^h - q_j^h}\right) \left(g_j(a_i) - g_j(b_k^h) + \frac{p_j^h + q_j^h}{2}\right) & \text{if } q_j^h \leq g_j(a_i) - g_j(b_k^h) \leq p_j^h \\ 0 & \text{otherwise} \end{cases},$ $\sum_{j=1}^n w_j^h = 1 \text{ and } q_j^h \text{ and } p_j^h \text{ are respectively the indifference and the preference thresholds.}$
PROAFTN classifier [3]	$C(a_i, b_k^h) = \sum_{j=1}^n w_j^h \times C_j(a_i, b_k^h), \text{ where } C_j(a_i, b_k^h) = \text{Min} \{C_j^-(a_i, b_k^h), C_j^+(a_i, b_k^h)\},$ $C_j^-(a_i, b_k^h) = \frac{p_j^-(b_k^h) - \text{Min}\{S_j^1(b_k^h) - g_j(a_i), p_j^-(b_k^h)\}}{p_j^-(b_k^h) - \text{Min}\{S_j^1(b_k^h) - g_j(a_i), 0\}},$ $C_j^+(a_i, b_k^h) = \frac{p_j^+(b_k^h) - \text{Min}\{g_j(a_i) - S_j^2(b_k^h), p_j^+(b_k^h)\}}{p_j^+(b_k^h) - \text{Min}\{g_j(a_i) - S_j^2(b_k^h), 0\}},$ $g_j(b_k^h) \text{ is defined by the interval } [S_j^1(b_k^h), S_j^2(b_k^h)],$ $\sum_{j=1}^n w_j^h = 1 \text{ and } p_j^-(b_k^h) \text{ and } p_j^+(b_k^h) \text{ are two preference thresholds.}$
FBI classifier [31]	$C(a_i, b_k^h) = \sum_{j=1}^n w_j^h \times C_j(a_i, b_k^h), \text{ where } C_j(a_i, b_k^h) = \text{Min}\{S_j(a_i, b_k^h), S_j(b_k^h, a_i)\},$ $S_j(a_i, b_k^h) = \frac{p_j - \text{Min}\{g_j(b_k^h) - g_j(a_i), p_j\}}{p_j - \text{Min}\{g_j(b_k^h) - g_j(a_i), p_j\}},$ $\sum_{j=1}^n w_j^h = 1 \text{ and } q_j \text{ and } p_j \text{ are respectively the indifference and the preference thresholds.}$
TRI-NOMFC classifier [23]	$C(a_i, b_k^h) = \sum_{j=1}^n w_j^h \times C_j(a_i, b_k^h), \text{ where } C_j(a_i, b_k^h) = F_j(g_j(a_i) - g_j(b_k^h)) \text{ and } F_j \text{ is an adaptation of some}$ $\text{PROMETHEE criterion functions and } \sum_{j=1}^n w_j^h = 1.$

segregation Approach (PDA) (e.g. [4], [8], [9], [10] and [36]). In artificial intelligence, the IEA is known as Machine Learning (ML) (e.g. [2] and [28]).

Indeed, the mathematical model that provides the optimal values of NCD-based MCCs parameters is presented as follows:

$$(\Lambda) : \begin{cases} \text{Minimize}_{p_i \in P} & \Gamma = \sum_{a_i \in Z} \sum_{h=1}^H (\mu_{ih}(p_1, p_2, \dots, p_s) - \eta_i^h)^2 \\ \text{Subject to} & \text{Structural Constraints (SC)} \\ & \text{Decision-Maker's Constraints (DMC)} \end{cases},$$

where

$$\eta_i^h = \begin{cases} 1 & \text{if } a_i \in C^h \\ 0 & \text{otherwise} \end{cases}, \quad \mu_{ih} \equiv \mu(a_i, C^h), \quad P = \{p_i\}_{i=1 \dots s}$$

is the parameters set and Z is the training set (i.e. a set of objects which assignments is known in advance).³ Γ is the objective function of cumulative classification errors and should be minimised, i.e. the difference between the *estimated membership degree* of a_i (i.e. $\mu(a_i, C^h)$) obtained by applying a NCD-based MCC and the *true membership degree* η_i^h of a_i given a priori in the training set Z . Two types of constraints are considered in Λ : Structural Constraints (SC) and Decision-Maker's Constraints (DMC). In general, the first type of constraints are articulated in function of the characteristics of the parameters and their mutual relationship (e.g. the sum

³The training set is obtained by partitioning the entire dataset in two subsets: the first one, called training subset, is used to elicit the values of the parameters and the second subset, called test subset, is used to evaluate the performance of the MCCs.

TABLE II
Local and Global Discordance Indices

Method	Formula
PIP and K-PIP classifiers [15]	$D(a_i, b_k^h) = 1 - \prod_{j=1}^n (1 - D_j(a_i, b_k^h))$ or $D(a_i, b_k^h) = \text{Max}_j D_j(a_i, b_k^h)$, where $D_j(a_i, b_k^h) = \begin{cases} 1 & \text{if } g_j(a_i) - g_j(b_k^h) \leq -v_{jh}^- \\ \frac{1}{2} - \frac{1}{2} \sin\left(\frac{\pi}{v_{jh}^+ - v_{jh}^-}\right) \left(g_j(a_i) - g_j(b_k^h) + \frac{v_{jh}^+ + v_{jh}^-}{2}\right) & \text{if } -v_{jh}^+ \leq g_j(a_i) - g_j(b_k^h) \leq -v_{jh}^- \\ 0 & \text{if } g_j(a_i) - g_j(b_k^h) \leq v_{jh}^- \\ \frac{1}{2} + \frac{1}{2} \sin\left(\frac{\pi}{v_{jh}^+ - v_{jh}^-}\right) \left(g_j(a_i) - g_j(b_k^h) + \frac{v_{jh}^+ + v_{jh}^-}{2}\right) & \text{if } v_{jh}^- \leq g_j(a_i) - g_j(b_k^h) \leq v_{jh}^+ \\ 1 & \text{otherwise} \end{cases}$ <p>v_{jh}^- and v_{jh}^+ are respectively two veto thresholds.</p>
PROAFTN classifier [3]	$D(a_i, b_k^h) = 1 - \prod_{j=1}^n (1 - D_j(a_i, b_k^h))^{w_j^h}$ where $D_j(a_i, b_k^h) = \text{Max}\{D_j^-(a_i, b_k^h), D_j^+(a_i, b_k^h)\}$ $D_j^-(a_i, b_k^h) = \frac{g_j(a_i) - \text{Max}\{g_j(a_i), S_j^1(b_k^h) - p_j^-(b_k^h)\}}{p_j^-(b_k^h) - \text{Max}\{S_j^1(b_k^h) - g_j(a_i), v_j^-(b_k^h)\}}$ $D_j^+(a_i, b_k^h) = \frac{g_j(a_i) - \text{Min}\{g_j(a_i), S_j^2(b_k^h) + p_j^+(b_k^h)\}}{-p_j^+(b_k^h) - \text{Max}\{-S_j^2(b_k^h) + g_j(a_i), v_j^+(b_k^h)\}}$ <p>where $\sum_{j=1}^n w_j^h = 1$, $p_j^-(b_k^h)$ and $p_j^+(b_k^h)$ are two preference thresholds and $v_j^-(b_k^h)$ and $v_j^+(b_k^h)$ are two veto thresholds.</p>
FBI classifier [31]	$D_i(a_i, b_k^h) = 1 - \prod_{j=1}^n (1 - D_j^i(a_i, b_k^h))^{\alpha/n}$ where $D_j^i(a_i, b_k^h) = \text{Max}\{D_j^s(a_i, b_k^h), D_j^s(b_k^h, a_i)\}$, $D_j^s(a_i, b_k^h) = \text{Min}\left\{1, \text{Max}\left\{0, \frac{g_j(b_k^h) - g_j(a_i) - p_j}{v_j - p_j}\right\}\right\}$, $\alpha \in [1, n]$ is a technical parameter introduced to modify the degree of synergy between criteria and p_j and v_j are respectively the preference and the veto thresholds.
TRI-NOMFC classifier [23]	No discordance

TABLE III
Similarity Index Computation

Method	Formula
PIP and K-PIP classifiers [15]	<ul style="list-style-type: none"> - $SI(a_i, b_k^h) = C(a_i, b_k^h) \times (1 - D(a_i, b_k^h))$ - $SI(a_i, b_k^h) = \text{Min}(C(a_i, b_k^h), 1 - D(a_i, b_k^h))$
PROAFTN classifier [3]	<ul style="list-style-type: none"> - $SI(a_i, b_k^h) = C(a_i, b_k^h) \times (1 - D(a_i, b_k^h))$
FBI classifier [31]	<ul style="list-style-type: none"> - $SI(a_i, b_k^h) = \text{Min}(C(a_i, b_k^h), 1 - D(a_i, b_k^h))$
TRI-NOMFC classifier [23]	<ul style="list-style-type: none"> - $SI(a_i, b_k^h) = C(a_i, b_k^h)$ since there is no discordance.

of the criteria weights for each category is equal to 1, i.e. $\sum_{j=1}^n w_j^h = 1$; the indifference threshold is smaller or equal than the preference threshold for each criterion, i.e. $q_j^h \leq p_j^h$ ($j = 1 \dots n$). The second group of constraints expresses the preferences of decision-maker with respect to the NCD-based MCC parameters. For instance, the decision-maker may specify, for a particular category, that $\sum_{j \in I} w_j^h \geq \sum_{j \in J} w_j^h$ where $I, J \subset \{1 \dots n\}$ and $I \cap J = \emptyset$.

It's noteworthy that Belacel [3] and Henriet [15] have proposed similar mathematical models to infer the parameters values of their respective NCD-based MCC. When the values of the different thresholds are known, Henriet [15] has shown that for specific configurations of global concordance, discordance and similarity indices, the mathematical model (Λ) can be easily relaxed into linear program and then solved by classical optimization methods. In addition, Henriet [15] has pro-

TABLE IV
Membership Degree Computation

Method	Formula
PIP and K-PIP classifiers [15]	<ul style="list-style-type: none"> - PIP: $\mu(a_i, C^h) = \begin{cases} \text{Max}_{b_k^h \in V_K(a_i) \cap B^h} SI(a_i, b_k^h) & \text{if } V_K(a_i) \cap B^h \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$, - K-PIP: $\mu(a_i, C^h) = \begin{cases} 1 - \prod_{b_k^h \in V_K(a_i) \cap B^h} (1 - SI(a_i, b_k^h)) & \text{if } V_K(a_i) \cap B^h \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$, <p>where B^h is the profile set of the category C^h, $V_K(a_i)$ is the set of the K more indifferent (or similar) profiles to a_i in $B = \bigcup_{h=1}^H B^h$.</p>
PROAFTN classifier [3]	- $\mu(a_i, C^h) = \text{Max}_k SI(a_i, b_k^h)$
FBI classifier [31]	- $\mu(a_i, C^h) = \text{Max}_k SI(a_i, b_k^h)$
TRI-NOMFC classifier [23]	<ul style="list-style-type: none"> - $\mu(a_i, C^h) = \text{Min}_k SI(a_i, b_k^h)$ - $\mu(a_i, C^h) = \text{Max}_k SI(a_i, b_k^h)$ - $\mu(a_i, C^h) = \sum_{k=1}^{L_h} w_k^h(b_k^h) \times SI(a_i, b_k^h)$ where $w_k^h(b_k^h)$ is the likelihood degree of object b_k^h and $\sum_{k=1}^{L_h} w_k^h(b_k^h) = 1$.

TABLE V
Decision Rules to Assign a Given Object

Method	Decision rule
PIP and K-PIP classifiers [15]	- $a_i \in C^* \Leftrightarrow \mu(a_i, C^*) = \text{Max}_h \mu(a_i, C^h)$
PROAFTN classifier [3]	- $a_i \in C^* \Leftrightarrow \mu(a_i, C^*) = \text{Max}_h \mu(a_i, C^h)$
FBI classifier [31]	- $a_i \in C^* \Leftrightarrow \mu(a_i, C^*) = \text{Max}_h \mu(a_i, C^h)$
TRI-NOMFC classifier [23]	<ul style="list-style-type: none"> - $a_i \in C^* \Leftrightarrow \mu(a_i, C^*) = \text{Max}_h \left\{ \begin{array}{l} \alpha \times \left(\text{Max}_{k \in \{1..L_h\}} \{ SI(a_i, b_k^h) \} \right) \\ + (1 - \alpha) \times \text{Min}_{k \in \{1..L_h\}} \{ SI(a_i, b_k^h) \} \end{array} \right\}$, where $\alpha \in [0, 1]$ is a coefficient of relative optimism. - $a_i \in C^* \Leftrightarrow \mu(a_i, C^*) = \text{Max}_h \mu(a_i, C^h)$

posed two methods based respectively on K-Means and Genetic algorithms to identify the profiles which characterize each category. In this same perspective, Belacel *et al.* [4] have presented a methodology to infer the parameters of PROAFTN classifier. In their methodology, these authors have made some simplifications: only concordance concept is considered, the criteria weights

are assumed to be equal and each category is characterized by only one profile. Hence, the parameters that are inferred in their learning process are: the upper and the lower bounds of the interval $[S_j^1(b_k^h), S_j^2(b_k^h)]$ and the two preference thresholds $p_j^-(b_k^h)$ and $p_j^+(b_k^h)$. Belacel *et al.* [4] solved a mathematical model similar to (Λ) by using a training set and the Reduced Variable Neighbour-

hood Search (RVNS) meta-heuristic recently proposed by [29]. Belacel *et al.* [4] have reported that the Average Identification Rate (AIR)⁴ of PROAFTN is, in general, better than the AIR of other classification methods reported on the same datasets.

Since the objective function Γ of (Λ) is neither convex nor concave and may have many local optima, it will be difficult to find a global optimum for (Λ) . Therefore, it's not possible to use classical optimization methods (e.g. gradient algorithms and interior-point algorithms) to solve (Λ) . To overcome this difficulty, an ALM based on RCGA is proposed to approximate the optimal solution of (Λ) and consequently to infer the parameters values of NCD-based MCCs.

3.2. An ALM Based on RCGA to Infer the Values of NCD-Based MCC Parameters

Genetic Algorithms (GAs) are stochastic algorithms based on the mechanism of the genetic evolution (selection, cross-over and mutation) to solve complex and large optimization problems. GAs were initially introduced by John Holland [19], but they were popularized thanks to the book of Goldberg [12]. The main idea of GAs is to start with an initial population of potential solutions (or chromosomes) arbitrarily selected. Then, evaluate the relative performance of each solution through a fitness function. Then, on the basis of solutions performances, generate a new population using three evolutionary operators: selection, crossover and mutation. The selection operator identifies both the relatively "good" solutions that will be used to generate the new population and the relatively "bad" solutions that will be removed from the current population. The crossover operator swaps the structures of two "parent solutions" in order to form two similar "offspring solutions" that will be involved in the new population. The mutation operator alters arbitrarily the features of one or more solutions in order to increase the structural variability of the population. The above three operators are repeated until a stopping condition is met. A simplified structure of genetic algorithm is shown in Fig. 5.

The application fields of GAs are considerable. For instance, these algorithms are used in:

- Optimization, when the functions to optimize are complex, irregular and with high dimensionality;
- Physics, as optimization methods for real problems (e.g. structures optimization);
- Artificial intelligence, where the adaptive abilities of GAs are exploited;
- Economy, to model the behaviour of agents for instance;
- Image recognition, for example to classify the unknown objects to pre-defined categories;

⁴This is a performance measure of a classifier. It's defined by the following ratio: the number of objects that are correctly classified divided by the total number of objects.

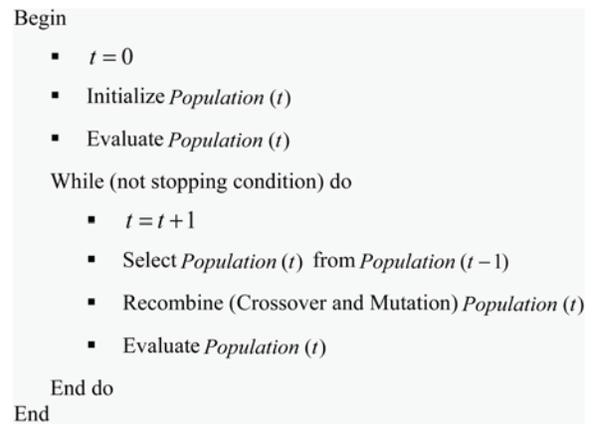


Fig. 5. A simplified architecture of a GA.

—Graph and game theories, to solve for example the Traveling Salesman Problem (TSP) or some problems in repetitive and differential games;

—...

The success of GAs is mainly due to their ability to exploit vast unknown search spaces in order to orient subsequent searches into useful subspaces. This feature makes GAs more efficient and effective search technique to explore large, complex, and poorly understood search spaces, where classical search tolls are inappropriate. Since many years, binary coded solutions (or chromosomes) have dominated GAs research. However, Michalewicz [25] has showed that this kind of representation for optimization problems with continuous variables may involves at least three drawbacks:

- It's difficult to use binary coding for optimization problems with high dimension and numerical precision. For instance, with 100 variables belonging to the interval $[-500, 500]$ and a precision of 6 decimal numbers, the size of a binary coded solution is 3000. This generates a search space of about 10^{1000} . For this kind of binary coded problems GAs will have weak performance [27];
- The Hamming⁵ distance between two neighborhood real numbers may be large in binary coding. For example, the Hamming distance between 0111 (which is equal to 7) and 1000 (which equal to 8), is equal to 4;
- When the crossover and the mutation operators are applied on binary coded continuous chromosomes they may generate new infeasible solutions.

The above drawbacks of the binary coding have motivated the development of other coding types. Real coding is particularly natural when optimization problems involve real variables. GAs with this type of coding are called real-coded GAs (RCGAs) [18]. In the recent years, RCGAs have been used to solve various continu-

⁵The Hamming distance between two binary coded strings is defined as the number of bits which are different in the two strings.

ous optimization problems (e.g. [26], [16], [17], [1]). In RCGAs, each solution (or chromosome) is treated as a vector of real numbers. Since the conventional crossover and mutation operators for binary coding are not applicable for real coding, many other adapted operators are proposed in the literature for real coding [18].

In this work, RCGAs will be used to infer the parameters of the NCD-Based MCCs since all of them are real numbers (e.g. thresholds, weights, etc.). To implement the RCGA, some technical choices have been made on its parameters (e.g. selection methods, crossover and mutation operators, etc). These choices will be specified in Section 5. In next section, we briefly describe the military dataset that will be used to test the performance of the different classifiers.

4. MILITARY DATASET DESCRIPTION

The military dataset that will be used in this work includes 2545 Forward Looking Infra-Red (FLIR) images belonging to eight different classes of ships. These images were provided by the U.S. Naval Weapons Center and Ford Aerospace Corporation. Typical silhouettes of the best image of each class and other related information about classes are listed in Table VI (see page 85).

Based on these 2545 FLIR images, Park and Sklansky [30] proposed to extract 11 features⁶ (or attributes/criteria). These attributes are obtained as follows:

- The first seven (7) attributes are represented by Hu's [20] moments m_i . These moments are invariant under scaling (different zoom factors), rotation (different look angles) and translation (silhouette not necessarily centered). The moments m_i are computed by using the second and the third order moment formula, let:

$$\mu_{nm} = \sum_{(x,y) \in S} (x - \bar{x})^n (y - \bar{y})^m,$$

where $(n + m)$ is the moment order; x (respectively y) is the horizontal (respectively the vertical) coordinates in the silhouette S ; \bar{x} and \bar{y} are the coordinates of the centroid of S . For instance, the first four (among the seven attributes) Hu's [20] invariant moments m_i ($i = 1 \dots 4$) are given as follows:

— $g_1 = m_1 = r/B$, where $r = \sqrt{\mu_{20} + \mu_{02}}$ the radius of gyration and B is the distance between the camera and the ship.

$$— g_2 = m_2 = \frac{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}{r^4}$$

$$— g_3 = m_3 = \frac{(\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{30})^2}{r^6}$$

$$— g_4 = m_4 = \frac{(\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{30})^2}{r^6}$$

⁶A feature is an abstraction of the raw data in order to represent the original information.

The seven Hu's [20] invariant moments are noted by $g_i = m_i$ for $i = 1 \dots 7$. It's worth noting that the weakness of invariant features is that they contain only information that deals with the general shape of the ship and thereby they represent poorly the other details of the observed object. To overcome this disadvantage, Hu [20] proposed four other attributes which provide more information details about ship;

- The last four (4) attributes represent the parameters of an Auto Regressive (AR) Model. They were extracted by fitting an AR model to one-dimensional sequence which represents the projection of a ship image onto horizontal axis. Let $r(i)$, $i = 1 \dots N$, denote the sequence of the projected ship image sampled at N equally spaced points. Based on these sequences, an AR model is defined recursively by:

$$r(i) = \sum_{j=1}^m \theta_j r(i-j) + \alpha + \sqrt{\beta} \varepsilon(i).$$

The above model expresses the projection r_i ($i = 1 \dots N$) as a linear combination of the previous projections $r(i-j)$ ($j = 1 \dots m$),⁷ plus a bias α and the error $\varepsilon(i)$ associated with the model. The parameters are estimated by a least square fit of the model to the one-dimensional sequence $r(1), r(2), \dots, r(N)$. Thus, if $\hat{\theta}$, $\hat{\alpha}$ and $\hat{\beta}$ denote the least squares estimates of θ , α and β respectively, the four (4) AR parameters ($m = 3$, $N = 30$) are presented as follows:

$$— g_{i+7} = \hat{\theta}_i, \quad i = 1 \dots 3$$

$$— g_{11} = \frac{\hat{\alpha}}{\sqrt{\hat{\beta}}}$$

Park and Sklansky [30] have shown that all AR parameters are invariant to rotation, translation and scaling, so that they may be used as features for classification purpose.

5. COMPUTATIONAL RESULTS

Only two NCD-based MCCs are implemented in this work: PROAFTN classifier [3] and K-PIP classifier [15]. This choice is justified by the two following facts. The first one is that K-PIP classifier is an enhanced version of the FBI classifier [31] (see [5]). Second, since we want to experiment the effects of both concordance and discordance concepts on classification results, TRI-NOMFC classifier [23] is removed from our list.

In the ALM proposed in this paper, only thresholds will be estimated for both PROAFTN and K-PIP classifiers:

- For PROAFTN classifier [3], we consider the upper and the lower bounds of the interval $[S_j^1(b_k^h), S_j^2(b_k^h)]$, the two preference thresholds $p_j^-(b_k^h)$ and $p_j^+(b_k^h)$ and

⁷Here m denotes the number of weight parameters.

TABLE VI
Military Dataset Description

Class	Class of ship	Number of images	Typical silhouette
1	Destroyer (D)	340	
2	Container (CO)	455	
3	Civilian Freighter (CF)	186	
4	Auxiliary Oil Replenishment (AOR)	490	
5	Landing Assault Tanker (LAT)	348	
6	Frigate (F)	279	
7	Cruiser (CR)	239	
8	Destroyer with Guided Missile (DGM)	208	

the two veto thresholds $v_j^-(b_k^h)$ and $v_j^+(b_k^h)$ for $j = 1 \dots 11$, $h = 1 \dots 8$ and $k = 1 \dots L_h$;

—For K-PIP classifier⁸ [15], we infer the indifference threshold q_j^h , the preference threshold p_j^h and finally the two veto thresholds v_{jh}^- and v_{jh}^+ for $j = 1 \dots 11$ and $h = 1 \dots 8$.

It's obvious that the dimensionality (or the number of parameters to infer) of the ALM for the above two classifiers is not the same. For instance, in K-PIP classifier there is only 352 parameters ($4 \times 11 \times 8$) to estimate while in PROAFTN classifier there is 5280 parameters ($6 \times 11 \times 8 \times 10$) to estimate if we

assume that each category is represented by only 10 profiles.

Since the criteria weights are not included in the ALM, they are estimated by the Entropy method [43]. Hence, the more the criterion discriminates between images the more it will be important. In the other hand, the profiles of each category are identified by using an improved version of K-Means algorithm.⁹ The number of profiles in each category is determined by a percentage of the total number of objects in this category. This percentage varies from 1% to 10%. Many others technical choices have been made to implement the RCGA on which is based the

⁸In this computational experiment, the number K of K-PIP classifier (see Table IV) is fixed to 5.

⁹In this improved version of K-means algorithm, we remove each profile that forms an empty group.

ALM:

- Four selection methods are implemented: (1) Roulette Wheel Selection (RWS), (2) Stochastic Remainder Without Replacement Selection (SRWRS), (3) Linear Rank based Selection (LRS) and (4) Tournament Selection (TS)¹⁰;
- Five crossover operators are implemented: (1) Flat Crossover (FC), (2) Arithmetical Crossover (AC), (3) BLX- α Crossover (BLXC), (4) Extended Line Crossover (ELC) and (5) Simple Crossover (SC)¹⁰;
- Four mutation operators are implemented: (1) Random Uniform Mutation (RUM), (2) Non Uniform Mutation (NUM), (3) Mühlenbein Mutation (MM) and (4) Gaussian Mutation (GM)¹⁰;
- The crossover and mutation probabilities vary respectively from 0.6 to 0.8 and from 0.05 to 0.1;
- The size of the generated populations varies from 30 to 80. Note that the initial population (or the set of initial chromosomes) is generated at random. However, the random values of parameters that constitute each chromosome are generated within specific intervals. These intervals are determined based on some statistical measures on the training dataset for each class and each feature (some examples of these statistical measures are presented in Fig. 6). The aim of these measures is to limit the variation domains of the parameters and thereby to make easy the random generation of the initial population;
- The maximum iteration number—fixed to 100—is used as stopping criteria for the RCGA. In fact, this number is not fixed at random. Indeed, by testing many data splits, we have observed that—in general—beyond 100 iterations the improvement of the classification accuracy of the tested classifiers becomes insignificant regarding the computational effort (in time) provided to execute an additional iteration;
- The method used for the evaluation of the classification accuracy is a **cross-validation** method called **repeated random sub-sampling validation**. This technique randomly splits the initial dataset into training and validation (or test) subsets. For each such split, the classifier is retrained with the training subset and validated (or tested) on the remaining subset. The results from each split are then averaged. Hence, according to this cross-validation technique the military dataset is randomly divided into two subsets: a training subset (which size varies from 50% to 70% of the entire dataset) used to infer the values of NCD-based MCC parameters and a test (or a validation) subset (which size varies from 50% to 30% of the entire dataset) used to evaluate the performance of the different MCCs.¹¹ Hence, for each MCC, 20 dif-

¹⁰To learn more about these evolutionary operators (crossover and mutation) and methods (selection), we refer the reader to the work of Herrera *et al.* [18].

¹¹Note that each subdivision constitutes a partition of the entire dataset, i.e. the union of the training subset and the test subset form the entire dataset.

ferent random splits are generated to test its performance.

All algorithms in the ALM—i.e. the RCGA, PROAFTN and K-PIP classifiers—are coded in Visual Basic (VB) and tested on a Pentium IV processor with 2.8 GHz and 512 Mb of RAM. The developed software involves some visualization and statistical tools on the entire, training and test datasets. For instance, Fig. 6 presents some statistical measures on the training dataset for each class and each attribute (or feature).

It's important to underline that two prior works ([37] and [30]) have used the same military dataset to test the performance of four different classifiers: Dempster-Shafer-based (DS) classifier, Modified-Bayes-based (MB) classifier, K-Nearest Neighbors (K-NN) classifier and Neural Net (NN) classifier. The results, expressed in Average Identification Rate (AIR), obtained by these works are presented in Table VII.

Note that the above works, i.e. Valin *et al.* (2006) and Park and Sklansky (1990), use the repeated random sub-sampling cross-validation method to evaluate the classification accuracy of all their tested classifiers. Valin *et al.* (2006) generate many random splits by using Monte-Carlo runs¹⁴ whereas Park and Sklansky (1990) generate only one random split.

The application of PROAFTN and K-PIP classifiers on the military dataset provides the results presented respectively in Tables VIII and IX. An example of screen showing the application of the ALM for K-PIP classifier is presented in Fig. 7. By observing Tables VIII and IX, we conclude that both PROAFTN and K-PIP NCD-based MCCs give, in general, good results: the AIR of PROAFTN is 86.78% and the AIR of K-PIP is 80.69%. Hence, these two MCCs have an AIR better than those of MB and DS classifiers but worse than those of K-NN and NN classifiers.

Some other comments may be made on the classification results of both PROAFTN and K-PIP classifiers:

- The AIRs of both classifiers on the training and test datasets are stable since they are situated around their average (small standard deviation for both classifiers and datasets). We can also use the coefficient of variation ($CV = (\sigma_{AIRs} / \bar{X}_{AIRs})\%$)¹⁵ as measure of the robustness (or the stability) of the obtained results. For instance, the CV of the PROAFTN AIRs for the test dataset is equal to $CV = 1.65/86.78 = 1.9\%$ (see Table VIII) which is small.
- The AIRs obtained on the training and the test datasets are too different for PROAFTN classifier (average range 7.04%), while these AIRs are nearly the same for K-PIP classifier (average range 2.53%). We believe that this is due both to the high number of

¹⁴The number of splits is not specified by these authors.

¹⁵Note that more the CV is small the more the observations are homogeneous, i.e. that the observations are concentrated around the mean. In this case we said that the mean is representative of its observations.

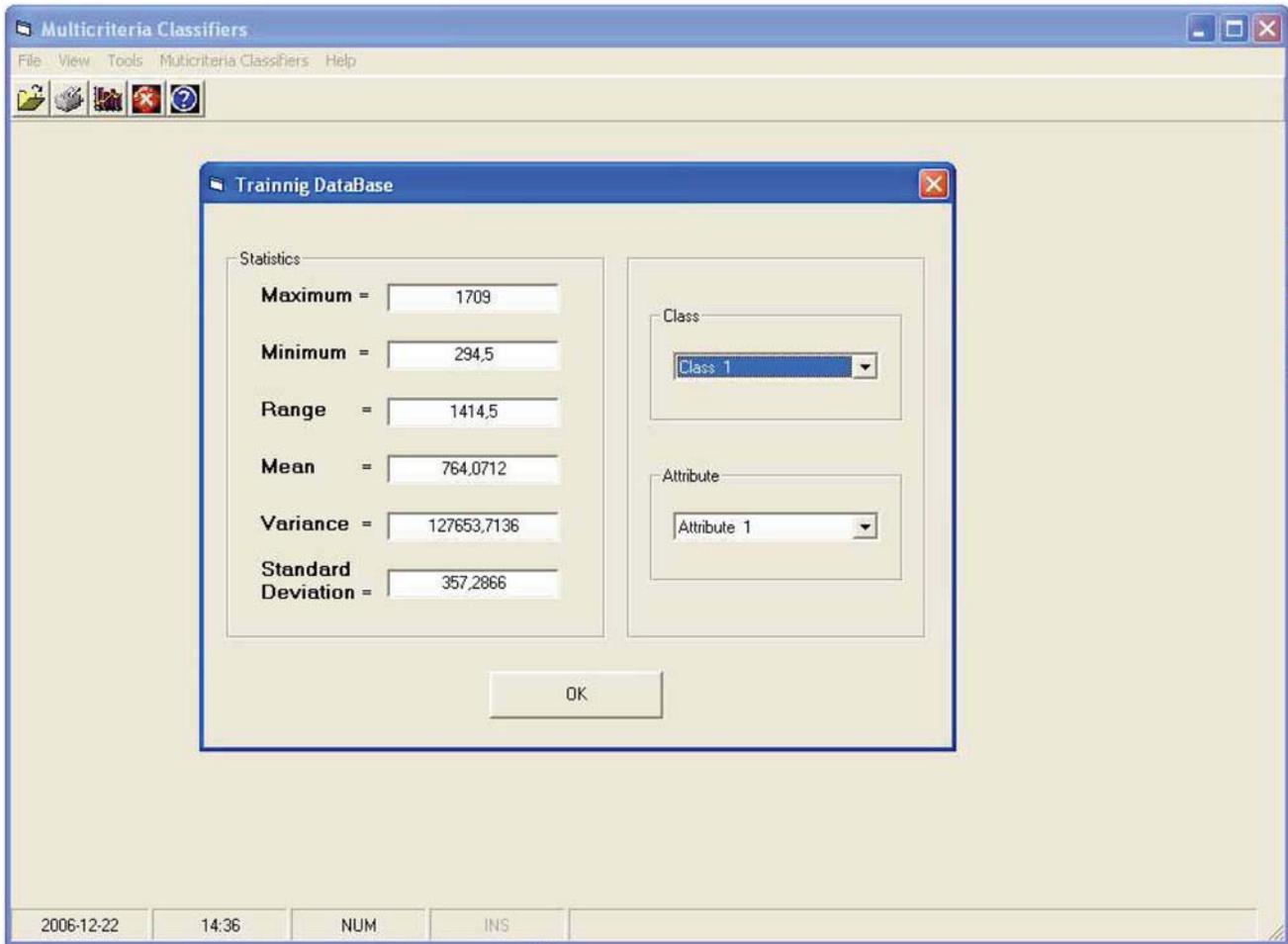


Fig. 6. Descriptive statistical tools.

TABLE VII
The Different Results of Prior Works on the Same Military Dataset

Papers	MB classifier	DS classifier	K-NN classifier	NN classifier
Valin <i>et al.</i> [37]	77.7%	74.5%	94.8% ¹²	92.7%
Park and Sklansky [30]	***	***	88.3% ¹³	***

¹²In this work, the K-NN classifier is applied with $K = 3$ and by using an Euclidean distance weighted by the inverse of the inter-categories covariance matrix.

¹³In this work, the K-NN classifier is applied with a simple Euclidean distance. The value of K that produces the highest AIR is chosen among the values of K between 1 and 17.

parameters in PROAFTN classifier and to the over-specification (or over-fitting) problem. This problem occurs when the parameters of the classifier became much specific to the data set from which they are assessed. Hence, when these parameters are used to classify another data set, the classification results obtained on this latter will be much different from those obtained on the first data set. Since PROAFTN classifier use more parameters than K-PIP classifier, the over-specification problem will be more apparent with the former classifier;

—According to our computational experiments, all selection methods and evolutionary operators (mutation

and crossover) seems to perform equally. However, we have observed that, in general, the AIRs of both MCCs increase when the population size increases;

—Some AIRs obtained by PROAFTN classifier for the training dataset exceed the best AIR obtained by all other classifiers (i.e. 94.8%). This is shows the ability of this classifier to provide better results and thereby we believe that it constitutes a promising classifier which merits to be improved (see Section 6 for eventual improvements);

—The AIR obtained by PROAFTN classifier is better than that of K-PIP classifier since, for a specific criterion, the first consider that the profiles of the same

TABLE VIII
PROAFTN Results

Problems	Genetic Algorithm		Population size	Selection Method	Crossover Step		Mutation Step	
	Training Database	Test database			Cross. Oper	Cross. Prob	Muta. Oper	Muta. Prob
Problem 1	92,62%	86,23%	30	RWS	FC	0,60	RUM	0,05
Problem 2	91,78%	85,03%	30	SRWRS	AC	0,65	NUM	0,07
Problem 3	94,33%	88,54%	30	LRS	SC	0,70	MM	0,09
Problem 4	90,44%	85,09%	40	TS	BLX	0,75	GM	0,10
Problem 5	95,22%	89,67%	40	RWS	ELC	0,80	RUM	0,05
Problem 6	93,02%	84,88%	40	SRWRS	FC	0,60	NUM	0,07
Problem 7	96,17%	88,98%	50	LRS	AC	0,65	MM	0,09
Problem 8	94,01%	86,11%	50	TS	SC	0,70	GM	0,10
Problem 9	93,28%	85,96%	50	RWS	BLX	0,75	RUM	0,05
Problem 10	93,72%	86,31%	60	SRWRS	ELC	0,80	NUM	0,07
Problem 11	95,12%	88,17%	60	LRS	FC	0,60	MM	0,09
Problem 12	91,73%	84,39%	60	TS	AC	0,65	GM	0,10
Problem 13	95,09%	87,81%	70	RWS	SC	0,70	RUM	0,05
Problem 14	96,00%	89,42%	70	SRWRS	BLX	0,75	NUM	0,07
Problem 15	93,08%	86,12%	70	LRS	ELC	0,80	MM	0,09
Problem 16	94,65%	88,23%	70	TS	FC	0,60	GM	0,10
Problem 17	95,00%	87,93%	80	RWS	AC	0,65	RUM	0,05
Problem 18	93,75%	85,59%	80	SRWRS	SC	0,70	NUM	0,07
Problem 19	92,82%	85,07%	80	LRS	BLX	0,75	MM	0,09
Problem 20	94,68%	86,12%	80	TS	ELC	0,80	GM	0,10
Mean	93,83%	86,78%						
Standad deviation	1,50%	1,65%						
Mean range	7,04%							

category don't have necessarily the same thresholds, while the second assumes that the profiles of the same category have identical thresholds. Hence, for a particular category and criterion, PROAFTN classifier provides more thresholds to each profile (i.e. more degree of freedom) than K-PIP classifier.

6. DISCUSSIONS AND CONCLUSIONS

In this paper, a classification methodology that combines the advantages of multi-criteria decision analysis and automated learning algorithms has been proposed. This classification methodology uses some selected NCD-based MCCs as aggregation models and an IEA to assess the parameters values of these MCCs. To understand the implementation of NCD-based MCCs, we have proposed a generalized framework to explain how these classifiers proceed to assign an object to a given category. The strength of the MCCs could be seen along three dimensions: (i) integration of subjective information like the decision-maker knowledge and preferences, (ii) rigorous manipulation of heterogeneous, conflicting and non commensurable information,

and (iii) easy to explain, and therefore are not black boxes. The IEA is implemented using a mathematical model that provides automatically the "optimal" values of the NCD-based MCCs parameters. An ALM based on RCGA has been proposed to approximate its "optimal" solution and consequently to infer the parameters values of these classifiers because the proposed model could not be solved by classical optimization tools (e.g. gradient algorithms and interior-point algorithms). The proposed ALM overcomes some simplifications made in prior works (e.g. [4]): both concordance and discordance concepts are taken into account, the criteria weights are used in the computation of the membership degree of an object to a pre-defined category and finally each category may be characterized by many profiles.

A military dataset of 2545 Forward Looking Infra-Red (FLIR) images representing eight different classes of ships is used to test the performance of two NCD-based MCCs (PROAFTN classifier [3] and K-PIP classifier [15]) with respect four other classifiers (Dempster-Shafer-based (DS) classifier, Modified-Bayes-based (MB) classifier, K-Nearest Neighbors (k-NN) classifier and Neural Net (NN) classifier). The computational re-

TABLE IX
K-PIP Results

Problems	Genetic Algorithm		Population size	Selection Method	Crossover Step		Mutation Step	
	Training Database	Test database			Cross. Oper	Cross. Prob	Muta. Oper	Muta. Prob
Problem 1	83,38%	80,63%	30	RWS	FC	0,60	RUM	0,05
Problem 2	81,47%	79,84%	30	SRWRS	AC	0,65	NUM	0,07
Problem 3	82,31%	80,89%	30	LRS	SC	0,70	MM	0,09
Problem 4	82,54%	79,58%	40	TS	BLX	0,75	GM	0,10
Problem 5	84,31%	81,03%	40	RWS	ELC	0,80	RUM	0,05
Problem 6	83,25%	80,22%	40	SRWRS	FC	0,60	NUM	0,07
Problem 7	85,02%	82,41%	50	LRS	AC	0,65	MM	0,09
Problem 8	84,59%	81,19%	50	TS	SC	0,70	GM	0,10
Problem 9	83,69%	80,07%	50	RWS	BLX	0,75	RUM	0,05
Problem 10	82,58%	80,97%	60	SRWRS	ELC	0,80	NUM	0,07
Problem 11	85,17%	83,72%	60	LRS	FC	0,60	MM	0,09
Problem 12	80,77%	78,61%	60	TS	AC	0,65	GM	0,10
Problem 13	85,04%	82,51%	70	RWS	SC	0,70	RUM	0,05
Problem 14	82,54%	79,91%	70	SRWRS	BLX	0,75	NUM	0,07
Problem 15	84,00%	81,25%	70	LRS	ELC	0,80	MM	0,09
Problem 16	80,87%	77,88%	70	TS	FC	0,60	GM	0,10
Problem 17	83,36%	80,11%	80	RWS	AC	0,65	RUM	0,05
Problem 18	82,43%	80,81%	80	SRWRS	SC	0,70	NUM	0,07
Problem 19	85,65%	83,09%	80	LRS	BLX	0,75	MM	0,09
Problem 20	81,47%	79,17%	80	TS	ELC	0,80	GM	0,10
Mean	83,22%	80,69%						
Standard deviation	1,46%	1,45%						
Mean range	2,53%							

sults show that NCD-based MCCs provide AIRs better than those provided by MB and DS classifiers but worse than those obtained by K-NN and NN classifiers. Although NCD-based MCCs don't provide the best AIRs in this application, we believe that they are promising classifiers and merit to be further explored. Note that NCD-based MCCs are not optimized for this kind of dataset. In fact, if qualitative information and human judgment are introduced, we are confident that NCD-based MCCs will certainly outrank K-NN and NN classifiers. Moreover, NCD-based MCCs are not black boxes and all their results are automatically explained.

Many improvements could be made to enhance the AIRs of NCD-based MCCs like:

- Integrating the profiles and the criteria weights in the ALM;
- Using other improved versions of K-Means algorithm for the profile identification (e.g. Y-Means [13] or J-Means [14]);
- Integrating the profile weights in the membership degree computation;
- Combining the aggregation operators of different NCD-based MCCs;
- Using the concept of specified classifier, i.e. for the classification purpose we only use a subset of criteria that discriminate more between objects;

—Implementing a parallel version of NCD-based MCCs to reduce the computation time. For instance concordance and discordance indices may be computed simultaneously;

- Since Genetic Algorithms (GAs) are inefficient to exploit local information of solutions in each population, it will be benefit to integrate, in each iteration of the GA, a local search strategy (e.g. steepest descent strategy) to fine-tuning these solutions locally.

We uphold that cross fertilization of multi-criteria decision analysis and information fusion concepts could be beneficial to both domains. Multiple criteria classifiers are designed to consider human in the loop and to support human decision making.

REFERENCES

- [1] M. M. Ali and A. Törn
Population set based global optimization algorithms: Some modifications and numerical studies.
Computers and Operations Research, 31, (2004), 1703–1725.
- [2] N. P. Arhcer and S. Wang
Application of the back propagation neural networks algorithm with monotonicity constraints for two group classification problems.
Decision Sciences, 24 (1993), 60–75.

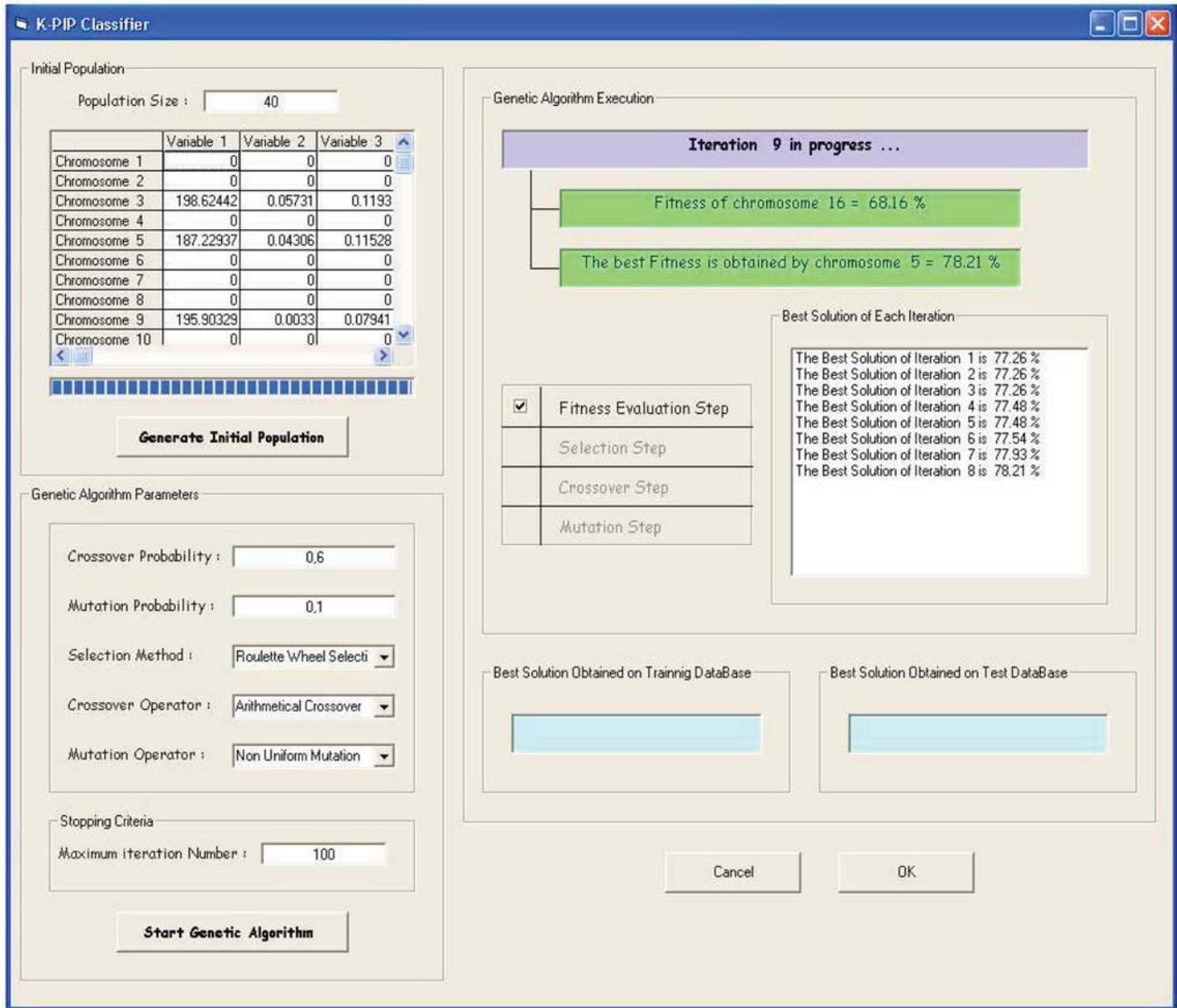


Fig. 7. Screen of the ALM for K-PIP classifier.

- [3] N. Belacel
Multicriteria assignment method PROAFTN: Methodology and medical application.
European Journal of Operational Research, 125 (2000), 175–183.
- [4] N. Belacel, H. B. Raval and A. P. Punnen
Learning multi-criteria fuzzy classification method PROAFTN from data.
Computer and Operations Research, 34 (2007), 1885–1898.
- [5] L. Benabbou, A. Guitouni and P. Lang
Méthode de classification: Revue de littérature, essai de caractérisation et de comparaison.
Document de travail 2004-023, (2004), Faculté des Sciences de l'Administration (FSA), Université Laval, Québec, Canada.
- [6] J-P. Brans and Ph. Vincke
A preference ranking organisation method: The PROMETHEE method for multiple criteria decision-making.
Management Science, 31 (1984), 647–656.
- [7] D. Bouyssou, T. Marchant, M. Pirlot, P. Perny, A. Tsoukias and Ph. Vincke
Evaluation and Decision Models: A Critical Perspective.
Boston/London/Dordrecht: Kluwer Academic Publishers, 2000.
- [8] L. C. Dias, V. Mousseau, J. Figueira and J. Climaco
An aggregation/disaggregation approach to obtain robust conclusions with ELECTRE TRI.
European Journal of Operational Research, 138 (2002), 332–348.
- [9] L. C. Dias and V. Mousseau
Inferring Electre's veto-related parameters from outranking examples.
European Journal of Operational Research, 170 (2006), 172–191.
- [10] M. Doumpos and C. Zopounidis
The use of the preference disaggregation analysis in the assessment of financial risks.
Fuzzy Economic Review, 3 (1998), 39–57.
- [11] J. Figueira, Y. De Smet and J-P. Brans
MCDA Methods for Sorting and Clustering Problems: PROMETHEE TRI and PROMETHEE CLUSTER.
20th European Conference on "Operational Research and the Management of Electronic Services," Rhodes Island, Greece, July 4–7, 2004.
- [12] D. Goldberg
Genetic Algorithm in Search.
Optimization and Machine Learning. Addison-Wesley, 1989.

- [13] Y. Guan, A. Ghorbani, N. Belacel and Y. Means
A Clustering Method for Intrusion Detection.
The Canadian Conference on Electrical and Computer Engineering, Montréal, Québec, Canada, May 4–7, 2003.
- [14] P. Hansen and N. Mladenovic
J-MEANS: A new local search heuristic for minimum sum of squares clustering.
Pattern Recognition Journal, 34 (2001), 405–413.
- [15] L. Henriot
Systèmes D'évaluation et de Classification Multicritère pour l'aide à la Décision.
Thèses de Doctorat, Université Paris Dauphine, France, 2000.
- [16] F. Herrera and M. Lozano
Two-loop real-coded genetic algorithms with adaptive control of mutation step sizes.
Applied Intelligence, 13 (2000), 187–204.
- [17] F. Herrera, M. Lozano and A. M. Sanchez
A Taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study.
International Journal of Intelligent Systems, 18 (2003), 309–338.
- [18] F. Herrera, M. Lozano and J. L. Verdegay
Tackling real-coded genetic algorithms: Operators and tools for behavioral analysis.
Artificial Intelligence Review, 12 (1998), 265–319.
- [19] J. H. Holland
Adaptation In Natural And Artificial Systems.
University of Michigan Press, 1975.
- [20] M. K. Hu
Visual pattern recognition by moment invariant.
IEEE Transactions on Information Theory, 8 (1962), 179–187.
- [21] E. Jacquet-Lagrèze
An application of the UTA discriminant model for the evaluation of R&D projects.
In P. M. Pardalos, Y. Siskos, C. Zopounidis (Ed.), *Advances in Multicriteria Analysis*, Dordrecht: Kluwer Academic Publishers, 1995, 203–211.
- [22] J. M. Keller, M. R. Gray and J. A. Givens
A fuzzy K-nearest neighbour algorithm.
IEEE Transactions on Systems, Man and Cybernetics, 15 (1985), 580–585.
- [23] J. Léger and J-M. Martel
A multicriteria assignment procedure for a nominal sorting problematic.
European Journal of Operational Research, 138 (2002), 349–364.
- [24] M. Massaglia and A. Ostanello
N-TOMIC: A decision support for multicriteria segmentation problems.
In P. Korhonen, (Ed.), *International Workshop on Multicriteria Decision Support, Lecture Notes in Economics and Mathematics Systems*, Berlin: Springer, 356 (1991), 167–174.
- [25] Z. Michalewicz
Genetic Algorithms+Data Structures=Evolution Programs.
Berlin: Springer-Verlag, 1992.
- [26] Z. Michalewicz
Genetic Algorithms+Data Structures=Evolution Programs.
Berlin: Springer-Verlag, 1996.
- [27] Z. Michalewicz, T. Logan and S. Swaminathan
Evolutionary operators for continuous convex parameters spaces.
In A. Sebald, L. Fogel (Ed.), *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, World Scientific Publishing, 1994, 84–107.
- [28] R. S. Michalski
A theory and methodology of inductive learning.
Artificial Intelligence, 20 (1983), 11–116.
- [29] N. Mladenovic and P. Hansen
Variable neighbourhood search: Principles and applications.
European Journal of Operational Research, 130 (2001), 449–467.
- [30] Y. Park and J. Sklansky
Automated design of linear tree classifiers.
Pattern Recognition, 23 (1990), 1393–1412.
- [31] P. Perny
Multicriteria filtering methods based on concordance and non-discordance principles.
Annals of Operations Research, 80 (1998), 137–165.
- [32] B. Roy
Classement et choix en présence de points de vue multiples (la méthode ELECTRE).
R.I.R.O. Recherche Opérationnelle, 2 (1968), 57–75.
- [33] B. Roy
Critères multiples et modélisation des préférences: l'apport des relations de surclassement.
Revue d'Économie Politique, 1 (1974).
- [34] B. Roy
ELECTRE III: Algorithme de classement basé sur une représentation des préférence en présence de critères multiples.
Cahiers du CERO, 21 (1978), 3–24.
- [35] B. Roy and J. Moscarola
Procédure automatique d'examen de dossiers fondée sur une segmentation trichotomique en présence de critères multiples.
RAIRO Recherche Opérationnelle, 11 (1977), 145–173.
- [36] Y. Siskos, E. Grigoroudis, C. Zopounidis and O. Saurais
Measuring customer satisfaction using a survey based preference disaggregation model.
Journal of Global Optimization, 12 (1998), 175–195.
- [37] P. Valin, F. Rhéaume, C. Tremblay, D. Grenier, A-L. Joussetme and E. Bossé
Comparative implementation of two fusion schemes for multiple complementary FLIR imagery classifiers.
Information Fusion Journal, 7 (2006), 197–206.
- [38] P. Vincke
L'aide multicritère à la décision.
Éditions de l'Université de Bruxelles, Belgique, 1989.
- [39] C. Zopounidis and M. Doumpos
A multicriteria decision aid methodology for sorting decision problems: The case of financial distress.
Computational Economics, 14 (1999), 197–218.
- [40] C. Zopounidis and M. Doumpos
Building additive utilities for multi-group hierarchical discrimination: The M.H.DIS method.
Optimization Methods and Software, 14 (2000), 219–240.
- [41] C. Zopounidis and M. Doumpos
Multicriteria classification and sorting methods: A literature review.
European Journal of Operational Research, 138 (2002), 229–246.
- [42] W. Yu
Aide Multicritère à la Décision Dans le Cadre de la Problématique du Tri: Concepts, Méthodes et Applications.
Thèse de doctorat, Université de Paris-Dauphine, France, 1992.
- [43] M. Zeleny
Multiple Criteria Decision Making.
McGraw-Hill, 1982.



Khaled Jabeur received his B.S. degree in computer engineering from Faculté des Sciences de Tunis in 1993. He received his Ph.D. and M.B.A. in operations and system decision from Université Laval in 2004 and 1997, respectively.

Dr. Jabeur is currently an assistant professor at l'Institut Supérieur de Commerce et de Comptabilité de Bizerte (ISCCB). His research interests include multicriteria decision analysis, operational research and group decision-making. He has published various papers in international journals (e.g. *European Journal of Operational Research*, *Group Decision and Negotiation*, *Information Systems and Operational Research*). In 2004, Dr. Jabeur has obtained the Governor General's Academic Gold Medal—which is the most prestigious award that students in Canadian universities can receive—to recognize his outstanding doctoral research.



A. Guitouni is a senior defence scientist with Defence R&D Canada–Valcartier. He specializes in decision support systems, operational research and artificial intelligence. His work includes planning and optimization, automated classification, decision analysis, multiple criteria decision aids, net-enabled dynamic resource management, and collaborative decision making (planning). He is an adjunct professor with Laval University and University of Victoria. He has designed and taught several graduate and undergraduate courses. He has supervised Ph.D., M.Sc. and MBA students.

A Critical Look at the PMHT

DAVID F. CROUSE
MARCO GUERRIERO
PETER WILLETT

We combine concepts from numerous papers to provide a derivation and description of a generalized Probabilistic Multi-Hypothesis Tracker that can track multiple targets in a cluttered environment, utilizing multiple sensors and feature measurements, if available. Additionally, we provide a full derivation of the algorithm, including parts omitted or abbreviated in other work. We also provide an improved analytic solution for the prior target-measurement probabilities conditioned on the number of observations, a simplified method of performing the maximization step of the algorithm when multiple sensors are used, a consistent covariance approximation of the algorithm when using multiple sensors, explore the use of deterministic annealing to improve performance, and discuss implementation difficulties.

Manuscript received November 11, 2008; revised April 3, 2009; released for publication April 8, 2009.

Refereeing of this contribution was handled by Yvo Boers.

This research was supported by the Office of Naval Research under contract N00014-07-1-0429, and by Vectrass, Inc.

We would like to thank Yaakov Bar-Shalom for his comments.

Authors' address: Department of Electrical and Computer Engineering, University of Connecticut, 371 Fairfield Way, U-2157, Storrs, Connecticut 06269, E-mail: ({crouse, marco.guerriero, willett}@engr.uconn.edu).

1557-6418/09/\$17.00 © 2009 JAIF

1. OVERVIEW

Since its creation by Streit and Luginbuhl in 1993 [60], much research has been done on the Probabilistic Multi-Hypothesis Tracker (PMHT). In this paper, we combine concepts from past works and provide a general version of the PMHT algorithm allowing for tracking in the presence of clutter (false alarms) and missed detections and the utilization of classification data, range rate information, and multiple synchronous sensors. This version makes no changes to the basis of the original algorithm, which is the Expectation Maximization (EM) algorithm. As a result, this generalized PMHT algorithm may be used as an improved foundation for other versions of the PMHT that build upon or alter the basis of the algorithm, such as the Multi-Frame Assignment PMHT (MFPMT) accounting for missed detections by Blanding, Willett, Streit, and Dunham [7]. Being a generalized version of the PMHT, the algorithm might be interchangeably named the PMHT or the Multi-Sensor PMHT (MSPMHT), in line with the naming convention of previous work.

In the subsequent sections, we derive the general form of the PMHT algorithm while discussing implementation difficulties. In Section 2, an overview of previous contributions to the algorithm is provided. Section 3 describes the EM algorithm, which forms the basis of the PMHT. Section 4 derives the state estimates within the PMHT, discussing implementation issues associated with precision problems in 4.3, how and why one might wish to include deterministic annealing to improve performance in 4.3, what to do if each sensor has a different field of view in 4.4, and out-of-order measurement delivery in 4.5. In Section 5, we compare the complexity of the PMHT against that of the Joint Probability Data Association Filter (JPDAF), which is a popular non-batch tracking algorithm.¹ We describe the conditions under which the PMHT has a lower complexity than the MSJPDAF. Section 6 explains how the PMHT, a batch algorithm, can be used over data sequences longer than the batch. Section 7 then discusses state covariance estimation in the PMHT. The algorithm is summarized in Section 8. Section 9 provides a simulation of the PMHT with multiple sensors to verify that the covariance estimation procedure of Section 7 provides, under certain conditions, consistent estimates when multiple targets are used, and demonstrates that deterministic annealing can significantly improve tracker performance when using multiple sensors. Section 10 summarizes the paper. The appendices provide derivations of the target-measurement association probabilities conditioned on the observations, and the target-measurement association probabilities conditioned only on the number of observations.

¹It is also referred to as the Multi-Sensor JPDAF (MSJPDAF) in the multisensor case.

2. PREVIOUS WORK ON THE PMHT ALGORITHM

The Probabilistic Multi-Hypothesis Tracker (PMHT) is a linear-complexity, EM algorithm based, batch target tracking algorithm for use in tracking multiple targets in the presence of clutter when the target-measurement associations are unknown.² Having a memory of the last N scans of data, it attempts to find the maximum a posteriori estimate of the target state in the current scan. This is similar to a later algorithm by Pulford and Logothetis [50], which estimates the target-measurement associations under different measurement models. Being a batch algorithm, it can easily handle delayed measurements, which may simply be added to the batch when they arrive, as was mentioned by Efe, Ruan and Willett [19]. For use as a practical tracker, which must produce track estimates before receiving a full N scans of data, the PMHT can be run at each step on a growing window until a full N scans of data have been acquired, at which point the window slides. This growing and sliding window has been shown to be more effective than other methods by Willett, Ruan, and Streit [70].

The first EM based tracking algorithm was a maximum-likelihood (ML) batch algorithm by Avitzour [2]. This tracker had a high complexity, requiring the calculation of all target-measurement association probabilities. A later algorithm by Molnar and Modestino [41] was a non-batch EM approach to tracking that calculates the maximum a posteriori (MAP) estimate and uses a Markov random field to model the target-measurement associations, resulting in a significantly lower complexity. Jeong and Park [31] used an alternative version of the EM algorithm to produce a recursive MAP target tracker that also estimates various parameters, reducing its complexity by approximating the joint association event probabilities reducing its complexity by approximating multiple target joint association event probabilities as products of single target joint association event probabilities. Pulford and La Scala [49] used the EM algorithm coupled with the Viterbi algorithm to estimate target maneuvers.

In contrast, the PMHT uses an arguably incorrect measurement model in order to reduce its complexity. That is, when told that a particular measurement originated from a particular target, it ignores any conditioning this may imply when determining the posterior association probabilities of the other measurements. As a result, the probability of a particular measurement coming from a particular target is independent of whether that or any other target produced any of the other measurements, and each target is allowed to produce any number of measurements. However, each measurement can only originate from a single target, which is realistic when the targets are well resolved. A version of the PMHT accounting for unresolved targets has also been developed by Davey [11].

It should be noted that the PMHT is not the only algorithm utilizing an “incorrect” measurement model. Particle filter based trackers by Hue, Le Cadre, and Pérez [27], [28] as well as by Gilholm, Godsill, Maskell, and Salmond [25] utilize the same model. The measurement model has some appeal when high resolution sensors are able to over-resolve the target.

The PMHT was first proposed by Streit and Luginbuhl in 1993 [60] with the first full statement of the algorithm appearing in a Naval technical report two years later [61]. The PMHT algorithm was defined very generally in [61], allowing for the target dynamics and measurement model to have arbitrary distributions. However, such a generic model did not allow for the maximization step of the EM algorithm, upon which it is based, to be easily performed, and thus the practical implementation presented had a discrete-time linear motion model, as used in the Kalman filter. Since then, quite a few variants of the PMHT algorithm have emerged, many of which have been compared by Willett, Ruan, and Streit in [71].

The best performing variants of the PMHT, that is those having better track-loss characteristics than the JPDAF, are the Turbo PMHT, by Ruan and Willett [56] and Willett, Ruan, and Streit [69] and Multi-Frame PMHT (MFPMHT) algorithms. The currently best-performing version of the PMHT is the MFPMHT that accounts for missed detections, by Blanding, Willett, Streit, and Dunham [7], which is a modification of an earlier MFPMHT by Streit [59]. However, the better performance of the MFPMHTs comes with an increased complexity, being roughly exponentially complex over the last L frames of an N frame batch.

The homeothetic PMHT, first derived by Rago, Willett, and Streit in [53], was an *ad-hoc* approach to improving the performance of the PMHT through the use of multiple measurement models with different noise covariances. The different covariances were intended to overcome estimation problems in the PMHT and not to function as multiple models for states inherent to the targets. However, this may be thought of as a forerunner to multiple model PMHT algorithms. Logothetis, Krishnamurthy, and Holst [35]³ were the first to develop a form of the PMHT algorithm involving multiple models to account for target maneuvers. The transition between model states was governed by a Hidden Markov Model (HMM). The maneuvers were handled using additional unknown “control” inputs to the Kalman filter. Willett, Ruan, and Streit [69] also did this, modeling the maneuvers as increases in the process noise of the targets forming the MPMHT. In both approaches, the active model of the target was modeled as one of the “nuisance” variables in the EM algorithm. Pulford and La Scala [49] took a different approach, making the maneuvers part of the quantity to be estimated in the EM algorithm. Their maneuver-estimation approach can be used

²A though the focus is on target tracking, the PMHT algorithm has found use in other applications, such as cartography [10].

³This is the journal version of an earlier conference paper [34].

with various EM-algorithm based trackers including the PMHT. Willett, Ruan, and Streit created an interactive multiple model approach to tracking using the PMHT in the presence of maneuvers (IMM-PMHT), replacing the backward-forward algorithm used in [69] with an interacting multiple model (IMM). They also derived a turbo-coding based extension to the MPMHT, which they dubbed the Turbo MPMHT. This later matured into the Turbo PMHT, as described by Ruan and Willett [56]. More recently, Luginbuhl, Ainsleigh, Mathews, and Streit [36] demonstrated how to derive the observed data likelihood function for the family of manoeuvring PMHT trackers.

The basic PMHT is an algorithm that tracks targets based upon discrete observations at each scan. However, a variant called the Histogram PMHT (H-PMHT) allows the PMHT to process continuous data directly from a sensor. The concept was first introduced by Luginbuhl and Willett [37] as a method of tracking a general frequency-modulated signal in noise (explained in more detail in [38]), and a variant appeared in [58]. Walsh, Graham, Streit, Luginbuhl, and Mathews [64] later presented a one-dimensional application of the algorithm. Pakfiliz and Efe [44] presented a two-dimensional application, and most recently, Davey, Rutten, and Cheung [15] compared it against other track-before-detect methods. In this paper, we do not consider the processing of continuous sensor measurements. The use of the PMHT in tracking problems with bearing-only measurements has been studied by Giannopoulos, Streit, and Swaszek [23].

In its original version, the PMHT did not account for clutter. Rago, Willett, and Streit [52] extended the PMHT to cluttered environments under a number of assumptions, regarding the probability of a measurement originating from clutter, by modifying the target measurement assignment probabilities $w_{k_r(t),l}(t)$. In the next year, Hutchins and Dunham produced a similar version of the PMHT for use in cluttered environments [29], involving an *ad-hoc* constant in the denominator of the target measurement assignment probabilities. In later publications, an analytically derived solution has been used, but no complete derivation has been given. In this paper, we provide an explicit derivation of the PMHT including clutter, and we provide a full Bayesian derivation of the prior and posterior association probabilities $\pi_{k_r(t)}(n_t)$ and $w_{k_r(t),l}(t)$. The probabilities $\pi_{k_r(t)}(n_t)$ have previously been derived by Wieneke and Koch [66], but here it is developed in such a way that the solution could be simplified by omitting “fictitious targets” that had been used in [66].

Davey, Gray, and Streit [14] introduced the use of target classification measurements into the PMHT. Namely, extra data can be used to identify the type of each observations, e.g., whether an observation is clutter, a plane or a missile. A more complete analysis of this work is given in Davey’s PhD thesis [9]. In this paper, we show how classification measurements can be used in a multisensor environment.

The simplest approach to multisensor tracking with the PMHT was first considered by Rago, Willett, and Streit [51]. They pooled all of the measurements from all of the sensors together and ran the PMHT as if all of the measurements came from a single sensor. One can justify this by the fact that the PMHT’s measurement model allows for a single target to produce multiple observations. Hempel [26] considered the robustness of the PMHT to registration errors when the measurements from all sensors are pooled. However, versions of the PMHT specifically accounting for multiple sensors by modifying the likelihood function to reflect their presence have been developed, and have been shown to improve the performance of the tracker over the pooled measurement approach. These were developed concurrently by Krieg and Gray [33], by Giannopoulos, Streit, and Swaszek [24] and by Gauvrit, Le Cadre, and Jaufret [22]. All of these derivations used the Levenberg-Marquadt method (described, for example in [5]) for performing the maximization step of the EM algorithm. In this paper, we show that a simpler, non-iterative approach exists.

In its original form, the PMHT algorithm was meant to track a known number of targets and lacked any notion of track discovery, termination or merging. However, such tasks are necessary for a tracker to be usable in real-world situations. Several advances have been made in integrating track discovery and termination features into the PMHT. A complete track management system, in which tracks were discovered and terminated by separate algorithms outside of the PMHT algorithm, was first introduced by Luginbuhl, Sun, and Willett [39], whereby track extraction was done via the Hough transform. Alexiev [1] also considered the use of the Hough transform with the PMHT. Davey and Gray [12] later gave a comparison of various methods of track initiation, and Dunham and Hutchins [17] considered using the MHT as a track-finding front-end to the PMHT, whereby once tracks were stable they were handed off to the PMHT. Since then, however, additional methods of track management have emerged. Davey and Gray introduced the Hysteresis PMHT [13], which treats the existence of a target as an extra state in the estimator. Mušicki and Wang [43] used an *ad-hoc* approach of modifying the posterior association probabilities to do the same thing. Wieneke and Willett [68] looked at methods of determining track deletion and Wieneke and Koch looked at hypothesis tests for estimating the number of tracks present [67].

3. THE EXPECTATION MAXIMIZATION ALGORITHM AND DETERMINISTIC ANNEALING

The PMHT is based on the EM algorithm. The EM algorithm, discovered by Dempster, Laird, and Rubin [16], is a method of determining the ML or MAP estimate of data given incomplete information. Redner and Walker [54] specifically looked at the use of the

EM algorithm for ML estimation of parameters of mixture densities, a topic that is relevant to EM algorithm based target tracking. The EM algorithm is summarized here, which is extensively covered in the monograph by McLachlan and Krishnen [40], and we must note the tutorial by Moon in [42]. It should be noted that the EM algorithm does not provide the covariance of its estimate. This is, however, necessary for the tracker to be useful and is discussed in Section 7.

Let \mathbf{X} be an unknown random quantity the MAP estimate of which we would like to find. Let \mathbf{Z} be the set of observations, which are dependent upon \mathbf{X} and a set of unobservable random variables \mathbf{K} . We would like to find the MAP estimate of \mathbf{X} without having to determine \mathbf{K} , which might be a difficult or computationally complex task. The MAP estimate of \mathbf{X} may be expressed as

$$\hat{\mathbf{X}}_{\text{MAP}} = \arg \max_{\mathbf{X}} E\{\log(p(\mathbf{X} | \mathbf{Z}))\}, \quad (1)$$

in which p represents a probability density function (PDF). The expectation comes from the Law of Total Probability eliminating the unobservable random variable \mathbf{K} . However, in many cases the expectation may be difficult to evaluate. The EM algorithm avoids direct computation of this expectation. Define the following function:

$$Q(\mathbf{X}^{(n+1)}; \mathbf{X}^{(n)}) \triangleq \int_{\mathbf{K}} \log(p(\mathbf{X}^{(n+1)}, \mathbf{K} | \mathbf{Z})) p(\mathbf{K} | \mathbf{X}^{(n)}, \mathbf{Z}) d\mathbf{K}. \quad (2)$$

The integration in (2) is defined over whichever measure is appropriate for \mathbf{K} , which may be discrete. The EM algorithm is as follows: in each step, $\mathbf{X}^{(n+1)}$ is found as

$$\mathbf{X}^{(n+1)} = \arg \max_{\mathbf{X}^{(n+1)}} Q(\mathbf{X}^{(n+1)}; \mathbf{X}^{(n)}). \quad (3)$$

n is then incremented and one continues until a desired level of convergence has been attained.

In some instances, the PDF $p(\mathbf{X}^{(n+1)}, \mathbf{K} | \mathbf{Z})$ may be difficult to determine. If this is the case, by the definition of conditional expectation, it can be noted that

$$p(\mathbf{X}, \mathbf{K} | \mathbf{Z}) = \frac{p(\mathbf{Z}, \mathbf{X}, \mathbf{K})}{p(\mathbf{Z})}. \quad (4)$$

Substituting (4) into (2) and separating the logarithm we get

$$Q(\mathbf{X}^{(n+1)}; \mathbf{X}^{(n)}) = \int_{\mathbf{K}} \log(p(\mathbf{Z}, \mathbf{X}^{(n+1)}, \mathbf{K})) p(\mathbf{K} | \mathbf{X}^{(n)}, \mathbf{Z}) d\mathbf{K} - \log p(\mathbf{Z}). \quad (5)$$

Because $p(\mathbf{Z})$ is a constant, that term may be dropped from (5), since it has no effect on the location of the maximum and thus $p(\mathbf{X}^{(n+1)}, \mathbf{K} | \mathbf{Z})$ and $p(\mathbf{Z}, \mathbf{X}^{(n+1)}, \mathbf{K})$ may be used interchangeably in the EM algorithm.

Boyles [8] and Wu [30] studied the convergence properties of the EM algorithm, correcting a mistake

in Theorem 2 of Dempster, Laird, and Rubin's original paper [16]. They showed that the EM algorithm is guaranteed to converge to a saddle point or a local maximum, which need not be the desired global maximum. To which critical point it converges is highly dependent upon the initial estimate $\mathbf{X}^{(1)}$.

Over the years, a number of versions of the EM algorithm have been developed, many of which are summarized by McLachlan and Krishnen [40] and by Roche [55]. Most sought to increase the convergence speed of the algorithm. However, for the PMHT algorithm, the primary concern is avoiding convergence to local maxima. In order to reduce dependence on the initial estimate and encourage convergence to the global maximum, the Deterministic Annealing (DA) EM algorithm was developed by Ueda and Nakano [62], who recognized that solving the maximum likelihood problem is analogous to similar problems linking concepts in thermodynamics and information theory. This was applied to the PMHT first in 1999 by Strandlie and Zerubia [57] and was later applied in a more general form by Wieneke and Koch [66]. When tracking a single target, the basic PMHT algorithm with deterministic annealing is identical to the Deterministic Annealing Filter by Frühwirth and Strandlie [20]. As shall be described after the basic derivation of the PMHT, deterministic annealing can be added to almost any version of the PMHT algorithm. To derive the DA-EM algorithm, we shall use the definition of conditional expectation to note that

$$p(\mathbf{K} | \mathbf{X}^{(n)}, \mathbf{Z}) = \frac{p(\mathbf{Z}, \mathbf{K}, \mathbf{X}^{(n)})}{\int_{\mathbf{K}_1} p(\mathbf{Z}, \mathbf{K}_1, \mathbf{X}^{(n)}) d\mathbf{K}_1}. \quad (6)$$

The denominator in (6) is equal to $p(\mathbf{Z}, \mathbf{X})$. The DA-EM algorithm substitutes (6) into (2) and introduces the term β as

$$Q_{\text{DA}}(\mathbf{X}^{(n+1)}; \mathbf{X}^{(n)}) \triangleq \int_{\mathbf{K}} \log(p(\mathbf{X}^{(n+1)}, \mathbf{K} | \mathbf{Z})) \frac{p(\mathbf{Z}, \mathbf{K}, \mathbf{X}^{(n)})^\beta}{\int_{\mathbf{K}_1} p(\mathbf{Z}, \mathbf{K}_1, \mathbf{X}^{(n)})^\beta d\mathbf{K}_1} d\mathbf{K}. \quad (7)$$

In the original description of the DA-EM algorithm, β , the inverse of which corresponds to the "temperature" in an analogous thermodynamic problem, is initially set to a value between 0 and 1. Q_{DA} is then iterated with respect to $\mathbf{X}^{(n+1)}$ and $\mathbf{X}^{(n)}$ until convergence, as is done in the regular EM algorithm. Then β is increased to a value closer to one and Q_{DA} is again iterated until convergence. The DA-EM algorithm is complete when β has finally been increased to 1. Note however, that the original version of the DA-EM algorithm did not specify exactly how β was to be increased.

When $\beta < 1$, the PDF $p(\mathbf{K} | \mathbf{X}^{(n)}, \mathbf{Z})$ becomes flatter, which reduces dependence of the algorithm on $\mathbf{X}^{(n)}$. The reasoning behind the DA-EM algorithm is that by slowly increasing β , the effect of $p(\mathbf{K} | \mathbf{X}^{(n)}, \mathbf{Z})$ is increased at the same time that the estimate $\mathbf{X}^{(n)}$ improves.

In the final step $\beta = 1$ and (7) is equivalent to (2) and the EM algorithm should be more likely to converge to the global MAP estimate, because of the improved prior estimate $\mathbf{X}^{(n)}$.

In order to use the DA-EM algorithm in a practical implementation, one must have a method of increasing β . If β is increased very slowly, then in general, one iteration should be enough for convergence at each value of β . Thus, in [57] and [66], the DA-EM algorithm was carried out as follows:

Let n_{\max} be the number of iterations that one wishes to do. For each iteration from $n = 1$ onwards, set $\beta = n/n_{\max}$. Now iterate the EM algorithm as would normally be done. That is, \mathbf{X}^{n+1} is the set to the value maximizing $Q_{\text{DA}}(\mathbf{X}^{(n+1)}; \mathbf{X}^{(n)})$ at each step. In the final iteration $\beta = 1$ and the result is the EM algorithm result.

Although convergence to the global MAP estimate is not guaranteed, as long as n_{\max} is large enough, this approach will generally outperform the basic EM algorithm. In Section 9, we demonstrate how deterministic annealing improves tracking performance when multiple sensors are used. Although the above method is the procedure needed to get the MAP estimate, the ML estimate can be attained by replacing $p(\mathbf{X}^{(n+1)}, \mathbf{K} | \mathbf{Z})$ with $p(\mathbf{Z}, \mathbf{K} | \mathbf{X}^{(n+1)})$.

4. THE PMHT ALGORITHM: STATE ESTIMATES

We shall now derive a general form of the PMHT algorithm allowing for the presence of clutter, multiple synchronous sensors and the use of classification information. The most general form of the PMHT allows for a very generic target motion and measurement model [61]. However, due to the difficulty of the maximization step of the EM algorithm under a generic model, practical implementations of the PMHT are often based upon the motion models that, in the absence of target-measurement association uncertainty, contain the assumptions inherent to the basic Kalman filter (see, e.g., [4]). We shall derive the PMHT under such a model, accounting for clutter, taking advantage of multiple sensors, and utilizing classification data. It shall be assumed that the measurement noise between sensors is uncorrelated and that sensors have the same field of view. The case where the sensors have different fields of view and gating is present, is discussed in the following section.

Given M targets, the state vector at time t for the m th target shall be designated as \mathbf{x}_m . The observation originating from the m th target shall be designated $\mathbf{y}_m(t)$. The basic discrete-time kinematic motion and observations equations are given by⁴

$$\mathbf{x}_m(t+1) = \mathbf{F}_m(t)\mathbf{x}_m(t) + \mathbf{v}_m(t) \quad (8)$$

⁴The Kalman filter and Kalman smoother equations associated with this model are summarized in Appendix D.

and

$$\mathbf{y}_m(t) = \mathbf{H}_m(t)\mathbf{x}_m(t) + \mathbf{w}_m(t). \quad (9)$$

The process noise at time t , $\mathbf{v}_m(t)$, is assumed to be Gaussian distributed with zero mean and covariance $\mathbf{Q}_m(t)$. The measurement noise $\mathbf{w}_m(t)$ is also modeled as a zero-mean Gaussian random variable with covariance $\mathbf{R}_m(t)$ and is assumed to be uncorrelated with the process noise. The covariance of the true measurement from the m th target $\mathbf{R}_m(t)$ describing $\mathbf{w}_m(t)$ from (9), corresponds to the covariance of one of the measurements out of the set of all measurements at time t , whereby $\mathbf{R}_{r,s}(t)$ shall represent the covariance of measurement r from sensor s based upon the location of the observation, without stating a particular associated target.

Let \mathbf{Z} be all of the measurements and classification information from time $t = 1$ to N . Let \mathbf{X} be the states of all of the targets over the same time period and \mathbf{K} be the set of associations between targets and measurements. Let there be a total of S sensors that take measurements synchronously. If $\mathbf{z}_{r,s}(t)$ is the measurement r at time t from sensor s that came from target m , then we shall denote said association by $k_{r,s}(t) = m$. We would like to use the EM algorithm to estimate \mathbf{X} without explicitly determining which set of $k_{r,s}(t)$ from the set of all possible target to measurement associations, \mathbf{K} , is correct. We shall consider clutter to be target $m = 0$.

The inclusion of classification measurements in the PMHT was first discussed by Davey, Gray, and Streit [14]. We shall assume that some type of classification has already been done for each measurement, giving us $z_{r,s}^C(t)$, the classification data associated with measurement r from sensor s at time t . Including classification in the PMHT means estimating the type of each target. This is done via a confusion matrix \mathbf{C} whose elements are defined as

$$c(i, m) = \Pr(z_{r,s}^C(t) = i | k_{r,s}(t) = m). \quad (10)$$

i in (10) represents the i th classification out of the set of all M_C possible classifications. The true classification of each target is assumed to be time-invariant, which is why $c_{i,m}$ is not indexed against time.⁵ That is, the appearance of each target is assumed to be constant. It shall also be assumed independent of that state. The confusion matrix is the estimated probability that a target or a clutter measurement has a certain associated appearance. The confusion matrix shall be estimated along with \mathbf{X} in the PMHT algorithm. Thus, $\arg \max_i c_{i,m}$ will be the MAP estimate of the classification of target m at the end of the algorithm.

Let us find the first PDF in (5), including \mathbf{C} next to \mathbf{X} as an unknown to be estimated. Let $n_t(s)$ be the number of measurements at time t that came from

⁵This assumption plays a role in our subsequent estimation of the confusion matrix via the EM algorithm as part of the PMHT. However, if one does not wish to perform EM algorithmic estimation of the confusion matrix, then a time-varying confusion matrix may be used without modification to the rest of the PMHT.

sensor s . In order for $p(\mathbf{Z}, \mathbf{X}, \mathbf{K}, \mathbf{C})$ to be written, it shall be conditioned on $n_t(s)$. However, we will not explicitly write this conditioning except when necessary. $p(\mathbf{Z}, \mathbf{X}, \mathbf{K}, \mathbf{C})$ is given as

$$\begin{aligned}
p(\mathbf{Z}, \mathbf{X}, \mathbf{K}, \mathbf{C}) &= p(\mathbf{X})p(\mathbf{Z}, \mathbf{K}, \mathbf{C} | \mathbf{X}) \quad (11a) \\
&= \overbrace{\left(\prod_{m=1}^M p(\mathbf{x}_m(1)) \prod_{t_x=2}^N p(\mathbf{x}_m(t_x) | \mathbf{x}_m(t_x - 1)) \right)}^{p(\mathbf{X})} \\
&\quad \times \overbrace{\prod_{s=1}^S \prod_{t=1}^N \prod_{r=1}^{n_t(s)} \Pr(k_{r,s}(t) | \mathbf{x}_{k_{r,s}(t)}(t), n_t) p(\mathbf{z}_{r,s}(t) | k_{r,s}(t), \mathbf{x}_{k_{r,s}(t)}(t))}^{p(\mathbf{Z}, \mathbf{K} | \mathbf{X})} \\
&\quad \times \underbrace{c(\mathbf{z}_{r,s}^C(t), k_{r,s}(t))}_{p(\mathbf{C} | \mathbf{Z}, \mathbf{X}, \mathbf{K})}. \quad (11b)
\end{aligned}$$

In (11b), the PDF $p(\mathbf{z}_{r,s}(t) | k_{r,s}(t), \mathbf{x}_{k_{r,s}(t)}(t))$ depends upon whether the measurement came from clutter or from a target and is given by

$$\begin{aligned}
p(\mathbf{z}_{r,s}(t) | k_{r,s}(t), \mathbf{x}_{k_{r,s}(t)}(t)) &= \begin{cases} \mu(t, \mathbf{z}_{r,s}(t)) & \text{if } k_{r,s}(t) = 0 \\ \mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_{k_{r,s}(t)}(t), \mathbf{R}_{r,s}(t)\} & \text{if } k_{r,s}(t) \neq 0 \end{cases}. \quad (12)
\end{aligned}$$

In (12), μ denoted the PDF of the clutter, which we shall assume to be continuous as a function of $\mathbf{z}_{r,s}(t)$, and which need not be uniform, and $\hat{\mathbf{y}}_{k_{r,s}(t)}(t)$ is the estimate of \mathbf{y} from (9). That is,

$$\hat{\mathbf{y}}_{k_{r,s}(t)}(t) = \mathbf{H}_{k_{r,s}(t)}(t) \mathbf{x}_{k_{r,s}(t)}(t). \quad (13)$$

Define $w_{k_{r,s}(t),r}(t,s) \triangleq \Pr(k_{r,s}(t) | \mathbf{x}_{k_{r,s}(t)}(t), \mathbf{Z}(t), \mathbf{C}, n_t(s))$ as the probability of a particular measurement-target assignment at time t , whereby clutter is target $k_{r,s}(t) = 0$. We shall refer to these as the ‘‘posterior association probabilities.’’ One instance of \mathbf{K} defines $k_{r,s}(t)$ over all

Under the basic PMHT assumption, because each target can produce more than one measurement, all of the values of $w_{k_{r,s}(t),r}(t,s)$ at a particular time are independent. Additionally, because the current state and observation set are given, the values of $w_{k_{r,s}(t),r}(t,s)$ are also independent as a function of time. Because of this independence, $p(\mathbf{K} | \mathbf{X}, \mathbf{C}, \mathbf{Z})$, the second PDF in (5), may be obtained directly by multiplying the marginal probabilities over all time and measurements for all of the assignments:

$$p(\mathbf{K} | \mathbf{X}, \mathbf{C}, \mathbf{Z}) = \prod_{s=1}^S \prod_{t=1}^N \prod_{r=1}^{n_t(s)} w_{k_{r,s}(t),r}(t,s). \quad (15)$$

In order to make the notation in the above equation correct if there are no observations at a particular sensor at a certain time, that is if $n_t(s) = 0$, the following definition must be used:

$$\prod_{r=1}^0 w_{k_{r,s}(t),r}(t,s) \triangleq 1. \quad (16)$$

We would now like to determine the posterior association probability $w_{k_{r,s}(t),r}(t,s)$. In previous published works on the PMHT, no formal derivation of this in the presence of clutter has been done and because it is not immediately obvious, it shall be included here for completeness, in Appendix A. Define $\pi_{k_{r,s}(t)}(n_t(s), t) \triangleq \Pr(k_{r,s}(t) = m | \mathbf{x}_m(t), n_t(s))$. This shall be referred to as a ‘‘prior association probability.’’⁶ The $\pi_{k_{r,s}(t)}(n_t(s), t)$ values were derived using ‘‘imaginary’’ targets by Wieneke and Koch [66] and we have re-derived them in a simpler form in Appendix B. The original PMHT algorithm made them a parameter to be estimated by the EM algorithm.

Using the solution from Appendix A for $k_{r,s}(t) \neq 0$, that is when measurement r from sensor s is not clutter, then the posterior association probabilities for non-clutter targets are as follows (in the final solution to the PMHT algorithm, it will turn out that one never needs to evaluate $w_{0,r}(t,s)$):

$$w_{k_{r,s}(t),r}(t,s) = \frac{\pi_{k_{r,s}(t)}(n_t(s), t) \mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_{k_{r,s}(t)}(t), \mathbf{R}_{r,s}(t)\} c(\mathbf{z}_{r,s}^C(t), k_{r,s}(t))}{\pi_0(n_t(s), t) \mu(t, \mathbf{z}_{r,s}(t)) c(\mathbf{z}_{r,s}^C(t), 0) + \sum_{m=1}^M \pi_m(n_t(s), t) \mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_m(t), \mathbf{R}_{r,s}(t)\} c(\mathbf{z}_{r,s}^C(t), m)}. \quad (17)$$

measurements r and sensors s , for all time in the batch. The sum over \mathbf{K} is equal to the sum over all sensors of the sum over \mathbf{K}_s , which is defined as

$$\sum_{\mathbf{K}_s} (\cdot) = \sum_{k_{1,s}(1)=0}^M \sum_{k_{2,s}(1)=0}^M \cdots \sum_{k_{n_1(s),s}(1)=0}^M \sum_{k_{1,s}(2)=0}^M \cdots \sum_{k_{n_N(s),s}(N)=0}^M (\cdot). \quad (14)$$

In a simple model where all targets have the same probability of detection, $P_D(s)$, when viewed by sensor s , and the number of clutter points at each sensor

⁶‘‘prior association probability’’ is somewhat of a misnomer, since $\pi_{k_{r,s}(t)}(n_t(s), t)$ is conditioned on the number of observations. However, this naming convention helps differentiate it from the posterior association probabilities.

is Poisson distributed with mean $\lambda(s)V(s)$ where $\lambda(s)$ represents the mean amount of clutter per unit volume at sensor s and $V(s)$ is the volume of the viewing area for that sensor, then, as derived in the Appendices B and C, the prior association probabilities are the same for all non-clutter targets and may be divided across the numerator and denominator giving us the following expression for $w_{k_{r,s}(t),r}(t,s)$

$$w_{k_{r,s}(t),r}(t,s) = \frac{\mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_{k_{r,s}(t)}(t), \mathbf{R}_{r,s}(t)\} c(z_{r,s}^C(t), k_{r,s}(t))}{\bar{\pi}_s(n_t(s), t) \mu(t, \mathbf{z}_{r,s}(t)) c(z_{r,s}^C(t), 0) + \sum_{m=1}^M \mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_m(t), \mathbf{R}_{r,s}(t)\} c(z_{r,s}^C(t), m)}, \quad (18)$$

$$\bar{\pi}_s(n_t(s), t) = -M - \frac{{}_2F_0 \left[-M, -n_t(s); \frac{P_D(s)}{(1-P_D(s))\lambda(s)V(s)} \right]}{{}_2F_0 \left[1-M, 1-n_t(s); \frac{P_D(s)}{(1-P_D(s))\lambda(s)V(s)} \right]}, \quad (19)$$

here the function ${}_2F_0[a_1, a_2; z]$ is a hypergeometric function.

Combining (15) and (11b) and omitting the constant $p(\mathbf{Z})$ we may form the Q function for the basic EM algorithm in (5):

Equation (21) comes from the fact that at any given time the observations must have originated from a target or from clutter. $\mathbf{K}_s \setminus k_{r_1,s}(t_i)$ represents the set of all assignments involving sensor s except for $k_{r_1,s}(t_i)$ and (22) comes directly from the definition of $w_{k_{r,s}(t),r}^n(t,s)$. Note that when $k_{r,s}(t) = 0$, that is when observation r is clutter, $p(\mathbf{z}_{r,s}(t) | k_{r,s}(t), \mathbf{x}_{k_{r,s}(t)}(t))$ contains no terms

involving \mathbf{X} . Therefore, for purposes of maximizing Q , we may omit all clutter terms from the second set of sums, because they disappear when the derivative is taken. Using this fact and (21) and (22), Equation (20b) may be simplified as follows:

$$Q(\mathbf{X}^{n+1}, \mathbf{C}^{n+1}; \mathbf{X}^n, \mathbf{C}^n) = \sum_{\mathbf{K}} \log(p(\mathbf{Z}, \mathbf{X}^{(n+1)}, \mathbf{C}^{(n+1)}, \mathbf{K})) p(\mathbf{K} | \mathbf{X}^n, \mathbf{C}^n, \mathbf{Z}) \quad (20a)$$

$$\begin{aligned} &= \log \left(\prod_{m=1}^M p(\mathbf{x}_m^{(n+1)}(1)) \prod_{t_x=2}^N p(\mathbf{x}_m^{(n+1)}(t_x) | \mathbf{x}_m^{(n+1)}(t_x - 1)) \right) \\ &+ \sum_{\mathbf{K}} \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_t(s)} \log(\pi_{k_{r,s}(t)}(n_t(s), t)) \prod_{s_1=1}^S \prod_{t_1=0}^N \prod_{r_1=1}^{n_{t_1}(s_1)} w_{k_{r_1,s_1}(t_1),r_1}^{(n)}(t_1, s_1) \\ &+ \sum_{\mathbf{K}} \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_t(s)} \log(c(z_{r,s}^C(t), k_{r,s}(t))) \prod_{s_1=1}^S \prod_{t_1=0}^N \prod_{r_1=1}^{n_{t_1}(s_1)} w_{k_{r_1,s_1}(t_1),r_1}^{(n)}(t_1, s_1) \\ &+ \sum_{\mathbf{K}} \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_t(s)} \log(p(\mathbf{z}_{r,s}(t) | k_r(t), \mathbf{x}_{k_{r,s}(t)}^{(n+1)}(t))) \prod_{s_1=1}^S \prod_{t_1=0}^N \prod_{r_1=1}^{n_{t_1}(s_1)} w_{k_{r_1,s_1}(t_1),r_1}^{(n)}(t_1, s_1). \quad (20b) \end{aligned}$$

The superscripts in parentheses in (20b) indicate whether the values in question are to be calculated using the current or the previous estimate of \mathbf{X} in the EM algorithm. As pointed out by Davey [9] in his thesis, (20b) is simplified by use of the two identities

$$\sum_{\mathbf{K}_s} \prod_{t=1}^N \prod_{r=1}^{n_t} w_{k_{r,s}(t),r}(t,s) = 1 \quad (21)$$

and

$$\sum_{\mathbf{K}_s \setminus k_{r_1,s}(t_i)} \prod_{t=1}^N \prod_{r=1}^{n_t} w_{k_{r,s}(t),r}(t,s) = w_{k_{r_1,s}(t_i),r_1}(t_i, s). \quad (22)$$

$$Q(\mathbf{X}^{n+1}; \mathbf{X}^n)$$

$$\begin{aligned} &= \left. \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_t(s)} \sum_{m=1}^M \log(\pi_m(n_t(s), t)) w_{m,r}^{(n)}(t, s) \right\} Q_{\Pi} \\ &+ \left. \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_t(s)} \sum_{m=1}^M \log(c(z_{r,s}^C(t), m)) w_{m,r}^{(n)}(t, s) \right\} Q_{\mathcal{C}} \\ &+ \left. \log \left(\prod_{m=1}^M p(\mathbf{x}_m^{(n+1)}(1)) \prod_{t_x=2}^N p(\mathbf{x}_m^{(n+1)}(t_x) | \mathbf{x}_m^{(n+1)}(t_x - 1)) \right) \right\} Q_{\mathbf{X}} \\ &+ \sum_{s=1}^S \sum_{t=1}^N \sum_{m=1}^M \sum_{r=1}^{n_t(s)} \\ &\times \log(\mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_m^{(n+1)}(t), \mathbf{R}_{r,s}(t)\}) w_{m,r}^{(n)}(t, s) \quad (23) \end{aligned}$$

The maximization of the state component of the Q -function, $Q_{\mathbf{x}}$, from (23), is performed indirectly by finding an equation with the same gradient, $\nabla_{\mathbf{x}^{n+1}} Q_{\mathbf{x}}$, and thus the same inflection points. Omitting constant terms, the derivative taken over the innermost sum may be transformed as follows:

sensors but no data observation uncertainty. The second term of (27) may be rewritten as follows:

$$-\frac{1}{2} \sum_{t=1}^N \sum_{m=1}^M (\tilde{\mathbf{z}}_m - \mathbf{H}_m \mathbf{x}_m^{(n+1)}(t))' \tilde{\mathbf{R}}_m(t)^{-1} (\tilde{\mathbf{z}}_m - \mathbf{H}_m \mathbf{x}_m^{(n+1)}(t)). \quad (28)$$

$$\nabla_{\mathbf{x}} \left[\sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) (\mathbf{z}_{r,s}(t) - \mathbf{H}_{r,s} \mathbf{x}^{(n+1)}(t)) \mathbf{R}_{r,s}(t)^{-1} (\mathbf{z}_{r,s}(t) - \mathbf{H}_{r,s} \mathbf{x}^{(n+1)}(t))' \right] \quad (24a)$$

$$= \sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) \mathbf{H}'_{r,s} \mathbf{R}_{r,s}(t)^{-1} (\mathbf{z}_{r,s}(t) - \mathbf{H}_{r,s} \mathbf{x}^{(n+1)}(t)) \quad (24b)$$

$$= \sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) \mathbf{H}'_{r,s} \mathbf{R}_{r,s}(t)^{-1} \mathbf{z}_{r,s}(t) - \sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) \mathbf{H}'_{r,s} \mathbf{R}_{r,s}(t)^{-1} \mathbf{H}_{r,s} \mathbf{x}^{(n+1)}(t) \quad (24c)$$

$$= \left(\sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) \mathbf{H}'_{r,s} \mathbf{R}_{r,s}(t)^{-1} \mathbf{H}_{r,s} \right) \times \left(\left(\sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) \mathbf{H}'_{r,s} \mathbf{R}_{r,s}(t)^{-1} \mathbf{z}_{r,s}(t) \right) - \mathbf{x}^{(n+1)}(t) \right) \quad (24d)$$

$$= \tilde{\mathbf{R}}_{m,s} (\tilde{\mathbf{z}}_{m,s} - \mathbf{x}^{(n+1)}(t)) \quad (24e)$$

$$= \nabla_{\mathbf{x}} [(\tilde{\mathbf{z}}_{m,s}(t) - \mathbf{x}^{(n+1)}(t)) \tilde{\mathbf{R}}_{m,s}(t)^{-1} (\tilde{\mathbf{z}}_{m,s}(t) - \mathbf{x}^{(n+1)}(t))']. \quad (24f)$$

The synthetic measurements $\tilde{\mathbf{z}}_{m,s}(t)$ and $\tilde{\mathbf{R}}_{m,s}(t)^{-1}$ are defined by

$$\tilde{\mathbf{R}}_{m,s}(t)^{-1} = \sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) \mathbf{H}'_{r,s} \mathbf{R}_{r,s}(t)^{-1} \mathbf{H}_{r,s} \quad (25)$$

and

$$\tilde{\mathbf{z}}_{m,s}(t) = \tilde{\mathbf{R}}_{m,s}(t) \left(\sum_{r=1}^{n_t(s)} w_{m,r}^{(n)}(t,s) \mathbf{H}'_{r,s} \mathbf{R}_{r,s}(t)^{-1} \mathbf{z}_{r,s}(t) \right). \quad (26)$$

Note that $\tilde{\mathbf{R}}_{m,s}(t)^{-1}$ may not be invertible, so a pseudoinverse may be necessary. The equivalency between (24a) and (24f) means that $Q_{\mathbf{x}}$ from (23) has the same derivative as

$$\begin{aligned} & \hat{Q}(\mathbf{X}^{n+1}; \mathbf{X}^n) \\ &= \log \left(\prod_{m=1}^M p(\mathbf{x}_m^{(n+1)}(1)) \prod_{t=2}^N p(\mathbf{x}_m^{(n+1)}(t) | \mathbf{x}_m^{(n+1)}(t-1)) \right) \\ & - \frac{1}{2} \sum_{s=1}^S \sum_{t=1}^N \sum_{m=1}^M (\tilde{\mathbf{z}}_{m,s} - \mathbf{x}_m^{(n+1)}(t))' \\ & \times \tilde{\mathbf{R}}_{m,s}(t)^{-1} (\tilde{\mathbf{z}}_{m,s} - \mathbf{x}_m^{(n+1)}(t)). \end{aligned} \quad (27)$$

Equation (27) is the joint likelihood function of M targets for which there are observations from multiple

Letting \mathbf{I}_{x_m} be an identity matrix whose width is equal to the number of states in \mathbf{x}_m , $\tilde{\mathbf{z}}_m(t)$ and $\tilde{\mathbf{R}}_m(t)$ are given by

$$\tilde{\mathbf{z}}_m(t) = [\tilde{\mathbf{z}}_{m,1}(t), \tilde{\mathbf{z}}_{m,2}(t), \dots, \tilde{\mathbf{z}}_{m,S}(t)], \quad (29)$$

$$\mathbf{H}_m = [\mathbf{I}_{x_m,1}, \mathbf{I}_{x_m,2}, \dots, \mathbf{I}_{x_m,S}], \quad (30)$$

and

$$\tilde{\mathbf{R}}_m(t) = \text{diag}[\tilde{\mathbf{R}}_{m,1}, \tilde{\mathbf{R}}_{m,2}, \dots, \tilde{\mathbf{R}}_{m,S}]. \quad (31)$$

Substituting equation (28) into (27) is equivalent to a single-sensor system with no data association uncertainty having measurements given by (29), (30), and (31). The maximization of Q is thus the maximization of a single sensor system, the solution of which is well known (e.g., [4]) to be the use of the Kalman smoother (the equations for the Kalman smoother are summarized in Appendix D). This is a simpler approach than using the Levenberg-Marquardt nonlinear regression procedure, as suggested by Giannopoulos, Streit, and Swaszek [24] in the original derivation of the PMHT with multiple sensors. The use of the Kalman smoother was also present in the original single sensor PMHT algorithm.

This method of stacking measurements is a common method of measurement fusion for the Kalman filter when there is no target-measurement association uncertainty. Gan and Harris [21] showed that if at a

particular time for a particular track all sensors have the same measurement matrix, which being $\mathbf{I}_{\mathbf{x}_{m,1}}$ is true in this case, then the above method of merging the states is equivalent to a simpler method. Namely, each track is updated using a single, merged measurement given by

$$\tilde{\mathbf{z}}_m(t) = \left(\sum_{s=1}^S \tilde{\mathbf{R}}_{m,s}(t)^{-1} \right)^{-1} \sum_{s=1}^S \tilde{\mathbf{R}}_{m,s}(t)^{-1} \tilde{\mathbf{z}}_{m,s}(t) \quad (32)$$

and

$$\tilde{\mathbf{R}}_m(t) = \left(\sum_{s=1}^S \tilde{\mathbf{R}}_{m,s}(t)^{-1} \right)^{-1}. \quad (33)$$

The Kalman smoother may be thought of as running a forward Kalman filter, and then running a backwards smoothing operation on the track estimate resulting from the Kalman filter step. Note that the use of the pseudoinverse in (26) may be completely avoided if the information filter (described, for example, in [4]) is used in place of the Kalman filter in the first half of the Kalman smoother. The information filter calls for $\tilde{\mathbf{R}}_{m,s}(t)^{-1} \tilde{\mathbf{z}}_{m,s}(t)$, obviating the need to invert $\tilde{\mathbf{R}}_{m,s}(t)^{-1}$ in (26).

In general, except when range-rate information is provided by the sensors, all $\mathbf{H}_{r,s}$ for a particular sensor s will be the same for all measurements. In this instance, the steps leading up to (24f) may be simplified, resulting in the following simplified synthetic measurements

$$\tilde{\mathbf{z}}_{m,s}(t) = \left(\sum_{r=1}^{n_r(s)} w_{m,r}^{(n)}(t,s) \mathbf{R}_{r,s}(t)^{-1} \right)^{-1} \sum_{r=1}^{n_r(s)} w_{m,r}^{(n)}(t,s) \mathbf{R}_{r,s}(t)^{-1} \mathbf{z}_{r,s}(t) \quad (34)$$

and

$$\tilde{\mathbf{R}}_{m,s}(t) = \left(\sum_{r=1}^{n_r(s)} w_{m,r}^{(n)}(t,s) \mathbf{R}_{r,s}(t)^{-1} \right)^{-1}. \quad (35)$$

The form of the synthetic measurements in (34) and (35), allowing for each measurement to have a different covariance matrix was first given in [66]. Previous versions assumed that all measurements have the same covariance. The forms given in (25) and (26) allowing for different measurement matrices, as occurs with doppler measurements, are unique to this paper. Note that if each sensor has a different measurement matrix, then the measurement fusion method given in (32) and (33) is no longer optimal. In this case, the merged measurement given in (29) and (31) should be used with the modified merged measurement matrix,

$$\mathbf{H}_m = [\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_S] \quad (36)$$

where \mathbf{H}_s is the measurement matrix of the s th sensor.

The maximization of the confusion matrix via the gradient $\nabla_{\mathbf{c}_{n+1}} \mathcal{Q}_C$ from (23) is performed under the constraint

$$\sum_{i=1}^{M_C} c_{i,m} = 1. \quad (37)$$

Using (23) and (37), the Lagrangian to maximize is:

$$L_C = \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_r(s)} \sum_{m=1}^M \log(c(z_{r,s}^C(t), m)) w_{m,r}^{(n)}(t,s) + \sum_{m=1}^M \lambda_m^C \left(1 - \sum_{i=1}^{M_C} c(i, m) \right) \quad (38a)$$

$$= \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_r(s)} \sum_{m=1}^M \sum_{i=1}^{M_C} \delta(z_{r,s}^C - i) \log(c(i, m)) w_{m,r}^{(n)}(t,s) + \sum_{m=1}^M \lambda_m^C \left(1 - \sum_{i=1}^{M_C} c(i, m) \right). \quad (38b)$$

Equation (38a) is equivalent to (38b), where $\delta(t)$ is the Kronecker Delta function, which is one for $t = 0$ and zero otherwise. Differentiating (38b) with respect to a particular $c(i, m)$ gives

$$c_{i,m} = \frac{1}{\lambda_m^C} \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_r} \delta(z_{r,s}^C - i) w_{m,r}^{(n)}(t,s). \quad (39)$$

Applying the constraint given in (37) gives

$$\lambda_m^C = \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_r} \sum_{i=1}^{M_C} \delta(z_{r,s}^C - i) w_{m,r}^{(n)}(t,s) \quad (40a)$$

$$= \sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_r} w_{m,r}^{(n)}(t,s). \quad (40b)$$

Combining (39) and (40b) gives us the update for the $c_{i,m}$:

$$c_{i,m} = \frac{\sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_r} \delta(z_{r,s}^C - i) w_{m,r}^{(n)}(t,s)}{\sum_{s=1}^S \sum_{t=1}^N \sum_{r=1}^{n_r} w_{m,r}^{(n)}(t,s)}. \quad (41)$$

4.1. Regarding the Kalman Smoothing Step

The equations for the Kalman smoother are given in Appendix D. It should be noted that although the EM algorithm might call for the initial state estimate for each track, $\mathbf{x}_m(1)$, to be smoothed along with the rest, practically the algorithm is not usable in this manner. The first part of the Kalman smoother is done by running a Kalman filter forward on the data. This requires a covariance estimate for the initial state. On the first iteration of the EM algorithm, this is not a problem. On additional iterations, however, we do not have a valid estimate for the covariance of the smoothed initial estimate. If one were to use the covariance estimate coming out of the Kalman smoother, then this value would decrease every iteration as a result of ‘‘information incest.’’ That is, the initial state would repeatedly get smoothed by using much of the same data as before, but the Kalman

smoother would interpret this data as being “new” and at every iteration the covariance estimate of the initial state would decrease. After enough iterations, the covariance assumed for the initial state will approach zero even though we would not have supreme confidence in the initial state.

A solution to this problem is to forego smoothing the initial state at each step and to use its initial state covariance at every iteration. That is equivalent to taking the initial state out of the set of states \mathbf{X} that are to be estimated by the EM algorithm.

4.2. Precision Problems with the PMHT

Being based thereupon, all versions of the PMHT algorithm suffer the same precision problems that can occur with regular Kalman filter. Verhaegen discusses the source of some of these problems as well as their remedies [63] and such problems are also discussed in most textbooks, such as [4]. The PMHT, however, has a number of its own precision problems that must be taken into account when designing any implementation.

At any step when calculating the posterior association probabilities values for non-clutter targets, if all valid measurements are far from the predicted value $\hat{\mathbf{y}}_{k_r,s}^n(t)$, then it is quite likely that precision limitations will render all of the w s to be zero.⁷ In many such instances, one can forego the use of the w s and assume that there was a missed detection. In comparison with other algorithms, precision is a serious problem in the PMHT, because the non-clutter PDFs in w

4.3. Using Deterministic Annealing

The use of deterministic annealing can both help the PMHT to converge to the global MAP estimate as well as ameliorate precision problems associated with the posterior association probabilities. Which maxima the EM algorithm converges to is highly dependent on the initial state estimates over the entire batch. Deterministic annealing is an approach to reduce the dependence of the algorithm on the initial estimates. However, this may require more iterations of the EM algorithm than if deterministic annealing were not used. As a result, just incorporating deterministic annealing without changing the number of iterations used could theoretically worsen performance.

As shown in (7), the EM function with deterministic annealing replaces $p(\mathbf{K} | \mathbf{X}^n, \mathbf{Z})$ in the regular EM algorithm with

$$p(\mathbf{K} | \mathbf{X}^n, \mathbf{C}^n, \mathbf{Z}) = \frac{p(\mathbf{Z}, \mathbf{K}, \mathbf{X}^n, \mathbf{C}^n)^\beta}{\sum_{\mathbf{K}_1} p(\mathbf{Z}, \mathbf{K}_1, \mathbf{X}^n, \mathbf{C}^n)^\beta d\mathbf{K}_1}. \quad (42)$$

The solution for $p(\mathbf{K} | \mathbf{X}^n, \mathbf{Z})$ given in (15) is a product of w terms. The addition of the β terms in (42) to (15) can take place without explicitly decomposing $p(\mathbf{K} | \mathbf{X}^n, \mathbf{Z})$ into the parts listed in (42).

We note that each w term in (15) has a single value in the numerator as well as a sum of values in the denominator. The inclusion of the β s is done by modifying the w terms as follows. For non-clutter association probabilities, each w shall be adjusted from (17) according to

$$w_{k_r,s}(t,s) = \frac{(\pi_{k_r,s}(n_t(s), t) c(z_{r,s}^C(t), k_{r,s}(t)) \mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_{k_r,s}(t), \mathbf{R}_{r,s}(t)\})^\beta}{(\pi_0(n_t(s), t) \mu(t, \mathbf{z}_{r,s}(t)) c(z_{r,s}^C(t), 0))^\beta + \sum_{m=1}^M (\pi_m(n_t(s), t) c(z_{r,s}^C(t), m) \mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_m(t), \mathbf{R}_{r,s}(t)\})^\beta)}. \quad (43)$$

are normally distributed having a covariance equal to that of the measurement. In instances where the process noise covariance is large and the measurement noise covariance is small, precision errors can cause all of the w s for a particular target to be zero much of the time. Thus, paradoxically, the performance of the PMHT can worsen as the magnitude of the measurement covariance decreases. In instances where no measurement noise is present, the PMHT is unusable.

⁷If a clutter model is present, then there will always be a nonzero (clutter) term in the denominator of the w s, thus precision problems can render all of the w s to be zero. However, if there is no clutter term, then precision problems couples with distant measurements can result in the fraction computed for the w to evaluate as 0/0. This can signify that the target was not detected, or that the observation from the target was very far from the predicted position.

Equation (43) is equivalent to raising the numerator and each term in the denominator of (17) to the power of β . The same would be done with the w for the clutter assignment, which we have omitted; there too one would raise the numerator and each term of the denominator to β . Once all of the w terms have been multiplied, as in (15) the result is thus the same as (42) except all common terms have been canceled out between the numerator and denominator. That is, the result is still a numerator raised to β and a denominator consisting of a sum of terms each raised to β . This is the same as the solution given by Wieneke and Koch [66] without derivation and similar to what Strandlie and Zerubia [57] derived.

For precision purposes, the exponentiation of the normal PDFs in (43) is best performed by distributing β to the terms of the normal PDF, rather than evaluating the normal PDF and then exponentiating it. Because

$\beta \leq 1$, this increases the argument of the exponential function of the PDF, which is where underflow problems are most likely to occur.

4.4. Sensors with Different Fields of View

The state estimate for the PMHT was derived assuming that all sensors have the same field of view. When the sensors have different fields of view, the calculation of the posterior association probabilities (the w s) is different for each sensor.

Generally, a particular target will have a certain probability of detection when viewed by a particular sensor. In calculating the posterior association probabilities, this detection probability is necessary for calculating the prior association probabilities. This detection probability can be considered to be the product of the probability that the target is located within the field of

5. THE COMPLEXITY OF THE JPDAF VS. THE PMHT

The most complex part of the JPDAF is the evaluation of the posterior association probabilities. These are equivalent to the posterior association probabilities in the single-sensor PMHT, but with slightly different conditioning. The evaluation of these probabilities is complex, because it requires the evaluation and normalization of the likelihoods of all possible target-measurement assignment combinations, a task requiring the evaluation of the exponential function for every likelihood.

In the worst-case scenario, every measurement at time step t would fall in every target's gating region. Let n_t be the number of measurements at step t and M be the number of targets. The number of possible target-measurement assignments may be decomposed based upon the number of targets observed and is given as follows:

$$\begin{aligned}
 A_{\text{JPDAF}} &= \underbrace{\sum_{l=0}^{\min(n_t, M)}}_{\text{Sum over the number of targets observed}} \underbrace{\binom{M}{l}}_{\text{Choose which targets are observed}} \underbrace{\binom{n_t}{l}}_{\text{Choose which measurements are observed}} \underbrace{l!}_{\text{Assign the measurements to the targets}} \quad (44a) \\
 &= {}_2F_0[-n_t, -M; 1]. \quad (44b)
 \end{aligned}$$

view of the sensor times the probability that the sensor detects the target given that it is in its field of view.

As shown in Appendix B, the computation of the prior association probabilities (the π s) is combinatorially complex if all of the targets have different probabilities of detection. This is the case when one takes into account the probability that each target is within the field of view of each sensor. The complexity of this situation may be reduced either by assuming a constant detection probability for all targets within the field of view of a sensor and gating to targets that should be within the field of view given the state estimates. Once gating has been done, this means that the prior and posterior association probabilities for each sensor are calculated assuming a reduced number of targets: only those that fall within the gate for that sensor.

4.5. Out-of-Sequence Measurements

In many practical data fusion schemes, measurements may arrive at the fusion center out of sequence. As was noted by Efe, Ruan, and Willett [18], the PMHT handles such situations with ease. Because the PMHT is a batch algorithm, as long as newly received measurements correspond to a step that has not left the sliding window, the measurements may be added to the batch at any time and are used in the state update.

${}_2F_0$ refers to a generalized hypergeometric function. The step from (44a) to (44b) was performed by noting that the ratio of the a_{l+1} and the a_l th term of the sum in (44a) is:

$$\frac{a_{l+1}}{a_l} = \frac{(l - n_t)(l - M)}{1 + l}. \quad (45)$$

More information on the conversion of sums to hypergeometric functions may be found in [48].

In contrast, although the PMHT allows for more posterior association probabilities than the JPDAF, due to their product form (i.e., the assumed independence of the associations) these do not need to be enumerated individually. The evaluation of each measurement association probability w requires evaluating a single normal PDF, and in the end normalizing over all w terms and a clutter term. Thus, the number of evaluations of the exponential function that need be done for one iteration at one time step in the PMHT is equal to the number of w terms, which is $n_t M$. Thus, if the batch length of the PMHT is N , and I iterations are used then, noting that the first estimate in the batch does not change with each iteration, the overall complexity of the PMHT is:

$$A_{\text{PMHT}} = IM \sum_{t=2}^N n_t. \quad (46)$$

As shown in Table I, when the number of targets is small, the PMHT will have a higher complexity than

TABLE I
Number of Combinations Considered by the JPDAF versus the PMHT as a Function of the Number of Targets

M	A_{JPDAF}	A_{PMHT}
1	$n_t + 1$	$I \sum_{i=2}^N n_i$
2	$n_t^2 + n_t + 1$	$2I \sum_{i=2}^N n_i$
3	$n_t^3 + 2n_t + 1$	$3I \sum_{i=2}^N n_i$
4	$n_t^4 - 2n_t^3 + 5n_t^2 + 1$	$4I \sum_{i=2}^N n_i$

n_t is the number of measurements in the frame considered, M is the number of targets and I is the number of iterations that the PMHT uses.

the JPDAF. Consistent with the simulation results of Pao [46], practical implementations of the PMHT will never have a lower complexity when there is only one target. However, the complexity of the JPDAF scales exponentially with the number of targets, whereas the complexity of the PMHT scales linearly. So keeping the batch length N fixed, the PMHT has a lower complexity as the number of targets becomes large.

6. USING THE TRACKER OVER TIME

6.1. Growing and Sliding the Batch

The PMHT requires an initial estimate for all of the states in the batch, as well as a the covariance of the state estimate at the first time step. As empirically demonstrated by Willett, Ruan, and Streit [70], a practical, efficient way of running the PMHT using a finite-length batch is by growing and then sliding the batch. In other words, at time $t = 1$ one is given the initial state estimates of the targets $\mathbf{x}_m(1)$. From time $t = 2$ to time $t = N$, where N is the maximum batch-length, the PMHT is run while increasing the batch length by one each time. As mentioned in Section 4.1, in order to eliminate “data incest,” the state estimates at the beginning of the batch should be removed from the estimation that is not updated as part of the Kalman smoothing step. The initial estimates of the states for the rest of the batch are very important for convergence. Even with the use of deterministic annealing, as described in Section 4.3, if the initial state estimates for the batch are particularly bad, then the EM algorithm probably will not converge to the global maximum, nor to a nearby local maximum. For that reason, the best initial state estimates are the estimates from the previous time step. The best initial state estimate for each target at the new step, which was not estimated at the previous time step, is the the best a priori estimate, which is the Kalman filter estimate $\mathbf{x}_m(t | t - 1)$.

At all time steps after the batch length has reached N , the batch should slide one step forward, as demonstrated in Fig. 1 by the Single Shift batch with respect to the Pre-Shift Batch for a length $N = 4$ batch. In this case, the previous estimate from the first time step becomes the new initial state of the first step of the batch, which

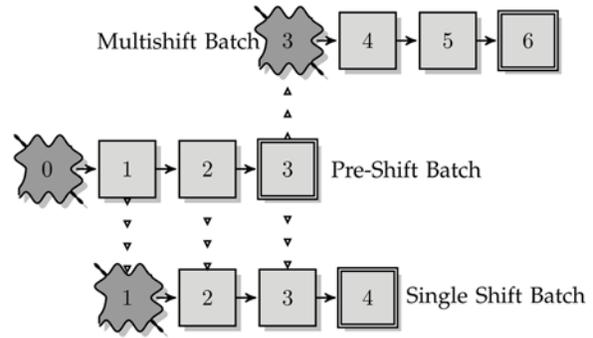


Fig. 1. Different methods of shifting the window for the next time step as shown on a length-4 batch. Each method has its own concerns regarding the consistency of the estimator and the avoidance of “data incest.”

must have an accurate covariance. Section 7 looks at ways of estimating the covariance of this state estimate as well as those of the other state estimates in the batch.

6.2. Other Methods of Sliding the Batch

It should be noted that the aforementioned method of sliding the batch is somewhat *ad-hoc*. Much literature has also focussed on sliding the batch over multiple time steps at once, as shown with the Multishift Batch in Fig. 1. If this approach is taken, one can not obtain a real-time estimate of the target’s location, but must wait until another batch of information has arrived. Ruan, Willett, and Streit [52] suggested that the batches should not overlap by more than one time step and later suggested [70] that the initial state of the slid batch, be it slid one step or many, be calculated using only information equal to or prior to that time period. For example, in the Single-Shift batch in Fig. 1, only information from times 0 and 1 could be used to create the initial state estimate at time 1. However, a length-one PMHT is not a very good tracker. As a result, by using this approach the initial estimates become progressively worse.

The main concern regarding reusing smoothed past state estimates is that it introduces “information incest” in the smoothed state. However, it should be noted that this concern only exists with the initial estimate at the beginning of the batch. The rest of the initial estimates in the batch affect to which local maximum the EM algorithm is likely converge, but they do not affect the location of the maxima in the likelihood function.

The new initial state when using a Single-Shift batch from Fig. 1 and the smoothed state estimate from the previous batch as the initial state, introduces information incest in that it has already been smoothed from future observations. In the Multi-Shift batch, where only a single state overlaps, no information incest is present. However, in the example of Fig. 1, the initial estimate for time steps 4, 5, and 6 in the slid batch would have to be Kalman filter predictions from the estimate at

time 3. If these are far from the true track location, then it is likely that the precision errors will occur, as described in Section 4.2 or that the EM algorithm is likely to converge to a local maximum far from the global maximum.

Thus, there is a tradeoff between how far one slides the batch and how much ‘‘information incest’’ one wishes to allow in the state estimate at the beginning of the slid batch. Wieneke and Koch [66] decided to shift a small number of steps, less than the batch length, and use deterministic annealing. However, in many practical situations, a state estimate is desired at every time step, and thus the method of growing and sliding the batch described in Section 6.1 is a simple approach.

7. COVARIANCE ESTIMATION IN THE PMHT

When sliding the batch, as described in the previous section, an accurate covariance is needed for the new estimate of the first time step of the batch. Additionally, at any time during tracking, one may wish to have a covariance for the target state estimate. Being based upon the EM algorithm, the PMHT does not directly provide this. The simplest approximation is to use the covariance estimates that are produced by the Kalman smoother at each step. However, based upon the Normalized Estimation Error Squared (NEES), which is more closely defined in Section 9, this has been shown to be inconsistent.

In this section, we will look at two methods of producing covariance estimations for the PMHT. One approach, originally presented by Walsh [65], is to use the inverse of the observed information to predict the covariance. A simpler *ad-hoc* approach by Blanding, Willett, Streit, and Dunham [6] is to obtain covariance approximations by normalizing the posterior association and using the covariance estimate from the JPDAF, (as described, for example, in [3]).

7.1. Using the Observed Information to Estimate the Covariance

Letting D be the dimensionality of the state, the observed information matrix is defined as the $DMN \times DMN$ Hessian of the joint likelihood function of the states and observations over all time, evaluated at the state estimate, in this case the EM estimate:

$$I[\hat{\mathbf{X}}, \mathbf{Z}] \triangleq -\nabla_{\hat{\mathbf{X}}}^2 \log p(\mathbf{Z}, \mathbf{X})|_{\mathbf{X}=\hat{\mathbf{X}}} \quad (47)$$

$$= [-\nabla_{\hat{\mathbf{X}}}^2 \log p(\mathbf{X}) - \nabla_{\hat{\mathbf{X}}}^2 \log p(\mathbf{Z} | \mathbf{X})]|_{\mathbf{X}=\hat{\mathbf{X}}} \quad (48)$$

$$= I_{\text{prior}}[\hat{\mathbf{X}}] + I_{\text{data}}[\mathbf{Z} | \hat{\mathbf{X}}]. \quad (49)$$

The inverse of the observed information gives the covariance for all states over all time. The covariances of the individual state estimates are in the $D \times D$ blocks lying on the diagonal of the matrix.

Walsh is the first to have derived these Hessians for the PMHT. We shall give a multisensor adaptation of the observed information matrix as explained by [6], assuming that all observations for a particular sensor at a particular time have the same covariance and measurement matrices. $I_{\text{prior}}[\hat{\mathbf{X}}]$ is given as

$$I_{\text{prior}}[\hat{\mathbf{X}}] = \text{diag} \left[\begin{array}{cc} \chi(t) & -\delta(t) \\ -\delta(t)' & \chi(t+1) \end{array} \right] : \quad t = 1, \dots, N-1 \quad (50)$$

with

$$\chi(t) = \begin{cases} \text{diag}[\mathbf{P}_m(1|1)^{-1} + \mathbf{F}_m(1)'\mathbf{Q}_m(1)^{-1}\mathbf{F}_m(1) : \\ \quad m = 1, \dots, M] & \text{for } t = 1 \\ \text{diag}[\mathbf{Q}_m^{-1}(t-1) + \mathbf{F}_m(t)'\mathbf{Q}_m^{-1}(t)\mathbf{F}_m(t) : \\ \quad m = 1, \dots, M] & \text{for } t = 2, \dots, T-1 \\ \text{diag}[\mathbf{Q}_m^{-1}(T-1) : \\ \quad m = 1, \dots, M] & \text{for } t = T \end{cases} \quad (51)$$

and

$$\delta(t) = \text{diag}[\mathbf{F}_m(t)'\mathbf{Q}_m(t)^{-1} : \\ m = 1, \dots, M] \quad \text{for } t = 1, \dots, T-1. \quad (52)$$

$\chi(t)$ and $\delta(t)$ are $MN \times MN$ matrices. The contribution from the data is given by

$$I_{\text{data}}[\mathbf{Z} | \hat{\mathbf{X}}] = \sum_{s=1}^N (\mathbf{B}(s) - \mathbf{C}(s) + \mathbf{D}(s)), \quad (53)$$

$$\mathbf{B}(s) = \text{diag}[\mathbf{B}(t, s) : t = 1, \dots, T], \quad (54)$$

$$\mathbf{C}(s) = \text{diag}[\mathbf{C}(t, s) : t = 1, \dots, T], \quad (55)$$

$$\mathbf{D}(s) = \text{diag}[\mathbf{D}(t, s) : t = 1, \dots, T], \quad (56)$$

$$\mathbf{B}(t, s) = \text{diag}[\mathbf{H}_{m,s}(t)'\tilde{\mathbf{R}}_{m,s}(t)^{-1}\mathbf{H}_{m,s}(t) : \\ m = 1, \dots, M], \quad (57)$$

$$\mathbf{C}(t, s) = \text{diag} \left[\mathbf{H}_s(t)'\mathbf{R}_s(t)^{-1} \right. \\ \left. \times \left(\sum_{r=1}^{n_i(s)} w_{m,r}(t, s) \nu_{m,r}(s, t) \nu_{m,r}(s, t)' \right) \right. \\ \left. \times \mathbf{R}_s(t)^{-1}\mathbf{H}_s(t) : m = 1, \dots, M \right], \quad (58)$$

$$\mathbf{D}(t, s) = \sum_{r=1}^{n_i(s)} \mathbf{D}_r(t, s) \mathbf{D}_r(t, s)', \quad (59)$$

and

$$\mathbf{D}_r(t, s) = \begin{bmatrix} w_{m,r}(t, s) \mathbf{H}_s(t)'\mathbf{R}_1^{-1}(t) \nu_{1,r}(t) \\ \vdots \\ w_{M,r}(t, s) \mathbf{H}_s(t)'\mathbf{R}_M^{-1}(t) \nu_{M,r}(t) \end{bmatrix}. \quad (60)$$

Note that $B(t, s)$ contains the synthetic measurement covariance $\tilde{\mathbf{R}}_{m,s}(t)$, whereas the other equations only contain the covariance of the measurements. The innovations are defined as follows:

$$\nu_{m,r}(s, t) = \mathbf{H}_s(t)\mathbf{x}_m(t) - \mathbf{z}_{r,s}(t). \quad (61)$$

The derivation of the observed information was performed under the assumption that the transition matrix \mathbf{F} and the process noise covariance matrix \mathbf{Q} are invertible. This is true when using the discretized continuous white noise acceleration model for the motion, but is not true when using the discrete white noise acceleration model [4], in which case a pseudoinverse would be necessary.⁸

7.2. A Simpler, *Ad-hoc* Approach to Covariance Estimation

We shall extend the *ad-hoc* covariance estimation approach taken by Blanding, Willett, Streit, and Dunham [6] to the multisensor case. That is, we shall show that estimator consistency is improved when the posterior assignment probabilities are normalized and the covariance estimate from the MSJPDAF is used. The MSJPDAF is a generalization of the JPDAF to multiple sensors (see, for example [3] for information on the basic JPDAF). There exist two forms of the MSJPDAF, a sequential and a parallel one, which were contrasted by Pao and Frei [45]. Because the sensor fusion is done in parallel at each step of the PMHT, we shall consider the parallel version of the MSJPDAF.

The MSJPDAF requires that the posterior association probabilities sum to one over all measurements for a particular target plus the probability that target was not detected. This is because the MSJPDAF does not make the same assumption as the PMHT, that each target can produce more than one measurement at a particular time. By noting that in the PMHT measurement model the assignment of one measurement to the target has no bearing on the probability that another measurement is assigned to the same target, the probability that a particular target m was not detected by a particular sensor s at a particular time t is given as follows:

$$\beta_{m,0,s}(t) = \prod_{r=1}^{n_r} (1 - w_{m,r}(t, s)). \quad (62)$$

Thus, the normalization over the observations, not changing the probability of a missed detection is:

$$\beta_{m,r,s}(t) = w_{m,r}(t, s) \frac{1 - \beta_{m,0,s}(t)}{\sum_{r=1}^{n_r} w_{m,r}(t, s)}. \quad (63)$$

⁸The Moore-Penrose pseudoinverse of matrix A is $A^{-1} = (AA')^{-1}A$. If AA' is poorly conditioned, then the pseudoinverse can produce bad results. In simulations using the discrete white noise acceleration model, we have observed that conditioning is often a problem in evaluating the observed inverse of the information matrix.

The covariance update from the parallel MSJPDAF is:

$$\mathbf{P}_m(t | t) = \left(\sum_{\mathcal{C}} \beta_{m,\mathcal{C}}(t) (\mathbf{P}_{m,\mathcal{C}}(t | t) + \mathbf{x}_{m,\mathcal{C}}(t | t) \mathbf{x}_{m,\mathcal{C}}(t | t)') \right) - \mathbf{x}_m(t | t) \mathbf{x}_m(t | t)'. \quad (64a)$$

Equation (64a) is the form given by Pao and Frei [45]. \mathcal{C} represents a particular combination of assignments between sensors for a particular target. Each $\beta_{m,\mathcal{C}}$ is a product of $\beta_{m,r,s}(t)$ terms over all sensors for a combination of assignments r , for each sensor. The whole set of \mathcal{C} is the set of all possible measurement to target and clutter assignments at a particular time over all sensors. This means that the covariance calculation is roughly exponentially complex as a function of the number of sensors.

In (64a), $\mathbf{x}_{m,\mathcal{C}}(t | t)$ represents the state update of the Kalman filter if the measurement-assignment for all sensors given by \mathcal{C} is correct. This means fusing the actual observations in the same way that the synthetic measurements were fused in (29), (30), and (31) and then updating the state estimate from the PMHT. By updating the PMHT state estimate reusing the observations, a certain degree of “data incest” is added, but again this method is *ad-hoc* and problems with the “incest” were not observed in previous work using a single sensor [6]. The covariance $\mathbf{P}_{m,\mathcal{C}}(t | t)$ is likewise what the covariance would be if assignment \mathcal{C} is correct. Since only the first state of the PMHT has a covariance, the pre-update covariance at time t would consist of $\mathbf{P}_m(t | t - 1)$, that is the forward predicted covariance from the previous estimate. In this manner, this covariance estimation algorithm must be done in order from the first to the last state estimate. $\mathbf{x}_m(t | t)$ is a the weighted average of these other state updates:

$$\mathbf{x}_m(t | t) = \sum_{\mathcal{C}} \beta_{m,\mathcal{C}} \mathbf{x}_{m,\mathcal{C}}(t | t). \quad (65)$$

8. A SUMMARY OF THE BASIC ALGORITHM

We shall give a summary of the basic PMHT algorithm assuming that all targets have the same probability of detection and that the measurements all have the same measurement matrix \mathbf{H} . If the detection probabilities are different for all targets, then the prior and posterior association probabilities referenced from the appendices should be used. If the observations have different measurement matrices, as might be the case if range-rate information is available, then the alternative state update equations given in Section 4 should be used.

- 1) Set the initial state estimate for each target at the current time step to the Kalman filter predicted value of the state $\mathbf{x}_m(t | t - 1)$.
- 2) For each sensor and observation, calculate the posterior assignment probabilities $w_{k_{r,s}(t),r}(t, s)$ according to (18) and (19).

- 3) Create the synthetic measurements $\tilde{\mathbf{z}}_{m,s}(t)$ with their corresponding measurement covariances $\tilde{\mathbf{R}}_{m,s}(t)$ for each target, measurement and sensor according to (25) and (26) or (35) and (34).
- 4) Merge the synthetic measurements between sensors according to (32) and (33).
- 5) Using the fixed initial state estimate $\mathbf{x}_m(1)$ and state covariance estimate $\mathbf{P}_m(1)$ for each track, run the Kalman smoother, as described in Appendix D using the merged measurements as the observations. Do not smooth the initial state.
- 6) Update the confusion matrix using (41).
- 7) Go to step 2. Repeat until convergence of the EM algorithm.
- 8) If desired, estimate the covariance of the updated state estimate $x_m(t|t)$ using a consistent approximation, such as (64a) or as described in Section 7.1.
- 9) Slide the batch window forward, using the proper covariance estimate for the new initial state, such as in (64a) or as described in Section 7.1.
- 10) Go to 1.

9. SIMULATION

We compared the consistency and track retention rates of the MSJPDAF and the MSPMHT with and without deterministic annealing and using the various covariance estimation methods of Section 7 when using two sensors and two targets. The sequential version of the MSJPDAF was used.⁹ We used the two-dimensional discretized continuous white noise acceleration model.¹⁰ Both sensors had the same field of view and measured in Cartesian coordinates without classification information. The ordering of the elements of the state was $[x, y, \dot{x}, \dot{y}]$. Using an 80% probability of detecting each track at each sensor, the simulation parameters were

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (66)$$

$$\mathbf{Q} = \begin{bmatrix} T^3/3 & 0 & T^2/2 & 0 \\ 0 & T^3/3 & 0 & T^2/2 \\ T^2/2 & 0 & T & 0 \\ 0 & T^2/2 & 0 & T \end{bmatrix} \sigma_p^2, \quad (67)$$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad (68)$$

⁹[47] discusses the sequential MSJPDAF algorithm, but provides an incorrect state covariance estimate. [32] provides the correct state covariance estimate when solving a different problem.

¹⁰The one dimensional version is described in [4]; the two-dimensional version follows from it.

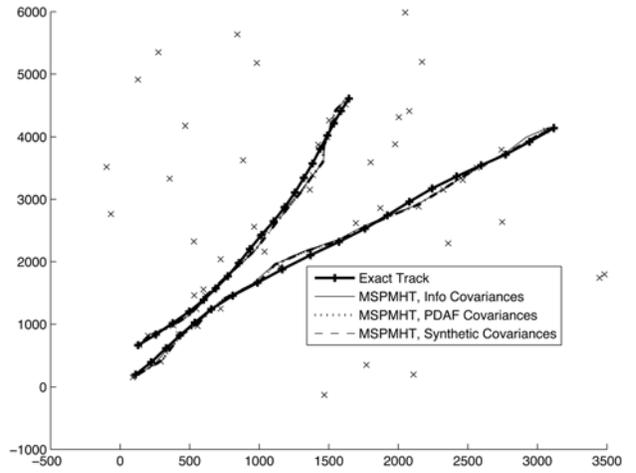


Fig. 2. A typical run of the simulation. The observations from the first sensor and the last frame of clutter for that sensor are shown.

and

$$\mathbf{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \sigma_m^2. \quad (69)$$

The sampling time T was set to 30 seconds. 1,000 Monte Carlo Runs were performed. The first target was placed at the origin given an initial velocity of 7 m/s at a 59° angle from the x axis. The second target was assigned the same speed, and was placed at 500 m on the y axis. Ascending at a 52° degree angle from the x axis. σ_m was chosen to be 50 m for both sensors and σ_p $0.1 \text{ m}^2/\text{s}^3$. Clutter was generated uniformly in a viewing rectangle bounded between $(-200 \text{ m}, -200 \text{ m})$ and $(3.5 \text{ km}, 6 \text{ km})$. The number of clutter points was determined at each time step by a Poisson random variable having mean 23. The MSPDAF was gated to observations within a 99.97 percent probability region around the estimated location of the target. The simulation was initialized by giving two correctly assigned measurements for each track to information filters (the information filter is described in [4]). For the PMHT, a window growing to a maximum of length 10 was used. After the 10th time step, the window slid. The PMHT was performed using 10 iterations at each step.

Fig. 2 shows a typical run. The proportion of tracks not lost at each step was calculated. A track was considered lost if at any point, the true location of the target was outside of the 99.97 percent confidence interval of the estimated target location. Fig. 3 shows the track-loss performance. As expected, deterministic annealing significantly improved the track-loss performance of the MSPMHT.

To evaluate the consistency of the trackers, the average normalized estimation error squared (NEES) for tracks that were not lost was calculated and averaged over all tracks and Monte Carlo runs. The NEES is defined as

$$\text{NEES} = (\mathbf{x}(t) - \mathbf{x}(t|t))\mathbf{P}(t|t)^{-1}(\mathbf{x}(t) - \mathbf{x}(t|t))'. \quad (70)$$

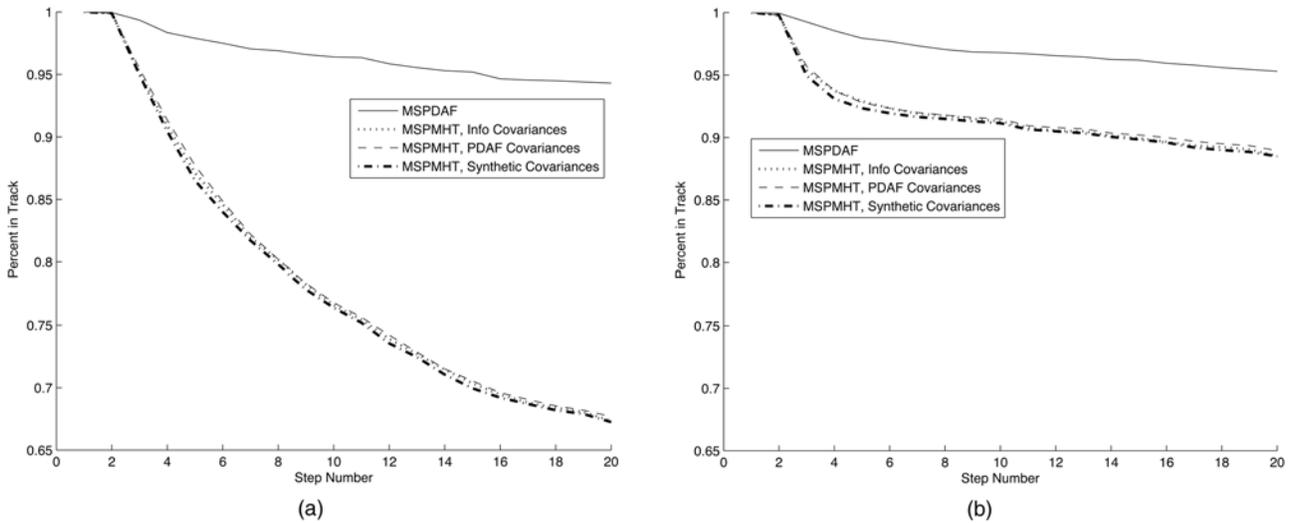


Fig. 3. The fraction of tracks not lost at each step shown with and without the use of deterministic annealing. (a) Without DA. (b) With DA.

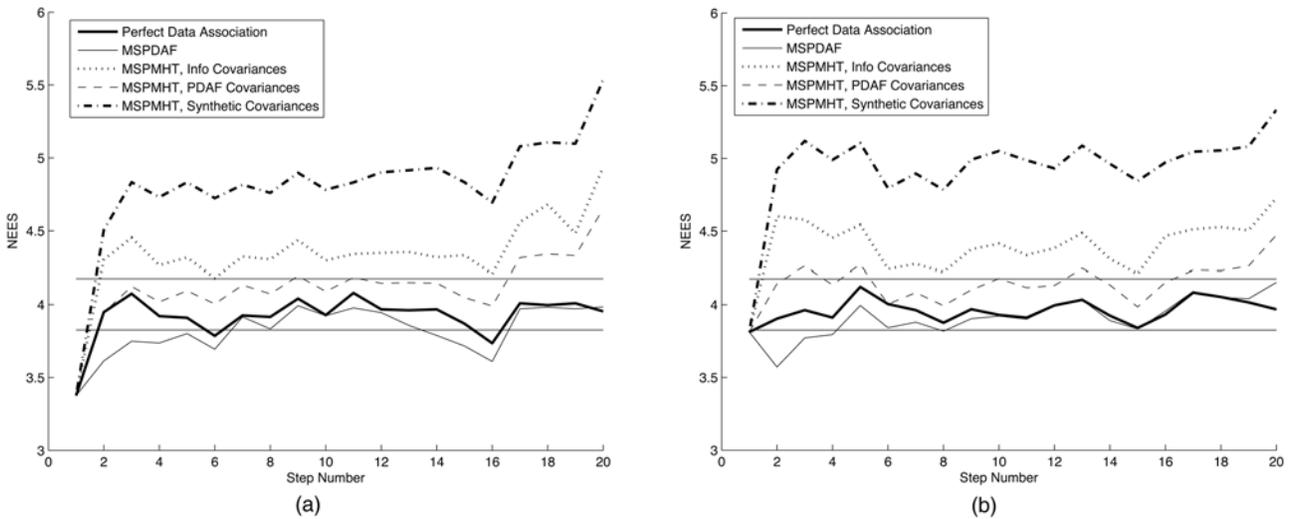


Fig. 4. The average NEES for tracks not lost by any of the trackers. The horizontal lines mark the 95 percent confidence interval of the NEES. (a) Without DA. (b) With DA.

The NEES for a particular track is a random variable with 4 degrees of freedom. The average NEES over n Monte Carlo Runs is a chi-squared random variable with $4n$ degrees of freedom. As mentioned in [4], the inverse Cumulative Distribution Function (CDF) of a chi-squared random variable given ν degrees of freedom, where $\nu > 100$, is approximately

$$\chi_{\nu}^2(p) \approx \frac{1}{2} \left(\mathcal{G}(p) + \sqrt{2\nu - 1} \right)^2. \quad (71)$$

p is the probability at which the inverse CDF is to be evaluated. $\mathcal{G}(p)$ is the inverse CDF of the standard normal distribution. Thus for the 95 percent confidence interval we get:

$$[\mathcal{G}(0.025), \mathcal{G}(0.975)] = [-1.96, 1.96]. \quad (72)$$

As can be seen in Fig. 4, the use of MSJPDFAF covariances in the two-sensor MSPMHT improves the consistency of the tracker, even outperforming those ob-

tained by using the observed information approximations. However, as can be seen in Fig. 3, the use of improved covariances has little effect on track retention. The apparent inconsistency of the initial estimate in Fig. 4 stems from the fact that those tracks most often lost by the PMHT were those where the covariance of the initial estimate was underestimated. Deterministic annealing significantly improves track retention. This coincides with previous results done using a single sensor on a single track [57] and [66].

All together, the MSPMHT performs worse than the MSJPDFAF in the two sensor scenario. However, the poor performance of the basic PMHT has been shown in previous literature and was one of the motivations for the creation of other PMHT algorithms, such as the MF-PMHT by Blanding, Willett, Streit, and Dunham [7]. The multisensor PMHT presented in this paper can form the basis of such modified algorithms.

10. CONCLUSION

We derived a general form of the PMHT involving clutter, multiple sensors, and classification measurements. We provided a simpler method of performing the maximization step when using multiple sensors. We demonstrated that deterministic annealing can significantly improve tracker performance and we showed that the JPDAF covariance approximations provide the most consistent covariance estimates in the simulation, but this consistency has little effect on the track retention as compared to using the synthetic covariances. We also provided a simplified solution for the prior association probabilities (the π s). Although having worse performance than comparable algorithms, such as the MSPDAF, the PMHT algorithm derived here can form the basis of other modifications of the PMHT, such as the Turbo PMHT and the MFPMHT, which achieve better track retention.

APPENDIX A. A DERIVATION OF THE POSTERIOR ASSOCIATION PROBABILITIES $w_{l,r}(t,s)$

The derivation of the posterior association probabilities for the PMHT is dependent upon the sensor in question. For simplicity, we shall assume that we are only considering that which is seen by sensor s and we shall suppress s from the notation. As has typically been done, we shall derive it assuming that all sensors see everything and that there is no gating.

In order to highlight the complexity reduction of the PMHT measurement model, we shall begin by assuming the regular target-measurement assignment model (i.e., that a target can produce only one observation per sensor at each time) before finishing the solution using the PMHT measurement model. Let there be a total of M non-clutter targets and one clutter target $m = 0$. Let $\mathbf{X}(t)$ be the state of the Kalman filters for all non-clutter targets at time t . Define $\mathbf{z}_r(t) \in \mathbf{Z}(t)$ to be the r th measurement out of the set of $\mathbf{Z}(t)$ measurements at time t , which consists of n_t measurements (in order to simplify the notation, we shall omit the subscript on n). $z_r^C(t)$ shall be the classification value associated with measurement $z_r(t)$. Define $k_r(t)$ to be source of the r th measurement at time t . \mathbf{C} shall be the set of all classification probabilities $c(i,m) = \Pr(z_r^C(t) = i | k_r(t) = m)$. The classification of the target is assumed independent of time and the target state. The probability that measurement z_r at time t came from target m given the current set of observations, the set of classification probabilities, and the estimated state of the set of Kalman filter is $w_{m,r}(t) = \Pr(k_r(t) = m | \mathbf{X}(t), \mathbf{Z}(t), \mathbf{C}, n)$. Using Bayes Rule, this may be decomposed as follows:

$$w_{m,r}(t) = \frac{\Pr(\mathbf{Z}(t) | k_r(t) = m, \mathbf{X}(t), \mathbf{C}, n) \Pr(k_r(t) = m | \mathbf{X}(t), \mathbf{C}, n)}{\sum_{p=0}^M \Pr(\mathbf{Z}(t) | k_r(t) = p, \mathbf{X}(t), \mathbf{C}, n) \Pr(k_r(t) = p | \mathbf{X}(t), \mathbf{C}, n)} \quad (73a)$$

$$= \frac{\Pr(\mathbf{Z}(t) | k_r(t) = m, \mathbf{X}(t), n) \pi_m(n) c(z_r^C, m)}{\sum_{p=0}^M \Pr(\mathbf{Z}(t) | k_r(t) = p, \mathbf{X}(t), n) \pi_p(n) c(z_r^C, p)} \quad (73b)$$

Once we fix $k_r(t) = m$, the probability of observing z_r is independent of the other track-measurement associations. Therefore, we may decompose $\Pr(\mathbf{Z}(t) | k_r(t) = m, \mathbf{X}(t), n)$ into two parts:

$$\begin{aligned} \Pr(\mathbf{Z}(t) | k_r(t) = m, \mathbf{X}(t), n) \\ = \Pr(\mathbf{Z}(t) \setminus z_r(t) | \mathbf{X}(t) \setminus x_m(t), n) \cdot \Pr(z_r(t) | k_r(t) = m, x_m(t), n). \end{aligned} \quad (74)$$

$\mathbf{Z}(t) \setminus z_r(t)$ represents the set $\mathbf{Z}(t)$ without measurement $z_r(t)$. $\mathbf{X}(t) \setminus x_m(t)$ represents the set $\mathbf{X}(t)$ without the elements corresponding to track m . Let $f_{t,m}(q)$ be the PDF of the estimated target location at time t for target m given $\mathbf{X}(t)$. If target m is not clutter, then this is the Kalman filter estimate, which is normally distributed. This normal distribution comes directly from the Kalman filter in (9) and has a covariance $\mathbf{R}_r(t)$ equal to that associated with the measurement. If target m is clutter, i.e., $m = 0$, then $f_{t,0}(q)$ is the PDF of the clutter at point q . Usually this is assumed to be uniformly distributed over the field of view, but we shall designate it by $\mu(t, q)$ to allow for the use of a generic continuous distribution to be used. We shall assume that $\mu(t, q)$ is continuous as a function of q . Hence we obtain

$$f_{t,m}(q) = \begin{cases} \mu(t, q) & m = 0 \\ \mathcal{N}(q; \mathbf{H}_r \mathbf{X}(t), \mathbf{R}_r(t)) & m \neq 0 \end{cases} \quad (75)$$

Because the PDFs of the measurements coming from targets and those originating from clutter are assumed continuous, $\Pr(z_r | k_r(t) = m, \mathbf{X}(t), n)$ may be expressed as the probability of the observation being within a certain region around the observation as the size of that region approaches zero. We shall denote the size of this region as Δ and the region itself, which is centered about the observation z_r as $\Delta(z_r)$. Formulating this probability as a limit allows us to deal with zeros in the numerator and denominator of (74).

$$\Pr(z_r | k_r(t) = m, \mathbf{X}(t), n) = \lim_{\Delta \rightarrow 0} \int_{q \in \Delta(z_r)} f_{t,m}(q) dq \quad (76)$$

$$= \lim_{\Delta \rightarrow 0} f_{t,m}(z_r) \Delta. \quad (77)$$

Substituting (77) and (74) into (73b) we get

$$w_{m,r}(t) = \lim_{\Delta \rightarrow 0} \frac{\Pr(\mathbf{Z}(t) \setminus z_r(t) | \mathbf{X}(t) \setminus x_m(t), n) \cdot f_{t,m}(z_r) \cdot \Delta \pi_m(n) \cdot c(z_r^C, m)}{\sum_{p=0}^M \Pr(\mathbf{Z}(t) \setminus z_r(t) | \mathbf{X}(t) \setminus x_p(t), n) f_{t,p}(z_r) \Delta \pi_p(n) c(z_r^C, p)} \quad (78a)$$

$$= \frac{\Pr(\mathbf{Z}(t) \setminus z_r(t) | \mathbf{X}(t) \setminus x_m(t), n) \cdot f_{t,m}(z_r) \cdot \pi_m(n) \cdot c(z_r^C, m)}{\sum_{p=0}^M \Pr(\mathbf{Z}(t) \setminus z_r(t) | \mathbf{X}(t) \setminus x_p(t), n) f_{t,p}(z_r) \pi_p(n) c(z_r^C, p)} \quad (78b)$$

The jump from (78a) to (78b) was accomplished by noting that Δ could be factored out of the sums and products and thus cancels in the numerator and denominator.

Equation (78b) is the solution assuming that each target can produce only a single measurement. The evaluation of $\Pr(\mathbf{Z}(t) \setminus z_r(t) \mid \mathbf{X}(t) \setminus x_m(t), n)$ is combinatorially complex. However, under the PMHT measurement model, whereby a single target can produce any number of measurements at one time, this becomes simpler. Under the PMHT measurement model, (74) simplifies as follows:

$$\begin{aligned} \Pr(\mathbf{Z}(t) \mid k_r(t) = m, \mathbf{X}(t), n) \\ = \Pr(\mathbf{Z}(t) \setminus z_r(t) \mid \mathbf{X}(t), n) \cdot \Pr(z_r(t) \mid k_r(t) = m, x_m(t), n). \end{aligned} \quad (79)$$

This thus leaves a common term in (78b) that can be cancelled, giving us:

$$w_{m,r}(t) = \frac{f_{t,m}(z_r) \pi_m(n) c(z_r^C, m)}{\sum_{p=0}^M f_{t,p}(z_r) \pi_p(n) c(z_r^C, p)}. \quad (80)$$

Substituting the appropriate distributions for $f_{t,m}(z_r)$ and $f_{t,p}(z_r)$ gives us the form given in (17).

APPENDIX B. A DERIVATION OF THE PRIOR ASSOCIATION PROBABILITIES

$$\pi_m(n_r(s), t)$$

B.1. A General Derivation of the Prior Association Probabilities

Adopting the notation from the previous section, the prior association probabilities are defined as

$$\pi_m(n) = \Pr(k_r(t) = m \mid \mathbf{X}(t), \mathbf{C}, n). \quad (81)$$

In this case, we are suppressing the conditioning on $\mathbf{X}(t)$ and \mathbf{C} in the notation of π , because, assuming that the clutter is uniformly distributed in the viewing area, the actual location of the target has no bearing on the solution. In this section, we shall also suppress the conditioning on time in order to simplify notation. We shall derive the prior association probabilities assuming that no gating is taking place and that all sensors have the same field of view. It should be noted that the derivation of the π values are the embodiment of the PMHT measurement model. However, the PMHT measurement model is only an approximation. For this purpose we shall derive the π s based upon the usually more realistic model that a single sensor can only observe at most one measurement of each target at each time step.

The derivation of the π s takes place under the constraint that each measurement came either from a target or from clutter, that is:

$$\sum_{m=0}^M \pi_m(n) = 1. \quad (82)$$

Thus, solving for the value of the non-clutter π is sufficient to tell us the value of the clutter π for a particular n . Let us now determine the values of $\pi_m(n)$

given that $m > 0$. Using the Law of Total Probability to perform a decomposition based upon the number n_o , of targets observed we may thus write:

$$\pi_m(n) \triangleq \Pr(k_r = m \mid n) \quad (83a)$$

$$= \sum_{k=1}^{\min(n,M)} \Pr(k_r = m \mid n, n_o = k) \Pr(n_o = k \mid n). \quad (83b)$$

The Law of Total Probability may be used to simplify the first term of (83b) by adding conditioning upon whether observation z_r originated from a target. The first term of (83b) may be simplified as follows:

$$\begin{aligned} \Pr(k_r = m \mid n, n_o = k) \\ = \Pr(k_r = m \mid n, n_o = k, \mathbf{z}_r \in \mathbb{M}) \\ \times \Pr(\mathbf{z}_r \in \mathbb{M} \mid n, n_o = k). \end{aligned} \quad (84)$$

The second term in (84) can be found by counting: if k targets are observed, then the probability that a particular observation is a target is simply the ratio of the number of targets observed to the total number of observations. We thus have

$$\Pr(\mathbf{z}_r \in \mathbb{M} \mid n, n_o = k) = \frac{k}{n}. \quad (85)$$

The first term of (84) may be decomposed using the Law of Total Probability:

$$\begin{aligned} \Pr(k_r = m \mid n, n_o = k, \mathbf{z}_r \in \mathbb{M}) \\ = \Pr(k_r = m \mid n, n_o = k, \mathbf{z}_r \in \mathbb{M}, m \text{ observed}) \\ \times \Pr(m \text{ observed} \mid n, n_o = k, \mathbf{z}_r \in \mathbb{M}) \quad (86a) \\ = \left(\frac{1}{k}\right) \Pr(m \text{ observed} \mid n, n_o = k, \mathbf{z}_r \in \mathbb{M}). \end{aligned} \quad (86b)$$

The notation “ m observed” in the conditioning is an abbreviation for “the m th target was observed.” The jump from (86a) to (86b) was done by noting that if we know that target m was observed, observation r is a target and that k targets were observed, then by counting, we know that the association probability is $1/k$.

Let us define some additional notation. Let \mathbb{M} be the set of all measurements originating from a target, and p_m be the probability of detecting target m on a particular scan. There shall be no p_0 for clutter. We shall designate the set of all p_m as \mathbb{P}_d . There are $\binom{M}{k}$ ways of choosing which k targets are observed for each item in the sum. Let $\mathbb{P}_d(k)$ be the set of all products of k -combinations from \mathbb{P}_d without repetition. For example, if $k = 2$ and $M = 3$, then $\mathbb{P}_d(2) = \{p_1 p_2, p_1 p_3, p_2 p_3\}$ (the specific ordering of the terms is not important). Define $e_k(y)$ to be an enumerating function over \mathbb{P}_k , giving us the y th ordered element from $\mathbb{P}_d(k)$. $\mathbb{M}_k(y)$ shall represent the set of k targets whose detection probabilities are part of

$e_k(y)$, i.e., it is a set of observed targets. Let $\bar{e}_k(y)$ be, if $k < M$, the product of all $M - k$ elements of \mathbb{P}_d that are not used in $e_k(y)$, or $\bar{e}_k(y) = 1$ if $k = M$. For example, for $k = 2$ and $M = 3$, then $e_2(y)$ for $y = 1$ is $p_1 p_2$, for $y = 2$ is $p_1 p_3$, and for $y = 3$ is $p_2 p_3$. Likewise $\bar{e}_2(y)$ for each y is respectively $1 - p_3$, $1 - p_2$, and $1 - p_1$. Let $I(x)$ be an indicator function that is 1 if x is nonzero.

Equation (86b) is equal to the sum of the probabilities of all combinations of k observed targets such that the m th target is observed:

$$\begin{aligned} \Pr(m \text{ observed} \mid n, n_o = k, \mathbf{z}_r \in \mathbb{M}) \\ = \frac{\sum_{y=1}^{\binom{M}{k}} e_k(y) \bar{e}_k(y) I(m \in \mathbb{M}_k(y))}{\sum_{l=1}^{\binom{M}{k}} e_k(l) \bar{e}_k(l)}. \end{aligned} \quad (87)$$

Combining (87) with (86b) and (85) to form (84), we get the following solution to the first term of (83b):

$$\begin{aligned} \Pr(k_r = m \mid n, n_o = k) \\ = \left(\frac{\sum_{y=1}^{\binom{M}{k}} e_k(y) \bar{e}_k(y) I(m \in \mathbb{M}_k(y))}{\sum_{l=1}^{\binom{M}{k}} e_k(l) \bar{e}_k(l)} \right) \frac{1}{n}. \end{aligned} \quad (88)$$

The second term of (83b) may be simplified using Bayes' Theorem:

$$\Pr(n_o = k \mid n) = \frac{\Pr(n \mid n_o = k) \Pr(n_o = k)}{\sum_{j=0}^{\min(n, M)} \Pr(n \mid n_o = j) \Pr(n_o = j)}. \quad (89)$$

$\Pr(n \mid n_o = k)$ from (89) is the probability that there are $n - k$ clutter points. We shall designate this probability by the function $\xi(n - k)$. $\Pr(n_o = k)$ from (89) is the probability that k targets are observed and may be written as follows:

$$\Pr(n_o = k) = \sum_{y=1}^{\binom{M}{k}} e_k(y) \bar{e}_k(y). \quad (90)$$

Substituting (90) into (89), we get

$$\Pr(n_o = k \mid n) = \frac{\xi(n - k) \left(\sum_{y=1}^{\binom{M}{k}} e_k(y) \bar{e}_k(y) \right)}{\sum_{i=0}^{\min(n, M)} \xi(n - i) \left(\sum_{y=1}^{\binom{M}{i}} e_i(y) \bar{e}_i(y) \right)}. \quad (91)$$

Substituting (91) and (88) back into (83b), we get the following expression for $\pi_m(n)$:

$$\pi_m(n) = \begin{cases} 1 - \sum_{j=1}^M \pi_j(n) & m = 0 \\ \frac{\sum_{k=1}^{\min(n, M)} \xi(n - k) \sum_{y=1}^{\binom{M}{k}} e_k(y) \bar{e}_k(y) I(m \in \mathbb{M}_k(y))}{n \sum_{i=0}^{\min(n, M)} \xi(n - i) \left(\sum_{y=1}^{\binom{M}{i}} e_i(y) \bar{e}_i(y) \right)} & m \neq 0 \end{cases} \quad (92)$$

B.2. Simplifying the Prior Association Probabilities

Typically, the detection probability of the targets is unknown a priori. In this case, it is often simplest to assume that all of the targets have the same detection probability P_D , which would be chosen based upon the properties of the sensor and typical targets. When that is the case, $\Pr(n_o = k)$ from (90) is the same as the probability of having k successes out of M Bernoulli trials each with a success probability of P_D :

$$\Pr(n_o = k) = \binom{M}{k} P_D^k (1 - P_D)^{M-k}. \quad (93)$$

Additionally, $\Pr(m \text{ Observed} \mid n, n_o = k, \mathbf{z}_r \in \mathbb{M})$ from (86b) may easily be solved by counting. With M targets total and k targets observed, the probability of observing any particular target is just k/M . Hence we obtain

$$\Pr(m \text{ Observed} \mid n, n_o = k, \mathbf{z}_r \in \mathbb{M}) = \frac{k}{M}. \quad (94)$$

Using these simplifications, $\pi_m(n)$ is the same for all $m \neq 0$ and (92) may be written in the following simplified form:

$$\pi_m(n) = \begin{cases} 1 - M\pi_1(n) & m = 0 \\ \frac{\sum_{k=1}^{\min(n, M)} k \xi(n - k) \binom{M}{k} P_D^k (1 - P_D)^{M-k}}{Mn \sum_{i=0}^{\min(n, M)} \xi(n - i) \binom{M}{i} P_D^i (1 - P_D)^{M-i}} & m \neq 0 \end{cases}. \quad (95)$$

The probability $\xi(k)$ of observing k clutter points at time t is often modeled as a Poisson probability mass function with mean λV where λ represents the mean amount of clutter per unit volume and V is the volume of the viewing area. Thus, we have

$$\xi(k) = \frac{(\lambda V)^k}{k!} e^{-\lambda V}. \quad (96)$$

Substituting (96) into (95) for $m \neq 0$, we get

$$\pi_m(n) \Big|_{m \neq 0} = \frac{\sum_{k=1}^{\min(n, M)} \frac{k}{(\lambda V)^k (n - k)!} \binom{M}{k} P_D^k (1 - P_D)^{M-k}}{Mn \sum_{i=0}^{\min(n, M)} \frac{1}{(\lambda V)^i (n - i)!} \binom{M}{i} P_D^i (1 - P_D)^{M-i}}. \quad (97)$$

It can be noted that the ratio of consecutive terms of the sums in the numerator and denominator of (97) are ratios of polynomials in k . That is, the ratio of the a_{k+1} and the a_k th term of the sum in the numerator is

$$\frac{a_{k+1}}{a_k} = \frac{(k - M)(k - n)}{k} \left(\frac{P_D}{(1 - P_D)\lambda V} \right). \quad (98)$$

Likewise the ratio of the a_{k+1} and the a_k th term of the sum in the denominator is

$$\frac{a_{k+1}}{a_k} = \frac{(k - M)(k - n)}{k(k + 1)} \left(\frac{P_D}{(1 - P_D)\lambda V} \right). \quad (99)$$

TABLE II

Examples of $\pi_m(n)|_{m \neq 0}$ and $\bar{\pi}$ when all Targets have the Same Detection Probability and a Poisson Clutter Model is Used

M	$\pi_m(n) _{m \neq 0}$	$\bar{\pi}$
1	$\frac{P_D}{nP_D + (1 - P_D)\lambda V}$	$\left(\frac{1}{P_D} - 1\right) + n - 1$
2	$\frac{P_D^2(n - 1 - \lambda V) + P_D \lambda V}{P_D^2 n(n - 1) + 2nP_D \lambda V(1 - P_D) + (1 - P_D)^2 \lambda^2 V^2}$	$\frac{P_D(n - 1)}{P_D(1 - n + \lambda V) - \lambda V} + \left(\frac{1}{P_D} - 1\right) + n - 1$

Thus, following the method in [48], the sums may be rewritten in terms of hypergeometric functions

$$\pi_m(n) = \begin{cases} 1 - M\pi_1(n) & m = 0 \\ -\frac{{}_2F_0\left[1 - M, 1 - n; \frac{P_D}{(1 - P_D)\lambda V}\right]}{{}_2F_0\left[-M, -n; \frac{P_D}{(1 - P_D)\lambda V}\right]} & m \neq 0 \end{cases} \quad (100)$$

The function ${}_2F_0(a_1, a_2; z)$ in (100) is a hypergeometric function.

Table II shows $\pi_m(n)|_{m \neq 0}$ from (100) for one and two targets. Previous publications, such as [3], have also derived (100) for a single target under a Poisson clutter model.

APPENDIX C. FURTHER SIMPLIFICATIONS TO THE w s

If the probability of detection is the same for all targets, then the π s and the w s may be simplified one step further. Examining (95), it can be seen that all of the π values are the same for all targets. Using the simplified form of the w s in equation (17), we can divide the π term from the numerator into the denominator to get

$$w_{k_{r,s}(t),r}(t,s) = \frac{\mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_{k_{r,s}(t)}(t), \mathbf{R}_{r,s}(t)\}c(z_{r,s}^C(t), k_{r,s}(t))}{\bar{\pi}(n_t(s), t)\mu(t, \mathbf{z}_{r,s}(t))c(z_{r,s}^C(t), 0) + \sum_{m=1}^M \mathcal{N}\{\mathbf{z}_{r,s}(t); \hat{\mathbf{y}}_m(t), \mathbf{R}_{r,s}(t)\}c(z_{r,s}^C(t), m)} \quad (101)$$

This lowers the total number of multiplications needed and it is more efficient to calculate $\bar{\pi}$ than to calculate the π values for clutter and targets separately. When a Poisson clutter model is used, as is the case in (100), then, maintaining the notation from Section B, $\bar{\pi}$ is given as follows:

$$\bar{\pi} = \frac{\pi_0}{\pi_{m \neq 0}} \quad (102a)$$

$$= -M - \frac{{}_2F_0\left[-M, -n; \frac{P_D}{(1 - P_D)\lambda V}\right]}{{}_2F_0\left[1 - M, 1 - n; \frac{P_D}{(1 - P_D)\lambda V}\right]} \quad (102b)$$

Some expressions for $\bar{\pi}$ as a function of the number of targets are given in Table II.

APPENDIX D. THE KALMAN SMOOTHER

In this section, we summarize the equations for the Kalman smoother, as described in [4]. $\mathbf{x}(t_1 | t_2)$ represents the state estimate at time t_1 given all observations from time 1 to t_2 , whereby $\mathbf{x}(0 | 0)$ is the initial estimate. $\mathbf{P}(t_1 | t_2)$ is the covariance of the aforementioned state estimate. $\mathbf{F}(t)$ is the state transition matrix and $\mathbf{H}(t)$ the measurement matrix, as shown in the dynamic equations in (8) and (9). In keeping with the dynamic model, $\mathbf{Q}(t)$ is the process noise covariance and $\mathbf{R}(t)$ the measurement noise covariance, the noises being zero-mean and white. $\mathbf{z}(t)$ is the observation at time t and $\mathbf{R}(t)$ is its covariance. The Kalman smoother consists of a Kalman filtering step, after which a smoothing step is applied. Letting \mathbf{I} be the identity matrix, the Kalman filter equations to calculate $\mathbf{x}(t | t)$ from $\mathbf{x}(t - 1 | t - 1)$ and the observations at time t are given by

$$\mathbf{x}(t | t - 1) = \mathbf{F}(t)\mathbf{x}(t - 1 | t - 1), \quad (103)$$

$$\hat{\mathbf{y}}(t) = \mathbf{H}(t)\mathbf{x}(t | t - 1), \quad (104)$$

$$\mathbf{P}(t | t - 1) = \mathbf{F}(t)\mathbf{P}(t - 1 | t - 1)\mathbf{F}(t)' + \mathbf{Q}(t), \quad (105)$$

$$\mathbf{W}(t) = \mathbf{P}(t | t - 1)\mathbf{H}(t)'[\mathbf{R}(t) + \mathbf{H}(t)\mathbf{P}(t | t - 1)\mathbf{H}(t)']^{-1}, \quad (106)$$

$$\mathbf{P}(t | t) = [\mathbf{I} - \mathbf{W}(t)\mathbf{H}(t)]\mathbf{P}(t | t - 1)[\mathbf{I} - \mathbf{W}(t)\mathbf{H}(t)]' + \mathbf{W}(t)\mathbf{R}(t)\mathbf{W}(t), \quad (107)$$

and

$$\mathbf{x}(t | t) = \mathbf{x}(t | t - 1) + \mathbf{W}(t)[\mathbf{z}(t) - \hat{\mathbf{y}}(t)]. \quad (108)$$

Assuming that there are N time-periods of data available, the smoothing is performed by starting with the final estimate $\mathbf{x}(N | N)$ and going backwards along the previous estimates, smoothing them using the following

equations

$$\mathbf{C}(t) = \mathbf{P}(t | t)\mathbf{F}(t)'\mathbf{P}(t + 1 | t)^{-1}, \quad (109)$$

$$\mathbf{x}(t | N) = \mathbf{x}(t | t) + \mathbf{C}(t)[\mathbf{x}(t + 1 | N) - \mathbf{x}(t + 1 | t)], \quad (110)$$

and

$$\mathbf{P}(t | N) = \mathbf{P}(t | t) + \mathbf{C}(t)[\mathbf{P}(t + 1 | N) - \mathbf{P}(t + 1 | t)]\mathbf{C}(t)'. \quad (111)$$

REFERENCES

- [1] K. M. Alexiev
Multiple target tracking using Hough transform PMHT algorithm.
In *First International IEEE Symposium on "Intelligent Systems"*, vol. 1, 2002, 227–232.
- [2] D. Avitzour
A maximum likelihood approach to data association.
IEEE Transactions on Aerospace and Electronic Systems, **28**, 2 (Apr. 1992), 560–566.
- [3] Y. Bar-Shalom and X-R. Li
Multitarget-Multisensor Tracking: Principles and Techniques.
Storrs, CT: YBS Publishing, 1995.
- [4] Y. Bar-Shalom, X-R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation.
New York: John Wiley & Sons, 2001.
- [5] D. P. Bertsekas
Nonlinear Programming (2nd ed.).
Belmont, MA.: Athena Scientific, 2003, 107–108.
- [6] W. R. Blanding, P. Willett, R. L. Streit, and D. Dunham
Consistent covariance estimation for PMHT.
In *Proceedings of SPIE: Signal and Data Processing of Small Targets Conference*, vol. 6699, San Diego, CA, Aug. 2007, 66 990I-1–66 990I-11.
- [7] W. R. Blanding, P. Willett, R. L. Streit, and D. Dunham
Multi-frame assignment PMHT that accounts for missed detections.
In *Proceedings of the 10th International Conference on Information Fusion*, Quebec, Canada, July 2007.
- [8] R. A. Boyles
On the convergence of the EM algorithm.
Journal of the Royal Statistical Society. Series B (Methodological), **45**, 1 (1983), 47–50.
- [9] S. J. Davey
Extensions to the probabilistic multi-hypothesis tracker for improved data association.
Ph.D. dissertation, The University of Adelaide, Adelaide, South Australia, Sept. 2003.
- [10] S. J. Davey
Simultaneous localization and map building using the probabilistic multi-hypothesis tracker.
IEEE Transactions on Robotics, **23**, 2 (Apr. 2007), 271–280.
- [11] S. J. Davey
Tracking possibly unresolved targets with PMHT.
In *Information, Decision and Control, 2007*, Feb. 2007, 47–52.
- [12] S. J. Davey and D. A. Gray
A comparison of track initiation methods with the PMHT.
In *Information, Decision and Control*, I.E.E.E. Press, Mar. 2002, this paper is found in the printed book; only a conference schedule is on the IEEE's web site.
- [13] S. J. Davey and D. A. Gray
Integrated track maintenance for the PMHT via the hysteresis model.
IEEE Transactions on Aerospace and Electronic Systems, **37**, 1 (Jan. 2007), 93–111.
- [14] S. J. Davey, D. A. Gray, and R. L. Streit
Tracking, association, and classification: A combined PMHT approach.
Digital Signal Processing, **12**, 2–3 (2002), 372–382.
- [15] S. J. Davey, M. G. Rutten, and B. Cheung
A comparison of detection performance for several track-before detect algorithms.
EURASIP Journal on Advances in Signal Processing, **2008** (2008), article ID 428036.
- [16] A. P. Dempster, N. M. Laird, and D. B. Rubin
Maximum likelihood from incomplete data via the EM algorithm.
Journal of the Royal Statistical Society. Series B (Methodological), **39**, 1 (1977), 1–38.
- [17] D. T. Dunham and R. G. Hutchins
Hybrid tracking algorithm using MHT and PMHT.
In *Proceedings of SPIE Conference: Signal and Data Processing of Small Targets*, vol. 4728, Orlando, FL, Apr. 2002, 166–175.
- [18] M. Efe, Y. Ruan, and P. Willett
Probabilistic multi-hypothesis tracker-addressing some basic issues.
In *IEEE Proceedings on Radar, Sonar and Navigation*, vol. 151, no. 4, Aug. 2004, 189–196.
- [19] M. Efe, Y. Ruan, and P. Willett
The pedestrian PMHT.
In *Proceedings of the 5th International Conference on Information Fusion*, vol. 2, Annapolis, MD, 2002, 838–845.
- [20] R. Frühwirth and A. Strandlie
Track fitting with ambiguities and noise: A study of elastic tracking and non-linear filters.
Computer Physics Communications, **120**, 2 (Aug. 1999), 197–214.
- [21] Q. Gan and C. Harris
Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion.
IEEE Transactions on Aerospace and Electronic Systems, **37**, 1 (Jan. 2001), 273–279.
- [22] H. Gauvrit, J. P. Le Cadre, and C. Jauffret
A formulation of multitarget tracking as an incomplete data problem.
IEEE Transactions on Aerospace and Electronic Systems, **33**, 4 (Oct. 1997), 1242–1257.
- [23] E. Giannopoulos, R. L. Streit, and P. Swaszek
Multi-target track segment bearings-only association and ranging.
In *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems & Computers*, vol. 2, Pacific Grove, CA, Nov. 1997, 1336–1340.
- [24] E. Giannopoulos, R. L. Streit, and P. Swaszek
Probabilistic multi-hypothesis tracking in a multi-sensor multi-target environment.
In *ADFS '96, First Australian Data Fusion Symposium*, Adelaide, SA, Australia, Nov. 1996, 184–189.
- [25] K. Gilholm, S. Godsill, S. Maskell, and D. Salmond
Poisson models for extended target and group tracking.
In *Proceedings of SPIE: Signal and Data Processing of Small Targets Conference*, vol. 5913, San Diego, CA, Aug. 2005, 59 130R-1–59 130R-12.
- [26] C. G. Hempel
The effects of sensor registration error on the performance of PMHT.
In *Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, June 2008, 1975–1979.
- [27] C. Hue, J-P. Le Cadre, and P. Pérez
Tracking multiple objects with particle filtering.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 3 (July 2002), 791–812.

- [28] C. Hue, J-P. Le Cadre, and P. Pérez
Sequential monte carlo methods for multiple target tracking and data fusion.
IEEE Transactions on Signal Processing, **50**, 2 (Feb. 2002), 309–325.
- [29] R. G. Hutchins and D. T. Dunhan
Evaluation of a probabilistic multihypothesis tracking algorithm in cluttered environments.
In *Conference Record of the Thirtieth Asilomar Conference on Signals, Systems and Computers*, vol. 2, Nov. 1996, 1260–1264.
- [30] C. F. Jeff Wu
On the convergence properties of the EM algorithm.
The Annals of Statistics, **11**, 1 (1983), 95–103.
- [31] H. Jeong and J-H. Park
An EM-based adaptive multiple target tracking filter.
International Journal of Adaptive Control and Signal Processing, **16** (2002), 1–23.
- [32] M. Kalandros and L. Y. Pao
Multisensor covariance control strategies for reducing bias effects in interacting target scenarios.
IEEE Transactions on Aerospace and Electronic Systems, **41**, 1 (Jan. 2005), 153–173.
- [33] M. L. Krieg and D. A. Gray
Track fusion in the presence of an interference.
In *Fourth International Symposium on Signal Processing and Its Applications, 1996 (ISSPA '96)*, vol. 1, Aug. 1996, 192–195.
- [34] A. Logothetis, V. Krishnamurthy, and J. Holst
On maneuvering target tracking via the PMHT.
In *Proceedings of the 36th Conference on Decision and Control*, vol. 5, San Diego, CA, Dec. 1997, 5024–5029.
- [35] A. Logothetis, V. Krishnamurthy, and J. Holst
A Bayesian EM algorithm for optimal tracking of a maneuvering target in clutter.
Signal Processing, **82**, 3 (Mar. 2002), 473–490.
- [36] Luginbuhl, P. Ainsleigh, S. Mathews, and R. L. Streit
Accurate likelihood evaluation for multiple model PMHT algorithms.
In *Aerospace Conference 2008*, Mar. 2008.
- [37] T. Luginbuhl and P. Willett
Tracking a general, frequency modulated signal in noise.
In *Proceedings of the 38th Conference on and Control*, Phoenix, AZ, Dec. 1999.
- [38] T. Luginbuhl and P. Willett
Estimating the parameters of general frequency modulated signals.
IEEE Transactions on Signal Processing, **52**, 1 (Jan. 2004), 117–131.
- [39] T. Luginbuhl, Y. Sun, and P. Willett
A track management system for the PMHT algorithm.
In *Proceedings of the 4th International Conference on Information Fusion*, Montreal, Canada, 2001.
- [40] G. J. McLachlan and T. Krishnan
The EM Algorithm and Extensions.
New York: John Wiley & Sons, Inc., 1997.
- [41] K. J. Molnar and J. W. Modestino
Application of the EM algorithm for the multitarget/multi-sensor tracking problem.
IEEE Transactions on Signal Processing, **46**, 1 (Jan. 1998), 115–129.
- [42] T. Moon
The expectation-maximization algorithm.
IEEE Signal Processing Magazine, **13**, 6 (Nov. 1996), 47–60.
- [43] D. Mušicki and X. Wang
Track management and PMHT.
In *Proceedings of the 10th International Conference on Information Fusion*, Quebec, Canada, July 2007, 1–5.
- [44] A. G. Pakfiliz and M. Efe
Multi-target tracking in clutter with histogram probabilistic multi-hypothesis tracker.
In *Proceedings of the IEEE 13th Signal Processing and Communications Applications Conference*, May 2005, 167–168.
- [45] L. Y. Pao and C. W. Frei
A comparison of parallel and sequential implementations of a multisensor multitarget tracking algorithm.
In *Proceedings of the American Control Conference*, vol. 3, Seattle, WA, June 1995, 1683–1687.
- [46] L. Y. Pao and R. M. Powers
A comparison of several different approaches for target tracking with clutter.
In *Proceedings of the 2003 American Control Conference*, vol. 5, Denver, CO, June 2003, 3919–3924.
- [47] L. Y. Pao and L. Trailović
The optimal order of processing sensor information in sequential multisensor fusion algorithms.
IEEE Transactions on Automatic Control, **45**, 8 (Aug. 2000), 1532–1536.
- [48] M. Petkovšek, H. S. Wilf, and D. Zeilberger
A = B.
Wellesley, MA: A K Peters, Ltd., 1996, ch. 3.
- [49] G. W. Pulford and B. F. La Scala
MAP estimation of target maneuver sequence with the expectation-maximization algorithm.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 2 (Apr. 2002), 367–377.
- [50] G. Pulford and A. Logothetis
An expectation-maximisation tracker for multiple observations of a single target in clutter.
In *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 5, Dec. 1997, 4997–5003.
- [51] C. Rago, P. Willett, and R. L. Streit
Direct data fusion using the PMHT.
In *Proceedings of the American Control Conference*, vol. 3, Seattle, WA, June 1995, 1698–1702.
- [52] C. Rago, P. Willett, and R. L. Streit
A modified PMHT.
In *Proceedings of the 1995 Conference on Information Sciences and Systems*, Mar. 1995.
- [53] C. Rago, P. Willett, and R. Streit
A comparison of the JPDAF and PMHT tracking algorithms.
In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-95)*, vol. 5, Detroit, MI, May 1995, 3571–3574.
- [54] R. A. Redner and H. F. Walker
Mixture densities, maximum likelihood and the EM algorithm.
SIAM Review, **26**, 2 (Apr. 1984), 195–239.
- [55] A. Roche
EM algorithm and variants: An informal tutorial.
CEA, Tech. Rep., 2003.
- [56] Y. Ruan and P. Willett
The turbo PMHT.
IEEE Transactions on Aerospace and Electronic Systems, **40**, 4 (Oct. 2004), 1388–1398.
- [57] A. Strandlie and J. Zerubia
Particle tracking with iterated Kalman filters and smoothers: the PMHT algorithm.
Computer Physics Communications, **123**, 1 (Dec. 1999), 77–86.
- [58] R. L. Streit
Tracking on intensity-modulated data streams.
NUWC-NPT, Tech. Rep. 11,221, May 2000.

- [59] R. L. Streit
PMHT algorithms for multi-frame assignment.
In *Proceedings of the 9th International Conference on Information Fusion*, Florence, Italy, July 2006.
- [60] R. L. Streit and T. E. Luginbuhl
A probabilistic multi-hypothesis tracking algorithm without enumeration and pruning.
In *Proceedings of the Sixth Joint Service Data Fusion Symposium*, Laurel, Maryland, June 1993, 1015–1024.
- [61] R. L. Streit and T. E. Luginbuhl
Probabilistic multi-hypothesis tracking.
NUWC-NPT, Tech. Rep. 10,428, Feb. 1995.
- [62] N. Ueda and R. Nakano
Deterministic annealing EM algorithm.
Neural Networks, **11**, 2 (Mar. 1998), 271–282.
- [63] M. Verhaegen
Improved understanding of the loss-of-symmetry phenomenon in the conventional Kalman filter.
NASA, Tech. Rep. Technical Memorandum 100041, Dec. 1987.
- [64] M. J. Walsh, M. L. Graham, R. L. Streit, T. Luginbuhl, and L. Mathews
Tracking on intensity-modulated sensor data streams.
In *Proceedings of the IEEE Aerospace Conference*, vol. 4, Big Sky, MT, 2001, 4/1901–4/1909.
- [65] M. J. Walsh
Computing the observed information matrix for dynamic mixture models.
NUWC-NPT, Tech. Rep. 11,768, Sept. 2006.
- [66] M. Wieneke and W. Koch
The PMHT: Solutions for some of its problems.
In *Proceedings of SPIE Conference: Signal and Data Processing of Small Targets Conference*, vol. 6699, San Diego, CA, Aug. 2007, 1–12.
- [67] M. Wieneke and W. Koch
On sequential track extraction within the PMHT framework.
EURASIP Journal on Advances in Signal Processing, **2008** (2008), article ID 276914.
- [68] M. Wieneke and P. Willett
On track-management within the PMHT framework.
In *Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, June 2008, 736–743.
- [69] P. Willett, Y. Ruan, and R. Streit
The PMHT for maneuvering targets.
In *Proceedings of SPIE: Signal and Data Processing of Small Targets Conference*, vol. 3373, Orlando, FL, Apr. 1998, 416–427.
- [70] P. Willett, Y. Ruan, and R. L. Streit
A variety of PMHTs.
In *Proceedings of the ISIS/PMHT Workshop*, Paris, France, Nov. 1998.
- [71] P. Willett, Y. Ruan, and R. L. Streit
PMHT: Problems and some solutions.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 3 (July 2002), 738–754.



David F. Crouse (S'05) received the B.S. and M.S. degrees in electrical engineering, in 2005 and 2008, respectively, from the University of Connecticut (UConn), Storrs, and the B.A. degree in German from the University of Connecticut in 2006, for which he spent a year with the Ruprecht-Karls-Universität, Heidelberg, Germany.

He is currently pursuing the Ph.D. degree in electrical engineering at UConn. His research interests lie in the areas of statistical signal processing and tracking.



Marco Guerriero (S'06) was born in Salerno, Italy, on June 18th, 1981. He received his B.A.Sc. and M.Sc. (electrical engineering) from the University of Salerno, Italy in 2002 and 2005 respectively. During his M.Sc. he also spent a semester at Delft University of Technology (TUDELFT), Netherlands, where he took graduate courses in electrical engineering.

He is currently working towards his Ph.D. degree in electrical engineering at the University of Connecticut (UConn), Storrs. In the summer of 2007, he was as a visiting scientist at the NATO Undersea Research Centre (NURC) La Spezia, Italy. His research interests lie in the areas of signal processing, with particular focus on distributed detection and estimation in sensor networks, target tracking and data fusion.



Peter Willett (S'83—M'86—SM'97—F'03) received the B.Sc. degree in engineering science in 1982 from the University of Toronto, Toronto, Canada. He received the M.E. and M.S. degrees in 1983 and 1984, respectively, and the Ph.D. degree in 1986, all in electrical engineering, from Princeton University, Princeton, NJ.

He is a professor at the University of Connecticut, Storrs, where he has worked since 1986. His interests are generally in the areas of detection theory, target tracking, and signal processing.

Dr. Willett is an associate editor for the *IEEE Transactions on Systems, Man, and Cybernetics* and the Editor-in-Chief for the *IEEE Transactions on Aerospace and Electronic Systems*.

Multitarget Multisensor Tracking in the Presence of Wakes

ANDERS RØDNINGSBY
YAAKOV BAR-SHALOM
ODDVAR HALLINGSTAD
JOHN GLATTETRE

In this paper we focus on targets which, in addition to reflecting signals themselves, also have a trailing path behind them, called a wake, which causes additional detections. When the detections are fed to a tracking system like the probabilistic data association filter (PDAF), the estimated track can be misled and sometimes lose the real target because of the wake. This problem becomes even more severe in multitarget environments where targets are operating close to each other in the presence of wakes. To prevent this, we have developed a probabilistic model of the wakes in a multitarget environment. This model is used to augment the joint probabilistic data association filter (JPDAF) for both coupled and decoupled filtering.

This paper provides a systematic comparison of the standard data association filters (PDAF and JPDAF) and their modified versions presented here in a multitarget multisensor environment. Simulations of two targets with wakes in four different scenarios show that this modification gives good results and the probability of lost tracks is significantly reduced. The targets are observed by two sensors and it is shown that tracks estimated in a centralized fusion configuration are better than those from the local sensors. It is also shown that applying the wake model to targets that do not generate a wake, yields almost no deterioration of the tracking performance.

Manuscript received October 8, 2008; revised February 16, 2009; released for publication June 24, 2009.

Refereeing of this contribution was handled by Dr. Yvo Boers.

Authors' addresses: A. Rødningsby and O. Hallingstad, University Graduate Center, NO-2027 Kjeller, Norway; Y. Bar-Shalom, University of Connecticut, U-2157 Storrs, CT; J. Glattetre, Kongsberg Maritime, NO-3191 Horten, Norway.

1557-6418/09/\$17.00 © 2009 JAIF

1. INTRODUCTION

Targets in real tracking scenarios may be detected by their reflection of signals emitted from a radar [6], a sonar [26], or by the use of optical sensors [24]. In addition to target-originated measurements there will also be a number of detections due to noise and clutter, called false alarms. A well-known tracking method to handle targets in clutter is the probabilistic data association filter (PDAF) [3, ch. 3.4]. The PDAF accounts for the measurement origin uncertainty by calculating for each validated measurement at the current time the association probabilities to the target of interest.

In a multitarget environment [5] the association of measurements is more problematic because the individual targets no longer can be considered separately as in the PDAF. For this purpose the joint probabilistic data association filter (JPDAF) [3, ch. 6.2], [14] was developed to consider a known number of targets in the data association simultaneously. This method evaluates the measurement-to-target association probabilities for the latest set of measurements and then combines them into the state estimates. In the JPDAF the targets' states, conditioned on the past, are assumed independently distributed so that filtering can be done decoupled. As an alternative, the targets' states, given the past, can be considered as correlated. This leads to the joint probabilistic data association coupled filter (JPDAF) [2], [3, pp. 328–329], where the correlation between the targets' estimation errors is accounted for. A modified version of the JPDAF, called coupled data association filter (CPDA), was presented in [9] to also account for partial target detections. In this paper an equivalent filter to the CPDA, but where the covariance calculation is in symmetrical form (to avoid numerical problems experienced by the CPDA), is modified to also account for targets in the presence of wakes. This filter is called modified JPDAF.

A more powerful source of false measurements than those due to noise and clutter, is the wake phenomenon that appears behind certain targets. This could be air bubbles from a diver, the wake behind a ship, or the wake from ballistic vehicles in the re-entry stage. One possible approach to this problem is to handle both the target and the wake behind it as an extended target. A problem with this approach is the varying and unknown size of the wake which may reach far behind the target yielding a large bias. In this paper the wake is not considered as part of the target, but rather as a special kind of clutter. When these measurements are fed to the tracking system, it becomes important to associate them correctly to prevent a lost track. In [1] a probabilistic editing method is used to handle the wake-dominated measurements in the tracking algorithm. This probabilistic editing method is based on a single measurement extracted for each time step, and that this measurement originates from either the tar-

get or the wake. In [21] a modified PDAF is developed to handle false measurements originating from the bubbles behind a diver (the wake). This modified single target tracking method does not restrict the number of false measurements for each time step, but assumes a set of measurements where each false measurement originates from either random clutter or the wake. In this paper we extend the modified PDAF to handle multiple targets in the presence of wakes. A probabilistic wake model is used for each target in the multitarget environment that has a wake behind it. These single wake models are combined to form a joint wake model, and the modified JPDAF and JPDAF are developed to incorporate this additional joint wake model.

In recent years there has been an extensive interest in using multiple sensors in surveillance systems. This leads to data fusion where there exist several possible configurations [3, ch. 8.2]. Primarily due to the bandwidth constraints in real systems, it is sometimes not feasible to transmit all measurement information to a fusion center (centralized configuration). Instead, only local estimates are transmitted to a fusion center (at a reduced rate), and a track-to-track association followed by track fusion is carried out (decentralized configuration). However, the best performance is achieved using the centralized configurations where all measurements are transmitted from the local sensors to a fusion center. In this paper we use the centralized configuration with sequential filtering [3, p. 88] where the global estimate is updated by the measurements from each local sensor, one sensor at the time.

In Section 2 the tracking problem in the presence of a wake is reviewed for a single target. In Section 3 the modified JPDAF is developed for a multitarget environment, and the modified version of the JPDAF, which accounts for partial target detections, is derived. In Section 4 a brief review of multisensor tracking is given. The data association methods are then compared in Section 5 by simulations of two targets with wakes in four different multisensor scenarios, before conclusions are given in Section 6.

2. BACKGROUND

2.1. Model of Tracking

The standard discrete linear model in tracking is

$$x_{k+1} = Fx_k + v_k, \quad z_k = Hx_k + w_k \quad (1)$$

where

x :	target state	F :	transition matrix
z :	measurement	H :	measurement matrix
v :	process noise	w :	measurement noise
k :	time index		

The process and measurement noises are assumed independent, white and Gaussian with covariance matrices

$$E\{v_k v_k^T\} = Q \quad \text{and} \quad E\{w_k w_k^T\} = R. \quad (2)$$

For this system, a Kalman filter is optimal as long as the measurement z_k originates from the target at each time k . In many real world problems this is unfortunately not true due to the presence of false measurements originating from noise and clutter. Instead, a set of m_k measurements $Z_k = \{z_k(1), z_k(2), \dots, z_k(m_k)\}$ is available at time k so that data association is needed. A simple and efficient method to solve this problem is the PDAF.

2.2. Standard PDAF

The approach of the PDAF is to calculate the association probabilities for each validated measurement (that falls in a gate around the predicted measurement) at the current time to the target of interest. The posterior track probability density is therefore a mixture of Gaussian probability density functions (pdf), but is then forced back to Gaussianity by moment-matching for the succeeding scan. For a derivation of the PDAF see [3, ch. 3.4], and in the following a brief overview of the PDAF will be given.

Assume that the target state at time $k-1$ is estimated as $\hat{x}_{k-1|k-1}$ with associated covariance $P_{k-1|k-1}$. This means that the estimate is conditioned on the entire past up to time $k-1$. Then the following assumptions are made:

- The track is already initialized.
- The past information about the target is summarized approximately by the Gaussian pdf

$$p(x_k | Z^{k-1}) \approx \mathcal{N}(x_k; \hat{x}_{k|k-1}, P_{k|k-1}) \quad (3)$$

where

$$Z^{k-1} = \{Z_0, Z_1, \dots, Z_{k-1}\}. \quad (4)$$

- A validation region or gate is set up for each time step to select the candidate measurements for association.
- At time k there are m_k validated measurements but at most one of them can be target-originated. The rest are assumed to be due to i.i.d. uniformly spatially distributed false alarms, independently across time.
- Detections of the real target occur independently over time with known detection probability P_D .

At each time k , the algorithm goes through the following steps:

- Predict the target state, associated covariance and measurement at time k based on the estimates at $k-1$:

$$\begin{aligned} \hat{x}_{k|k-1} &= F\hat{x}_{k-1|k-1} \\ P_{k|k-1} &= FP_{k-1|k-1}F^T + Q \\ \hat{z}_{k|k-1} &= H\hat{x}_{k|k-1}. \end{aligned} \quad (5)$$

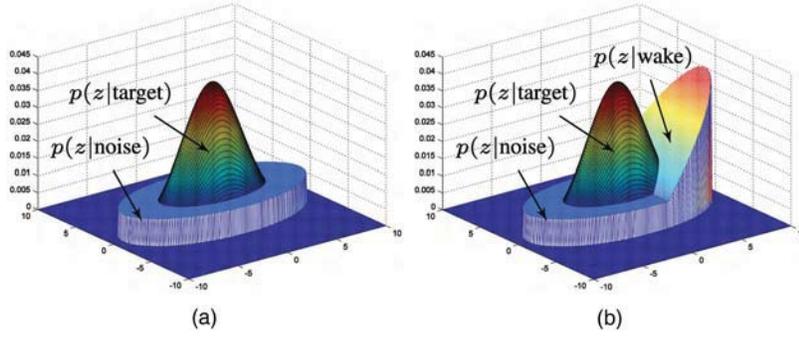


Fig. 1. Illustration of the pdfs for the measurements originating from the target, noise or wake in: (a) regular PDAF, (b) modified PDAF with wake model.

2) Compute the innovation covariance for the true (target-originated) measurement

$$S_k = HP_{k|k-1}H^T + R \quad (6)$$

and use S_k to form the measurement validation gate where the validated measurements Z_k result in m_k innovations:

$$\nu_k(i) = z_k(i) - \hat{z}_{k|k-1} \quad i = 1, \dots, m_k, \quad (7)$$

3) Calculate the association probabilities $\beta_k(i)$, $i = 1, \dots, m_k$ that measurement $z_k(i)$ originates from the true target, and $\beta_k(0)$ as the probability that all measurements are false alarms

$$\beta_k(i) = \begin{cases} ce^{-(1/2)\nu_k(i)^T S_k^{-1} \nu_k(i)} & i = 1, \dots, m_k \\ c|2\pi S_k|^{1/2} m_k \frac{1 - P_G P_D}{V_k P_D} & i = 0 \end{cases} \quad (8)$$

Here c is a normalizing constant to ensure that $\sum_{i=0}^{m_k} \beta_k(i) = 1$, V_k is the volume of the gate and P_G is the probability that the true measurement falls inside the gate. In (8) a diffuse prior [3, p.135] is used for the point mass function (pmf) of the number of false measurements in the validation region.

4) Calculate the Kalman gain and the combined innovation

$$W_k = P_{k|k-1}H^T S_k^{-1} \quad \text{and} \quad \nu_k = \sum_{i=1}^{m_k} \beta_k(i) \nu_k(i) \quad (9)$$

to update the track according to

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + W_k \nu_k. \quad (10)$$

5) The state estimation covariance is updated by

$$\begin{aligned} P_{k|k} = & \beta_k(0)P_{k|k-1} \\ & + [1 - \beta_k(0)](P_{k|k-1} - W_k S_k W_k^T) \\ & + W_k \left[\sum_{i=0}^{m_k} \beta_k(i) \nu_k(i) \nu_k(i)^T - \nu_k \nu_k^T \right] W_k^T \end{aligned} \quad (11)$$

where the last term in (11) is the ‘‘spread of the innovations.’’

2.3. Modified PDAF

Targets with a wake behind them may cause detections from the wake that mislead the tracking algorithm and are likely to result in a lost track. This is because the uniform distribution assumption for the false measurements (assumption (d) in Section 2.2) is violated. To prevent this, an extension of the regular PDAF incorporating a special probabilistic model of the wake was developed in [21]. The PDAF with the wake model is illustrated in Fig. 1. The modified PDAF takes into account that the false measurements can originate from either the wake with pdf $p_W(\cdot)$ with a priori probability P_W , or from i.i.d. uniformly distributed noise/clutter with a priori probability $1 - P_W$, independently across time. This modification affects the PDA in the calculation of the $\beta_k(i)$ in (8) and yields

$$\beta_k(i) = \begin{cases} c \frac{e^{-(1/2)\nu_k(i)^T S_k^{-1} \nu_k(i)}}{V_k \left[\frac{1 - P_W}{V_k} + \frac{P_W}{P_{GW}} p_W(z_k(i)) \right]} & i = 1, \dots, m_k \\ c|2\pi S_k|^{1/2} m_k \frac{1 - P_G P_D}{V_k P_D} & i = 0 \end{cases} \quad (12)$$

The bracketed parenthesis in the denominator in $\beta_k(i)$ for $i = 1, \dots, m_k$ is the pdf of a false measurement

$$\begin{aligned} p(z_k(i) | \text{measurement } i \text{ is false}) \\ = \frac{1 - P_W}{V_k} + \frac{P_W}{P_{GW}} p_W(z_k(i)) \end{aligned} \quad (13)$$

where P_{GW} is used to account for restricting the density of the wake model $p_W(z_k(i))$ to the validation gate. The calculation of P_{GW} for a linear $p_W(\cdot)$ is presented in detail in [21]. As expected, in the limit as P_W goes to zero, (12) becomes (8).

2.4. Track Formation and Termination

The data association filters discussed above assume that the track is already initialized, and when a track

is established, there are no included rules for how to terminate the track. Hence, procedures for formation and termination of tracks are necessary. A simple and common method to initialize tracks is the two-point differencing method [4, p. 247]. Any successive pair of detections within a maximum distance based on target maximum motion parameters and measurement noise variances initiates a preliminary track. This preliminary track, containing the initial state and the corresponding covariance, can now initialize the PDAF. To reduce the amount of false tracks, a “ p/q ” logic-based track formation procedure can be used. In this procedure a preliminary track has to receive measurements for a minimum of p time steps during the first q scans to become valid.

To terminate a track a logic suitable for the application is needed, and a set of rules has to be made. The rules used in this paper, called termination events, are described in Section 5.4. It should also be noted that in some filters, such as the integrated probabilistic data association filter (IPDAF) [18] or the version of the interacting multiple model probabilistic data association filter (IMMPDAF) presented in [3, ch. 4.4], the track formation and termination are included.

3. PROBABILISTIC DATA ASSOCIATION FOR MULTIPLE TARGETS IN THE PRESENCE OF WAKES

In a multitarget environment the data association algorithm needs to handle situations where a measurement could originate from different targets. For this purpose, the JPDAF was developed, and a derivation of this standard algorithm is given in [3, ch. 6.2]. Another problem arises when these targets have wakes behind them that result in misleading wake detections. In this section we will modify the JPDAF to handle this problem.

3.1. Assumptions

Assume there is a known number N_T of established targets at time $k - 1$. Notice that these targets are already initialized, e.g., by the method in Section 2.2. For each target t , where $t = 1, \dots, N_T$, the target state is estimated as $\hat{x}_{k-1|k-1}^t$ with associated covariance $P_{k-1|k-1}^t$. Then the following assumptions are made:

- a) Measurements from one target can fall in the validation gate of a neighboring target.
- b) The past information about target t is summarized approximately by the Gaussian pdf

$$p(x_k^t | Z^{k-1}) \approx \mathcal{N}(x_k^t; \hat{x}_{k|k-1}^t, P_{k|k-1}^t). \quad (14)$$

- c) At time k there are m_k validated measurements in the union of their validation gates, but for each target t at most one measurement can be target-originated. The rest are assumed to be due to the wakes with pdf $p_W(\cdot)$ with a priori probability P_W , or from i.i.d. uniformly

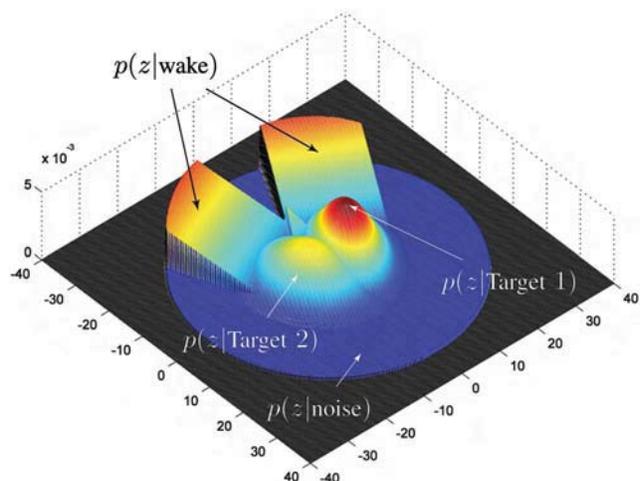


Fig. 2. Probability density functions for two targets with crossing trajectories. The distributions of the targets are Gaussian and overlap each other. Each wake behind the two targets is modeled as a pdf, linearly increasing from the target and backwards, and the sum of each single target’s wake model forms the joint wake model. The noise/clutter is uniformly spatially distributed inside the joint validation region.

distributed noise/clutter with a priori probability $1 - P_W$, independently across time.

In Fig. 2 an example of the pdfs for two targets that are starting to cross each other is shown. Here both targets have a wake behind them, and the joint wake model (the sum of each target’s single wake model) increases linearly behind the targets inside the joint validation region. The joint validation region contains all the candidate measurements, and restricts the spatially uniform distribution representing the noise/clutter. It should be noted that the linearly increasing wake models are not developed to approach the true density of the wake since the wake density would seemingly be higher close to the targets rather than farther away. Such an approach would easily misassociate true target-originated measurements as wake-originated ones. At the same time, in practice, a false wake-originated measurement is less detrimental when it is very close to the true target than farther behind. The adopted wake model is therefore a pragmatic approach to let the probability of having a wake-originated measurement instead of a target-originated one increase with the distance behind the true target. Further details about the joint wake model and the validation region are given in Appendix A.

3.2. Joint Association Events

Define the validation matrix Ω to represent all feasible association events at time k (the time index k is omitted for simplicity where it does not cause confusion)

$$\Omega = [\omega(j, t)] \quad j = 1, \dots, m \quad \text{and} \quad t = 0, \dots, N_T. \quad (15)$$

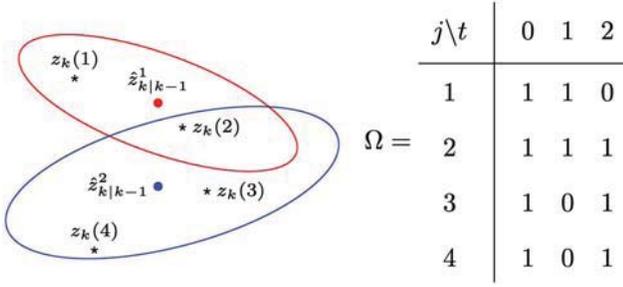


Fig. 3. Two targets with a measurement in the intersection of their validation gates are shown with corresponding validation matrix Ω .

Here, $\omega(j, t)$ is a binary element indicating whether measurement j lies in the validation gate of target t . The index $t = 0$ means that the measurement is from none of the targets and therefore it is a false measurement. An example where a measurement may originate from either of two targets, i.e., it lies in both targets' validation gates, is shown with the corresponding validation matrix Ω in Fig. 3. For all these possible joint association events, conditional probabilities have to be derived.

A joint association event Θ describes an unambiguous association between the measurements and the targets at time k

$$\Theta = \bigcap_{j=1}^m \theta(j, t_j) \quad (16)$$

where

- $\theta(j, t_j)$ is the event that measurement j originates from target t_j .
- t_j is the index of the target to which measurement j is associated in the event under consideration.

The event Θ can also be represented by the matrix

$$\Omega_{\Theta} = [\omega_{\Theta}(j, t)] \quad (17)$$

consisting of the units in Ω corresponding to the associations in Θ

$$\omega_{\Theta}(j, t) = \begin{cases} 1 & \text{if the event } \theta(j, t) \text{ is part of } \Theta \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

Using this, a feasible association event needs to fulfill the following requirements:

- 1) A measurement can have only one source, i.e.

$$\sum_{t=0}^{N_T} \omega_{\Theta}(j, t) = 1 \quad \forall j. \quad (19)$$

- 2) At most one measurement can originate from a target

$$\delta_{\Theta}^t \triangleq \sum_{j=1}^m \omega_{\Theta}(j, t) \leq 1 \quad t = 1, \dots, N_T. \quad (20)$$

The binary variable δ_{Θ}^t is called the target detection indicator since it indicates whether a measurement is

associated to a target t or not in event Θ . It is also convenient to define two more binary variables

$$\tau_{\Theta}(j) \triangleq \sum_{t=1}^{N_T} \omega_{\Theta}(j, t) \quad (21)$$

$$\phi_{\Theta} \triangleq \sum_{j=1}^m [1 - \tau_{\Theta}(j)] \quad (22)$$

where $\tau_{\Theta}(j)$ is the measurement association indicator to indicate if measurement j is associated to a target or not, and ϕ_{Θ} is the number of false (unassociated) measurements in event Θ .

3.3. Modified JPDAF with a Wake Model

The joint association event probabilities are derived using Bayes' formula

$$\begin{aligned} P\{\Theta_k | Z^k\} &= P\{\Theta_k | Z_k, m_k, Z^{k-1}\} \\ &= \frac{1}{c} p[Z_k | \Theta_k, m_k, Z^{k-1}] P\{\Theta_k | Z^{k-1}, m_k\} \\ &= \frac{1}{c} p[Z_k | \Theta_k, m_k, Z^{k-1}] P\{\Theta_k | m_k\} \end{aligned} \quad (23)$$

where c is a normalizing constant. In the last line of the above equation the irrelevant conditioning term Z^{k-1} has been omitted. The pdf of the measurements in (23) is derived by assuming that the states of the targets, conditioned on the past observations, are mutually independent

$$p[Z_k | \Theta_k, m_k, Z^{k-1}] = \prod_{j=1}^{m_k} p[z_k(j) | \theta_k(j, t_j), Z^{k-1}]. \quad (24)$$

Measurements not associated to a target are assumed either from the wakes with pdf $p_W(z_k(j))$ with a priori probability P_W , or from uniformly distributed noise/clutter with a priori probability $(1 - P_W)$. Defining V_k as the volume of the joint validation gate, the pdf of a measurement given its origin is

$$p[z_k(j) | \theta_k(j, t_j), Z^{k-1}] = \begin{cases} \mathcal{N}[z_k(j); \hat{z}_{k|k-1}^{t_j}, S_k^{t_j}] & \text{if } \tau_{\Theta_k}(j) = 1 \\ P_W \frac{p_W(z_k(j))}{P_{GW}} + (1 - P_W) \frac{1}{V_k} & \text{if } \tau_{\Theta_k}(j) = 0 \end{cases} \quad (25)$$

where $\hat{z}_{k|k-1}^{t_j}$ is the predicted measurement for target t_j with associated innovation covariance $S_k^{t_j}$. The constant P_{GW} is used for restricting $p_W(z_k(j))$ to the joint validation region, and has an analytical expression derived in Appendix A. Using the above equation, (24) can be

written as

$$\begin{aligned}
& p[Z_k | \Theta_k, m_k, Z^{k-1}] \\
&= \prod_{j=1}^{m_k} \{ \mathcal{N}[z_k(j); \hat{z}_{k|k-1}^j, S_k^j] \}^{\tau_{\Theta}(j)} \\
&\times \left\{ P_W \frac{p_W(z_k(j))}{P_{GW}} + (1 - P_W) \frac{1}{V_k} \right\}^{1 - \tau_{\Theta}(j)}.
\end{aligned} \tag{26}$$

Next, the last term in (23) will be derived. Let δ_{Θ} be the vector of detection indicators corresponding to event Θ_k

$$\delta_{\Theta} = [\delta_{\Theta}^1, \dots, \delta_{\Theta}^{N_T}]. \tag{27}$$

The vector δ_{Θ} and the number of false measurements ϕ_{Θ} follow from the event Θ under consideration. Using the definition of conditional probabilities [20, p. 28], this yields

$$\begin{aligned}
P\{\Theta_k | m_k\} &= P\{\Theta_k, \delta_{\Theta}, \phi_{\Theta} | m_k\} \\
&= P\{\Theta_k | \delta_{\Theta}, \phi_{\Theta}, m_k\} P\{\delta_{\Theta}, \phi_{\Theta} | m_k\}.
\end{aligned} \tag{28}$$

The first term in (28) is obtained using combinatorics:

1) In event Θ_k there are assumed $m_k - \phi_{\Theta}$ targets detected.

2) The number of events Θ_k , where the same targets are detected, is given by the number of ways of associating $m_k - \phi_{\Theta}$ measurements to the detected targets from a set of m_k measurements.

By assuming each such event a priori equally likely, one has

$$P\{\Theta_k | \delta_{\Theta}, \phi_{\Theta}, m_k\} = \frac{1}{m_k P_{m_k - \phi_{\Theta}}} = \frac{\phi_{\Theta}!}{m_k!}. \tag{29}$$

The last term in (28) is, assuming δ and ϕ independent,

$$P\{\delta_{\Theta}, \phi_{\Theta} | m_k\} = \prod_{t=1}^{N_T} (P_D^t)^{\delta_{\Theta}^t} (1 - P_D^t)^{1 - \delta_{\Theta}^t} \mu_F(\phi_{\Theta}) \tag{30}$$

where P_D^t is the detection probability of target t and $\mu_F(\phi_{\Theta})$ is the prior pmf of the number of false measurements. The indicators δ_{Θ}^t have been used to select the probabilities of detection and no detection events according to the event Θ_k under consideration. Combining (29) and (30) into (28) yields the prior probability of a joint association event

$$P\{\Theta_k | m_k\} = \frac{\phi_{\Theta}!}{m_k!} \prod_{t=1}^{N_T} (P_D^t)^{\delta_{\Theta}^t} (1 - P_D^t)^{1 - \delta_{\Theta}^t} \mu_F(\phi_{\Theta}). \tag{31}$$

The pmf of the number of false measurements $\mu_F(\phi)$ can, as in the case of the PDA, have two versions, parametric or nonparametric.

1) Parametric JPDA uses a Poisson pmf

$$\mu_F(\phi) = e^{-\lambda V} \frac{(\lambda V)^{\phi}}{\phi!} \tag{32}$$

which requires the spatial density λ of the false measurements.

2) Nonparametric JPDA uses a diffuse prior

$$\mu_F(\phi) = \epsilon \quad \forall \phi \tag{33}$$

which does not require the parameter λ .

Using the nonparametric model and combining (31) and (26) into (23) yields the joint association event probabilities

$$\begin{aligned}
P\{\Theta_k | Z^k\} &= \frac{\phi_{\Theta}!}{c} \prod_{t=1}^{N_T} (P_D^t)^{\delta_{\Theta}^t} (1 - P_D^t)^{1 - \delta_{\Theta}^t} \\
&\times \prod_{j=1}^{m_k} \{ \mathcal{N}[z_k(j); \hat{z}_{k|k-1}^j, S_k^j] \}^{\tau_{\Theta}(j)} \\
&\times \left\{ P_W \frac{p_W(z_k(j))}{P_{GW}} + (1 - P_W) \frac{1}{V_k} \right\}^{1 - \tau_{\Theta}(j)}
\end{aligned} \tag{34}$$

where the constants ϵ and $m_k!$ are brought into the normalization constant c . For comparison, the joint association event probabilities derived in [3, p. 318] for the standard JPDAF is

$$\begin{aligned}
P\{\Theta_k | Z^k\} &= \frac{\phi_{\Theta}! \cdot V_k^{-\phi_{\Theta}}}{c} \prod_{t=1}^{N_T} (P_D^t)^{\delta_{\Theta}^t} (1 - P_D^t)^{1 - \delta_{\Theta}^t} \\
&\times \prod_{j=1}^{m_k} \{ \mathcal{N}[z_k(j); \hat{z}_{k|k-1}^j, S_k^j] \}^{\tau_{\Theta}(j)}
\end{aligned} \tag{35}$$

where the third line in (34) is replaced by $V_k^{-\phi_{\Theta}}$. As for the modified PDAF, (34) reduces to (35) in the limit as P_W goes to zero. Finally, marginal association probabilities are obtained by summing over all the joint association events in which the marginal event of interest occurs

$$\beta_k^t(j) \triangleq P\{\theta_k(j, t) | Z^k\} = \sum_{\Theta_k} P\{\Theta_k | Z^k\} \omega_{\Theta}(j, t) \tag{36}$$

$$\beta_k^t(0) \triangleq 1 - \sum_{j=1}^{m_k} \beta_k^t(j). \tag{37}$$

By using these association probabilities in (8), the state estimation equations are exactly the same as in the PDAF, (5)–(11).

3.4. Modified JPDAF

The state estimation above is based on the assumption that the targets, conditioned on the past observations, are mutually independent. When measurements

are inside the validation gates for two or more targets at the same time, we say that the targets are “sharing” measurements. For targets that share measurements for several sampling times, a dependence of their estimation error ensues, and this can be taken into account by calculating the resulting error correlations [7]. The resulting JPDAF algorithm [3, pp. 328–329] does the filtering in a coupled manner, yielding a covariance matrix with cross-covariances that reflect the correlation between the targets’ state estimation errors. The effectiveness of the JPDAF approach in combination with the IMM was demonstrated on splitting targets in [2]. This JPDAF approach does not account for situations with partial target detections since the association events where all targets are detected are not separated from events where only some of them are detected. The association events need to be separated in groups where the group member events have the same vector of detection indicators δ_Θ , see (27). This situation was accounted for in the CPDA filter, derived in [9], where the CPDA in combination with hypothesis pruning was developed to avoid track coalescence. In our simulations, however, the CPDA approach did lead to numerical problems in the covariance calculations. An equivalent solution to the CPDA, but where the covariance calculation is in a symmetrical form, is therefore developed and used in this paper to avoid numerical problems. The modified JPDAF accounting for partial target detections and the presence of wakes, is derived next.

Assuming only two targets, the stacked state vector and its associated covariance are denoted as

$$\hat{x}_{k|k-1}^S = \begin{bmatrix} \hat{x}_{k|k-1}^1 \\ \hat{x}_{k|k-1}^2 \end{bmatrix} \quad \text{and} \quad P_{k|k-1}^S = \begin{bmatrix} P_{k|k-1}^1 & P_{k|k-1}^{12} \\ P_{k|k-1}^{21} & P_{k|k-1}^2 \end{bmatrix} \quad (38)$$

where $P_{k|k-1}^{12}$ is the cross-covariance between target 1 and 2. This cross-covariance will be zero before these targets become coupled, i.e., start to share measurements. The updated state estimate is

$$\hat{x}_{k|k}^S = \hat{x}_{k|k-1}^S + \sum_{\Theta_k} P\{\Theta_k | Z^k\} I_\Theta^x W_k^S I_\Theta^z \nu_k^S(\Theta) \quad (39)$$

where

$$\nu_k^S(\Theta) = z_k^S(\Theta) - \hat{z}_{k|k-1}^S \quad (40)$$

$$z_k^S(\Theta) = \begin{bmatrix} z_k(j_1(\Theta)) \\ z_k(j_2(\Theta)) \end{bmatrix} \quad (41)$$

$$\hat{z}_{k|k-1}^S = H^S \hat{x}_{k|k-1}^S \quad (42)$$

and $j_t(\Theta_k)$ is the index of the measurement associated with target t in the event Θ_k at time k . The filter gain in (39) is

$$W_k^S = P_{k|k-1}^S H^{ST} [H^S P_{k|k-1}^S H^{ST} + R^S]^{-1} \quad (43)$$

where

$$H^S = \begin{bmatrix} H^1 & 0 \\ 0 & H^2 \end{bmatrix} \quad \text{and} \quad R^S = \begin{bmatrix} R^1 & 0 \\ 0 & R^2 \end{bmatrix}. \quad (44)$$

The matrices I_Θ^x and I_Θ^z in (39) are used to choose only the innovation from the target(s) that are detected, given by the detection indicator in (20), such that

$$I_\Theta^x = \begin{bmatrix} \delta_\Theta^1 I_{n_x} & 0 \\ 0 & \delta_\Theta^2 I_{n_x} \end{bmatrix} \quad (45)$$

$$I_\Theta^z = \begin{bmatrix} \delta_\Theta^1 I_{n_z} & 0 \\ 0 & \delta_\Theta^2 I_{n_z} \end{bmatrix}. \quad (46)$$

Here, I_{n_x} and I_{n_z} are $n_x \times n_x$ and $n_z \times n_z$ identity matrices, where n_x and n_z are the dimensions of a single target state vector and a single target measurement, respectively. Notice that if a target is undetected in the joint association event Θ_k under consideration, the corresponding part of the innovation vector needs to be set to zero even though I_Θ^x is multiplied to the Kalman gain W_k^S . This is accomplished by I_Θ^z .

The updated stacked covariance $P_{k|k}^S$, conditioned on all measurements up to time k , Z^k , is derived in Appendix B and yields

$$\begin{aligned} P_{k|k}^S &= \sum_{\Theta_k} P\{\Theta_k | Z^k\} \\ &\times \{I_\Theta^x W_k^S I_\Theta^z (\nu_k^S(\Theta) \nu_k^S(\Theta)^T + R^S) I_\Theta^z W_k^{ST} I_\Theta^x \\ &\quad + (I - I_\Theta^x W_k^S I_\Theta^z H^S) P_{k|k-1}^S (I - I_\Theta^x W_k^S I_\Theta^z H^S)^T\} \\ &- \left(\sum_{\Theta_k} P\{\Theta_k | Z^k\} I_\Theta^x W_k^S I_\Theta^z \nu_k^S(\Theta) \right) \\ &\times \left(\sum_{\Theta_k} P\{\Theta_k | Z^k\} I_\Theta^x W_k^S I_\Theta^z \nu_k^S(\Theta) \right)^T. \end{aligned} \quad (47)$$

The joint association event probabilities $P\{\Theta_k | Z^k\}$ are calculated as for the decoupled filter in Section 3.3, and the prediction step is as in (5), but with stacked state and covariance.

4. MULTISENSOR TRACKING

The best performance in multisensor data fusion is achieved using centralized configurations where all measurements are transmitted from the local sensors to a fusion center.¹ Primarily due to the bandwidth constraints in real systems, the centralized configuration is sometimes not feasible because its requirement to transmit all measurement information to a fusion center. This is the motivation for the interest in decentralized track-

¹It is assumed that the sensors are properly registered and have no biases.

ing, with track-to-track association followed by track fusion, which has been compared to centralized tracking in [10], [11]. To make the centralized tracking more feasible for real systems, the measurement data can be compressed in the local sensors before they are transmitted [12]. When the measurements are transmitted to a fusion center in the centralized tracking, there are two different schemes for the way the state is updated. In parallel filtering the measurements from all sensors (if synchronized) are taken into account at the same time. The other alternative is sequential filtering where measurements from each sensor is processed one sensor at a time. The first sensor updates the state (and covariance) based on predictions from the previous time step as in a single-sensor algorithm. Then, this new updated state is used as a zero-time prediction to update with the measurements from the second sensor and so on. In [19] the sequential and parallel filtering schemes are compared in a multisensor JPDAF approach, and it is shown that sequential filtering is less computationally expensive as the number of sensors increases. According to [19], the sequential method yields better tracking performance on the average when data association is needed. This is primarily due to the fact that better filtered estimates are available after processing each sensor's data.

Another problem regarding multisensor systems is the positioning of the sensors, where there are several aspects to consider:

- The sensors' joint ability to cover the required area.
- The sensor specifications.
- The most likely target locations and trajectories.
- The possibility of tracking the targets from various view angles.

These factors, among others, have to be considered separately and in light of the main purpose of each specific tracking problem.

5. SIMULATIONS AND RESULTS

In this section the data association methods described previously (PDAF, Modified PDAF, JPDAF, Modified JPDAF and Modified JPDAF) are compared in four different multitarget simulation scenarios in the presence of wakes. These simulations consider an underwater surveillance system with active sonar sensors and scuba divers as the targets. The wakes are generated by the air bubbles from the divers. Results are shown using two sensors, working both as independent single sensors and together in a centralized tracking system. When the filters discussed above are used in multisensor (MS) situations in the centralized tracking configuration, they will be denoted as MSPDAF, Modified MSPDAF, MSJPDAF, Modified MSJPDAF and Modified MSJPDAF.

5.1. Simulation Scenarios

The four simulation scenarios are shown in Fig. 4, and are in the sequel denoted as:

1) *Crossing scenario*: The targets are starting in positions (25,32.5) m and (25,67.5) m with speed 1 m/s and course according to the trajectory crossing angle $\gamma = 20^\circ$, see Fig. 4. The nearly straight trajectories are crossing the 200 s run midway. In [22] a similar scenario with varying trajectory crossing angle $\gamma = [5^\circ, 6^\circ, \dots, 30^\circ]$ is simulated for a single sensor, showing significant reduction of track loss for the modified filters.

2) *Parallel scenario*: The targets are starting in positions (25,40) m and (25,60) m with speed 1 m/s and course according to the trajectory crossing angle $\gamma = 15^\circ$. When the distance between the targets is less than 3 m, their velocities are both set to [1,0] m/s, creating parallel trajectories with 3 m spacing. Then, after 130 s they separate in the same way as they joined each other.

3) *Sequential scenario*: The targets are starting in positions (22.5,40) m and (27.5,60) m with speed 1 m/s and course according to the trajectory crossing angle $\gamma = 15^\circ$. When the distance between the targets is less than 0.5 m in the y -direction their velocities are both set to [1,0] m/s. Since the first target started 5 m behind the second target in the x -direction, they will now move after each other in the same direction with about 5 m spacing. Then, after 130 s they separate in the same way as they joined each other. Note that Target 1 is moving inside the wake created by Target 2 before they separate.

4) *Meeting scenario*: The targets are starting in positions (25,50) m and (225,50) m with speed 1 m/s and course directly towards each other. The targets are passing each other without changing course. Note that both targets are moving inside the wake of the other one after the passing.

5.2. Simulation Setup

Two sensors, with the same specifications, are located in the positions (0,0) m and (250,100) m respectively. The sensors have 180° field of view with resolution about 0.7° in bearing (256 non-overlapping beams) and 0.2 m in range. Their maximum range of 250 m is assumed large enough to cover the targets throughout the 200 s long runs, consisting of 200 scans with sampling period $T = 1$ s. For both targets a two-dimensional direct discrete time nearly constant velocity model [4] is used in (1) and (2):

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (48)$$

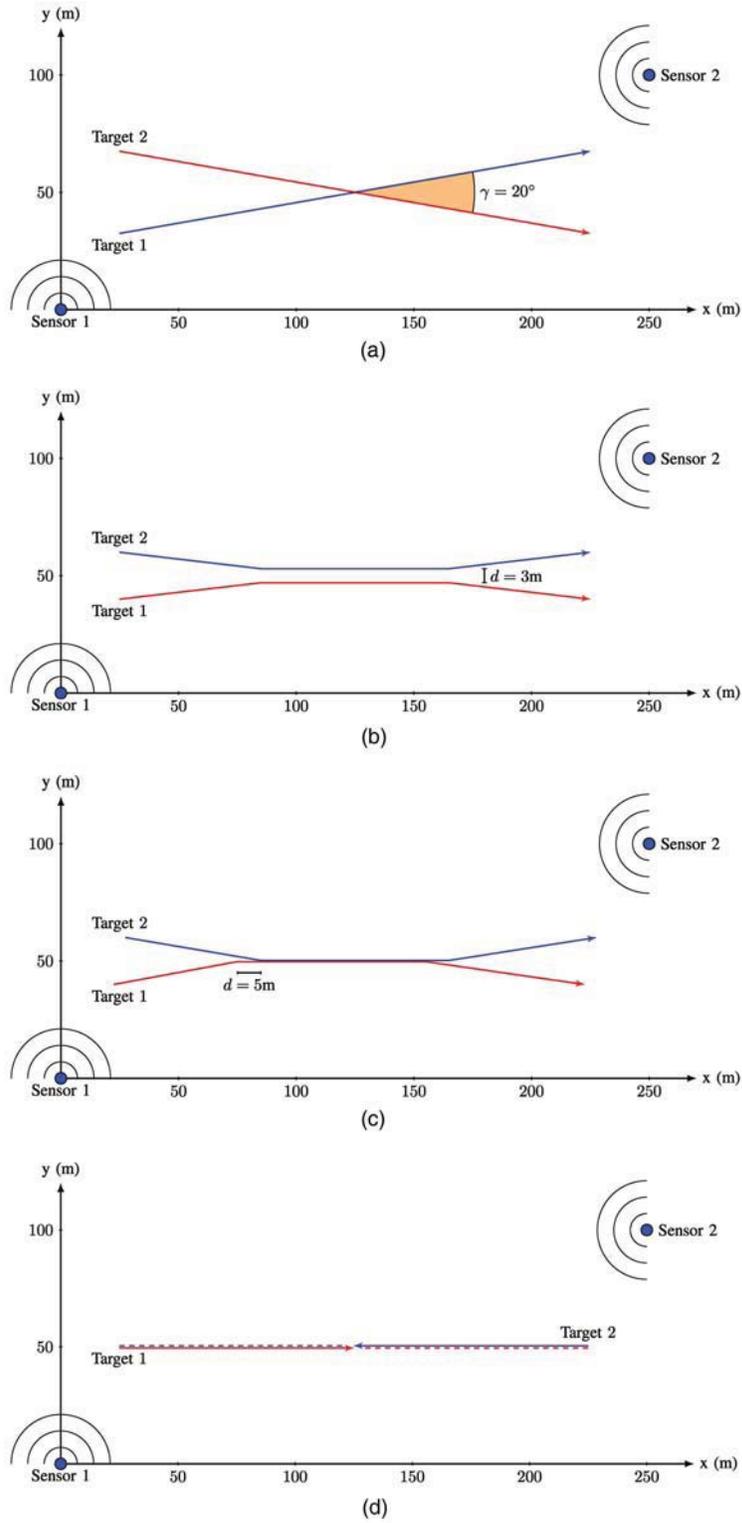


Fig. 4. Simulation scenarios of two targets observed by two sensors. Four different scenarios are shown. (a) Scenario 1: Crossing trajectories with trajectory crossing angle $\gamma = 20^\circ$. (b) Scenario 2: Parallel trajectories where the targets are moving side by side with spacing $d = 3\text{ m}$. (c) Scenario 3: Sequential trajectories where Target 1 is moving behind in the wake created by Target 2, with spacing $d = 5\text{ m}$. (d) Scenario 4: Meeting trajectories where the targets are moving towards each other, and passing each other inside the wake of the other target.

$$R_c = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{bmatrix} \quad R_p = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\psi^2 \end{bmatrix} \quad (49)$$

$$Q = \begin{bmatrix} \frac{1}{4}T^4 & \frac{1}{2}T^3 & 0 & 0 \\ \frac{1}{2}T^3 & T^2 & 0 & 0 \\ 0 & 0 & \frac{1}{4}T^4 & \frac{1}{2}T^3 \\ 0 & 0 & \frac{1}{2}T^3 & T^2 \end{bmatrix} \sigma_p^2. \quad (50)$$

TABLE I
Specification of Parameters

Parameter	Value	Specification
T	1.0 s	Sampling period
P_D	0.7	Detection probability
P_G	0.99999	Gate probability
P_W	0.9	Wake probability
P_{FA}	0.001	False alarm probability
$\sigma_p^2(s)$	$(0.001 \text{ m/s}^2)^2$	Process noise (simulation model)
$\sigma_p^2(f)$	$(0.05 \text{ m/s}^2)^2$	Process noise (filter model)
σ_r^2	$(0.2 \text{ m})^2$	Measurement noise (range)
σ_ψ^2	$(3.5 \cdot 10^{-3} \text{ rad})^2$	Measurement noise (bearing)
N	256×1250	Number of resolution cells
S	$180^\circ, 250 \text{ m range}$	Sensor coverage area
M	$250 \times 40 \text{ m}$	Measurement generation area
W	$5 \times 30 \text{ m}$	Wake area
V_S	98174 m^2	Volume of S
V_M	10000 m^2	Volume of M
V_W	150 m^2	Volume of W
λ_{clutter}	16.3	Expected number of correlated clutter measurements

The parameters in (48)–(50) and other simulation design parameters are given in Table I.

Originally, the position measurements are in polar coordinates (r, ψ) with (time invariant) measurement noise covariance R_p , but are transformed to Cartesian coordinates (x, y) with corresponding measurement noise covariance R_c using the standard conversion [4, pp. 397–399]. This results in a purely linear model so that a Kalman filter can be used in the tracking algorithm. The measurement noise matrix R_p is calculated assuming a uniformly distributed position error inside the resolution cell. Hence, the variance of the uniformly distributed error is given by the resolution, and this variance is heuristically used as the variance in the Gaussian distributed R_p

$$\sigma_r^2 = \frac{0.2^2}{12} \text{ m}^2 \quad \text{and} \quad \sigma_\psi^2 = \frac{(\pi/256)^2}{12} \text{ rad}^2. \quad (51)$$

Due to the high resolution in range (0.2 m), the targets will cover several resolution cells in the range direction, resulting in extended targets. Because of this, the actual range resolution is used as the standard deviation ($\sigma_r = 0.2 \text{ m}$) instead of the calculation in (51). This modification of σ_r in the simulations seems more reasonable since the targets (scuba divers) are extended in the range direction. To ensure controlled trajectories for the true targets, the added process noise in the simulation model $\sigma_p^2(s) = (0.001 \text{ m/s}^2)^2$ is set low, but not to zero. The process noise in the filter model $\sigma_p^2(f) = (0.05 \text{ m/s}^2)^2$ is set to approximate about 5 cm/s change in the velocity components between each scan.

When the targets are following after each other in the sequential scenario, there will be a problem using the filter modifications as described above. This problem

especially affects the target following behind the first target, because there will be wake detections surrounding this target both in front and behind it. If the wake model is used in this situation, the wake-dominated measurements behind the target will get lower weights than the wake-dominated measurements in front. These measurements in front, which originate from the wake of the first target, will mislead the tracker, and the estimated track will speed up until it catches up with the target in front. It is therefore likely that this target will be lost. An approach to prevent this is to only apply the wake model to the target in front, and use a regular data association filter for the target that is following the first one. By handling the two targets separately in two single-target tracking filters, the track of the target behind the first one will have better chance to survive in this hard situation. In the simulations a target following behind another one will therefore not use the wake model if the following criteria are fulfilled:

- 1) The target is inside the wake area W of the target in front. The wake area W is defined as a rectangle, L_w wide and reaching L_l backwards from the target ($L_w = 5 \text{ m}$, $L_l = 30 \text{ m}$).
- 2) The target is at least 2 m behind the target in front.
- 3) The difference between the moving direction of the following target and the target in front is less than 10° .

To reduce the computational load, the different versions of the multitarget tracking algorithms are substituted with their analogous single target tracking algorithms as long as targets are not “sharing” measurements. In other words, the standard PDAF is used instead of the JPDAF, and the modified PDAF is used instead of the modified JPDAF and JPDAF when the targets are apart. The multisensor (MS) filters are treated in the same way, i.e., the MSPDAF is used instead of the MSJPDAF.

5.3. Measurement Generation

The directional information (bearing) in an active sonar is given by the beamforming. Since no beamforming can achieve an ideal directivity pattern, there will be a leakage or scattering of the signal in one beam to the neighboring beams [16, ch. 5.3]. This is also known as the point spread function (psf) [25], and may yield detections from a point target in more than one bearing cell. In [22] the true target-originated measurements are simulated as single point detections, which, as described above, is a simplification of the real world. To generate measurements from the targets and their wakes in this paper, detections from a *real data set* of a scuba diver with an open breathing system are used. The data set consists of 500 scans, and is recorded by an active sonar with the same specifications as the sensors used in the simulations. The diver is swimming in a nearly straight

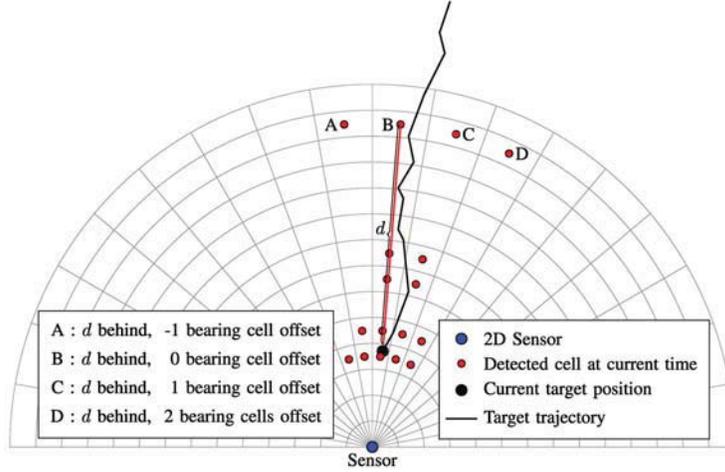


Fig. 5. Illustration of how each detection is specified by using the distance behind the target d and a bearing offset. The bearing offset describes the number of cells in the bearing direction between a detected cell and the cell where the target trajectory passes through, and with the same range as the detected one. As an example, the four detections marked with A, B, C and D are the same distance behind the target, but with offsets $-1, 0, 1$ and 2 respectively.

line, and its trajectory is estimated mainly by using a modified PDAF [21], but some manual corrections are done to get better position estimates. For each scan a cell averaging–constant false alarm rate (CA-CFAR) detector [15] is used to obtain the detections. The parameters of the CA-CFAR algorithm are the same as in [21], except for the following parameters:

- The average false alarm rate (probability of a false detection in a resolution cell) is set to $P_{FA} = 0.001$.
- The size of the averaging window used to estimate the local background noise parameter is increased to 51 cells in the range direction due to the increased resolution of the sensors used in this paper.

For each scan, the detections are stored and specified by a distance d behind the true target position and a bearing offset, see Fig. 5. The bearing offset describes the number of cells in the bearing direction between a detected cell and the cell where the target trajectory passes through, and with the same range as the detected one. As an example, the four detections marked with A, B, C and D in Fig. 5 are the same distance behind the target, but with bearing offsets $-1, 0, 1$ and 2 respectively. Finally, after going through the 500 scans in the *real data set*, this gives 500 different sets of detections of the true target and its wake, where the scattering in the bearing-direction is accounted for. In the simulations the detections originating from the target and its wake are generated by drawing from these 500 sets according to a first order Markov model. If set s was drawn at scan k , the probability of drawing the succeeding set $s + 1$ at time $k + 1$ is $\pi_{s,s+1} = 0.7$, and the probability of a random drawing $u \in [1, 500]$ (uniformly distributed) is $\pi_{s,u} = 1 - \pi_{s,s+1} = 0.3$. The targets' states are generated directly from (1), and with the position and velocity known, the target and wake originated measurements are added.

Another part of the measurements is the clutter or false measurements, and a standard assumption in simulations is that clutter is uniformly distributed in the surveillance area. In this paper the generation of clutter is done in two steps. The first step is under the standard assumption, where the probability of generating a clutter measurement in a resolution cell is $P_{FA}/2 = 0.005$, uniformly distributed across all cells in range and bearing. The second step is to generate spatially correlated clutter. These measurements are generated from a multimodal Gaussian pdf with equal weights for the different modes. This is an approach to reflect that some areas in the surveillance region yields more clutter, due to, e.g., a rough surface of the sea bed, banks, hills, large stones and other objects that creates variation in the surveillance area. The multimodal Gaussian pdf is regenerated for each run, and the number of modes is drawn as a uniform discrete variable between 1 and 10. The mean of each Gaussian mode is drawn uniformly in the surveillance area, and the covariance matrix is diagonal with standard deviations in the x and y directions drawn as uniform variables between 0 and 10. The number of correlated clutter measurements for each scan is Poisson distributed with parameter $\lambda_{clutter}$. Denote the coverage area for a sensor as S ($180^\circ, 250$ m range), and the measurement area covering the full trajectories of the targets as M (250×40 m), with volume V_S and V_M respectively. The Poisson parameter $\lambda_{clutter}$ is then given by

$$\lambda_{clutter} = 0.5P_{FA}N \frac{V_M}{V_S} \approx 16.3 \quad (52)$$

where P_{FA} is the probability of a false alarm in a resolution cell, and N is the number of resolution cells in S . Hence, in average there will be 16.3 correlated clutter measurements in M for each scan. An example of all measurements in one time frame for the crossing scenario is given in Fig. 6. Notice that the detections

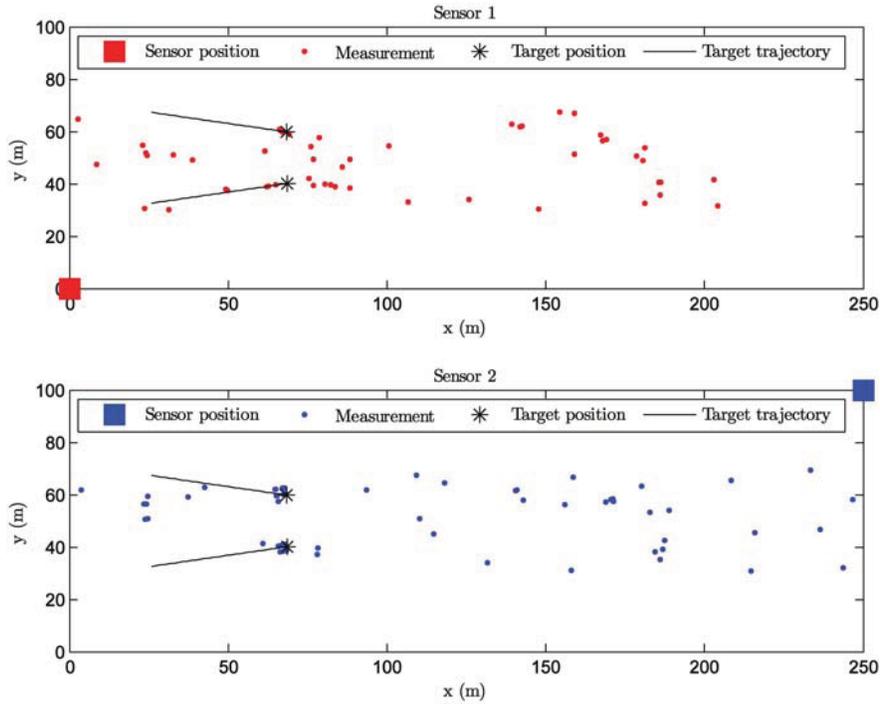


Fig. 6. Snapshot of all detections/measurements at Sensor 1 and Sensor 2 during a run in the crossing scenario.

of the targets are more spread out sideways in Sensor 2 than in Sensor 1. At this time the targets are closer to Sensor 1 than Sensor 2, and they are therefore better resolved by Sensor 1. The targets are also moving towards Sensor 2, and because of the scattering of the signal to the neighboring beams, the detections will be spread out more sideways from the direction of motion. Also notice how in some places the detections are located in groups due to the non-uniform spatial distribution of the clutter measurements. It is also possible that a target can be undetected, which is the case for the lower target at Sensor 1 in Fig. 6.

5.4. Track Formation and Termination

As can be seen in Fig. 6, the targets are often determined by a cluster of detections rather than a single point detection. In the simulations the tracks are initialized by two-point differencing [4, p. 247] of the cluster centroids from succeeding scans. The reason for this is to avoid confusion due to the many possibilities of two-point differencing that could have been set up among the point detections from one single target. The clustering method of the single point detections is described in [21], and is based on mathematical morphology [23]. Any successive pair of clusters within a maximum distance based on target maximum motion parameters and cluster measurement noise variances initiates a preliminary track. For the motion parameters, a maximum distance $d_{\max} = 1$ m together with the process noise matrix Q in (2) is used. The measurement noise for the clusters is computed from the different cells included in the cluster as a Gaussian mixture [4, pp. 55–56]. A preliminary

track has to receive measurements for a minimum of 4 time steps during the first 6 scans to become a confirmed track. This is also referred to as a “4/6” logic-based track formation procedure. Note that the clustering method is only used for the two-point differencing in the track initialization.

In the centralized tracking the multisensor filtering is described in Section 4, first updating with measurements from Sensor 1 and then with measurements from Sensor 2 in a sequential updating scheme. The track initialization in the centralized tracking algorithm is based on measurements that also contain velocity information. First, the two-point differencing is used at Sensor 1 to make an initial state. Then, the two-point differencing is used at Sensor 2, but these initial states are now used as measurements (including both position and velocity) to update the initial state from Sensor 1. The updating is done as in a regular PDAF, but since these measurements are formed by two-point differencing of cluster centroids from succeeding scans, they will not have the same measurement noise, yielding a varying innovation covariance (S_k in (6)). The innovation covariance is normally used to form the measurement validation gate in the PDAF, but in this case a fixed matrix

$$S_{\text{fix}} = \begin{bmatrix} \sigma_{\text{pos}}^2 & 0 & 0 & 0 \\ 0 & \sigma_{\text{pos}}^2 & 0 & 0 \\ 0 & 0 & \sigma_{\text{vel}}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\text{vel}}^2 \end{bmatrix} \quad (53)$$

is used instead of the non-constant innovation covariance to form a constant measurement validation gate.

The values used in S_{fix} is set based on the assumption that the standard deviation for these measurements are about 1 m for the position elements, and 0.5 m for the velocity elements ($\sigma_{\text{pos}} = 1$ m and $\sigma_{\text{vel}} = 0.5$ m).

In the modified filters, the wake assumption also affects the track initialization in the way that measurements inside the wake area W (defined in Section 5.2) are excluded in the initialization procedure.

To terminate a track one of the following events (termination events) must occur:

- 1) The estimated speed is outside the interval $[v_{\text{min}}, v_{\text{max}}]$, where $v_{\text{min}} = 0.1$ m/s and $v_{\text{max}} = 3$ m/s.
- 2) The estimate moves more than 5 m between two scans.
- 3) The position state estimation variance exceeds σ_{posmax}^2 , where $\sigma_{\text{posmax}}^2 = 50$ m².
- 4) There are no validated measurements received in a track within 5 successive scans.
- 5) The track is closer than d_{min} to another older track during 10 succeeding scans, where $d_{\text{min}} = 0.5$ m.

These track termination criteria are adopted rather than using more rigorous methods, such as the joint version of the IPDAF [17], because of their sensitivity to inaccurate estimates of the clutter density. In real sensor measurements, the signal is often scattered resulting in more than one target-originated detection. This will increase the clutter density resulting in an unrealistic low probability for the track to survive. This may be solved by the use of clustering, but for targets in the presence of wakes it is undesirable to blend the wake-originated detections together with the target-originated ones. The above termination criteria are more strict than those used in [21] due to the higher sensor resolution used in this paper.

5.5. Performance Analysis

The performance evaluation of a multitarget tracking system is always a difficult problem, and the quality of the results is difficult to quantify in terms of a few variables. When the evaluation is based on real data, where not all parameters are known, this problem becomes even harder. The results also depend on the simulation scenarios, and the performance of the JPDAF may, according to [13], show large local maxima and minima as a function of scenario parameters. However, by considering the basic scenarios described above and using a relatively large set of measures of performance (MOP), a certain amount of meaningful information should be obtained. The MOP considered are the following:

- 1) The percentage of lost tracks among all true tracks.
- 2) The percentage of swapped tracks among all true tracks (measured only when the targets are closer than 10 m).
- 3) The average fraction of each trajectory's total duration where the target is tracked (by a true track).

4) The average life length of a true track relative to its true target's life length.

5) The average time for target acquisition.

6) The number of false tracks per scan.

7) The average life length of a false track.

8) The position RMS error.

This section describes how these MOP are obtained before the corresponding results, based on 500 Monte Carlo runs for each of the four given scenarios, are shown. At a given time k there might exist several tracks, but for each target, at most one of them can be defined as true. The rest of the tracks are therefore by definition false. A track is first defined as true if the position estimation error is less than 1 m during the next 5 scans, and at the same time there are no other true track associated to the target. If there is more than one track fulfilling these requirements at the same time, the track with lowest average position estimation error during these 5 scans is defined as the true one. The true track will stay as such until either the position estimation error exceeds 5 m, or the position estimation error associated to a neighboring target is less than 1 m during the next 5 scans. In both situations the track will be declared as lost, but in the latter case it will also be defined as a swapped track.

1) *The percentage of lost tracks among all true tracks:*

In Fig. 7 the percentage of lost tracks is shown. The standard filters (PDAF and JPDAF) have the highest track loss percentage, and the JPDAF shows no improvement compared to the PDAF. The modified single target tracking algorithm (PDAF) performs better than the standard filters, but the best performance is achieved with the modified JPDAF and JPDAF. The difference between the standard filters and the modified PDAF is largest in the meeting and crossing scenarios where the targets are close to each other during a short time. When the targets stay together for a longer period of time, the modified PDAF is not significantly better than the standard filters because it does not account for the neighboring target and its wake like the modified JPDAF and JPDAF do. Also notice that there is almost no difference between the decoupled and coupled modified JPDAF, which indicates that the correlation between the targets' estimation errors is insignificant.

In the different scenarios considered the best performance is achieved for the meeting scenario. This is maybe a bit surprising since the density of the joint wake model after the passing is lowest between the targets, the area opposite to their moving direction. However, the high wake density in the whole joint validation region will at the same time give more confidence in the predicted target motion than the measurements. Because of this, and the fact that the velocities of the two targets are totally opposite to each other, the tracks will be less affected by the false measurements. The percentage of the lost tracks in the crossing scenario is the next best, and the good performance in both the meeting and crossing

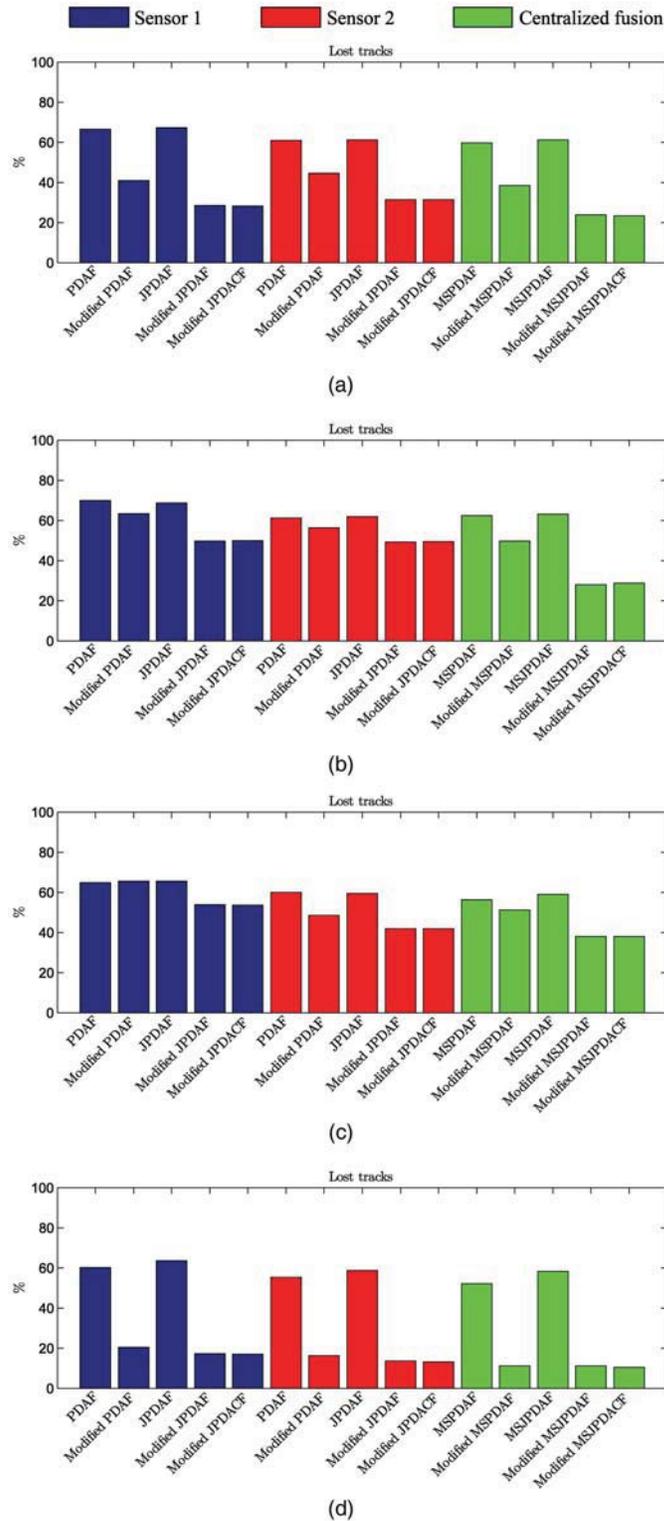


Fig. 7. Average percentage of lost tracks in the four simulation scenarios. (a) Scenario 1: Crossing trajectories. (b) Scenario 2: Parallel trajectories. (c) Scenario 3: Sequential trajectories. (d) Scenario 4: Meeting trajectories.

scenarios is as expected since the targets are only close to each other a short time. In these scenarios the results from the single sensor filters are almost as good as from the multisensor filters in the centralized tracking. This is not true for the parallel scenario where the performance is significantly improved by fusing the sensors'

data in the modified MSJPDAF and MSJPDA CF. In this scenario the targets are separated by only 3 m, which is close to the limit for having multiple targets in a single resolution cell (unresolved targets). By using two sensors in this situation, Sensor 1 resolves the targets relatively good in the beginning of the run, and Sen-

sensor 2 does the same at the end of the run. Because of this, the fusion of these two sensors data improves the performance significantly.

The most difficult scenario is the sequential, where a target is moving behind another target, surrounded by the wake. In this case the centralized tracking performs best, and a track loss under 40% is achieved by the modified MSJPDAF and MSJPDACF. In practice this means that, even in a hard case like this, at least one track will be kept throughout the run.

2) *The percentage of swapped tracks among all true tracks (measured only when the targets are closer than 10 m):* The percentage of the swapped tracks, shown in Fig. 8, is only measured when the distance between the targets is less than 10 m. The reason for this is to find the percentage of swapping among only the tracks where the two associated targets are close to each other. The swapping is, as expected, highest in the parallel scenario where the tracks are moving in parallel for a longer period of time. In this situation it is easy for a track to switch over to the neighboring target only 3 m away. In the meeting scenario the swapping phenomenon is totally absent for the modified filters, and practically absent for the standard filters (PDAF and JPDAF). The reason for this is the same as discussed under the previous MOP.

The modified PDAF has the most problems, especially in the parallel scenario, since it accounts for its own wake, but does not take into consideration that there is another target in the surrounding area. The standard filters, which do not consider the wakes, are more disposed to turn into their own wake than to swap to the neighboring target. Therefore, even if their track loss is higher, they have a lower swapping percentage than the modified PDAF.

The best performance is achieved by the modified MSJPDAF and MSJPDACF in the centralized tracking. This improvement is most significant in the parallel scenario, where the percentage of swapped tracks are almost halved for the modified MSJPDAF and MSJPDACF compared to the other filters.

3) *The average fraction of each trajectory's total duration where the target is tracked (by a true track):* In Fig. 9 the average percentage of the tracked part of the trajectories' duration is shown. Also here the modified JPDAF and JPDACF perform best, and by using the modified MSJPDAF or MSJPDACF in the centralized tracking, about 90% of the trajectories are tracked. Notice that the percentage of the tracked trajectory can be very good even with a high track loss percentage if tracks are quickly reacquired after a loss. It is therefore important to consider other MOP to get the total picture.

4) *The average life length of a true track relative to its true target's life length:* In Fig. 10 the average life length (in %) of the true tracks is shown. It is clear that the track length is significantly increased by the modified filters, and most by the modified multitarget tracking filters (JPDAF and JPDACF). The best perfor-

mance is achieved by the modified filters in the meeting scenario, where the average track length is about 80% of the true target's life length, more than twice as long as for the standard PDAF and JPDAF. The improvement by using multiple sensors is most significant for the modified MSJPDAF and MSJPDACF in the parallel scenario. In this situation the combination of both using the multitarget wake model, and for the targets to be well resolved by at least one sensor all the time throughout the run, is vital. In the sequential scenario the best track length is almost 60% for the same modified multisensor filters. This is due to the fact that when a track is first lost inside the wake of another target in front, it is very hard to reacquire a track on the rear target.

5) *The average time for target acquisition:* In many situations it is important to quickly initiate tracks and reacquire them once lost. Let the time for target acquisition be the time before a track is defined as true either in the beginning of a run or after a track was lost. The average time for target acquisition (or reacquisition) is shown in Fig. 11. For the crossing, the parallel and the meeting scenarios, the modified filters perform slightly better than the standard filters. At the sequential scenario the behavior is different in the way that the standard filters outperform the modified filters. This is due to the assumption that the measurements behind a target originate from a wake and not a target. Therefore, when the target following the target in front is lost, the real target-originated measurements will not be considered for a new track as long as they are inside the wake area W of the target in front.

In the first three scenarios it is harder to initiate/reacquire true tracks at Sensor 2 than for Sensor 1. The reason for this is that tracks are starting close to Sensor 1, and far from Sensor 2, and the detections of the targets (and wakes) will therefore be more spread out in the view of Sensor 2. This can be seen in Fig. 6, and makes it harder to acquire tracks in the two-point differencing of the cluster centroids from succeeding scans.

6) *The average life length of a false track:* As mentioned above, all tracks that are not defined as true, are considered false. The average life length of a false track is shown in Fig. 12, and the performance is almost the same for all filters, with an insignificant tendency for shorter life length of the false tracks in the modified filters.

7) *The number of false tracks per scan:* Another MOP considering the false tracks, is the average number of false tracks per scan, shown in Fig. 13. This number is higher for the standard filters than for the modified ones because the standard filters do not restrict the track formation inside the wake areas behind the targets. Also, there are more false tracks for the centralized tracking due to the fact that this tracking algorithm takes into account false measurements from both sensors.

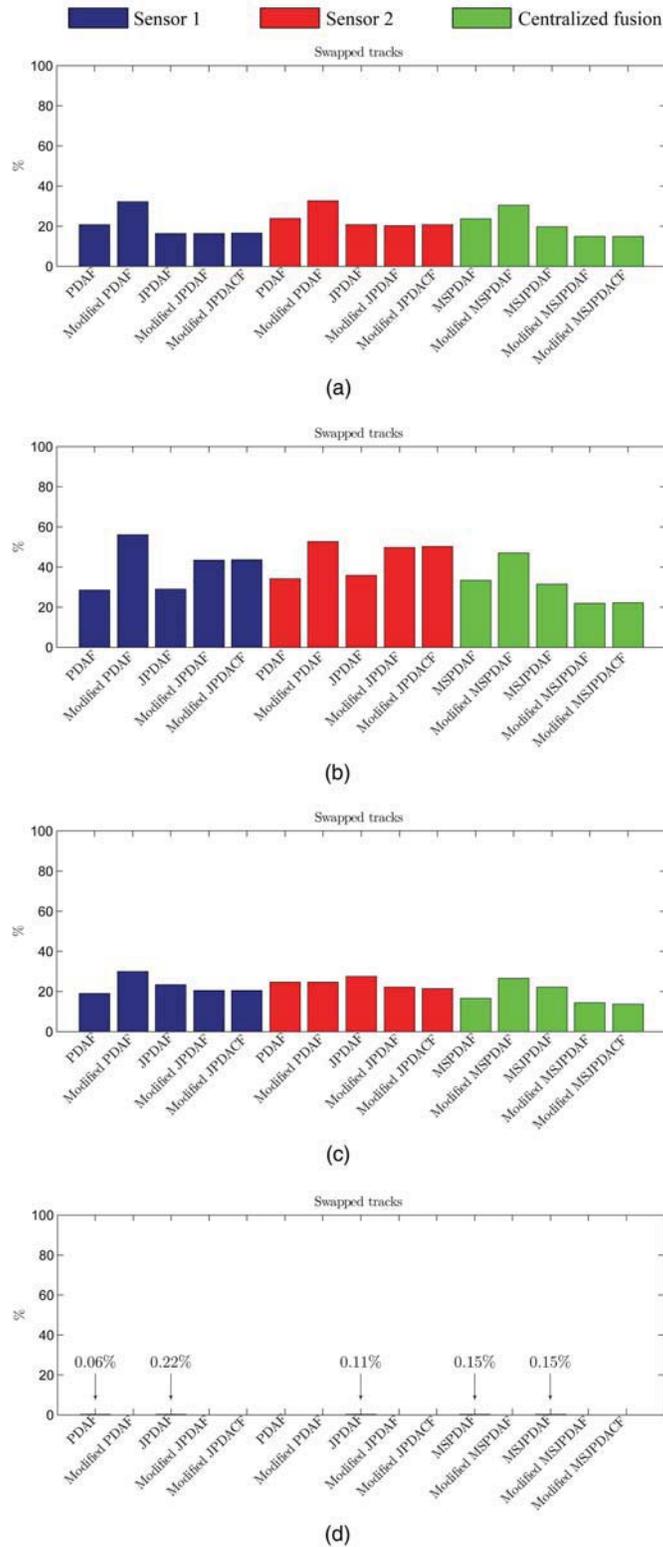
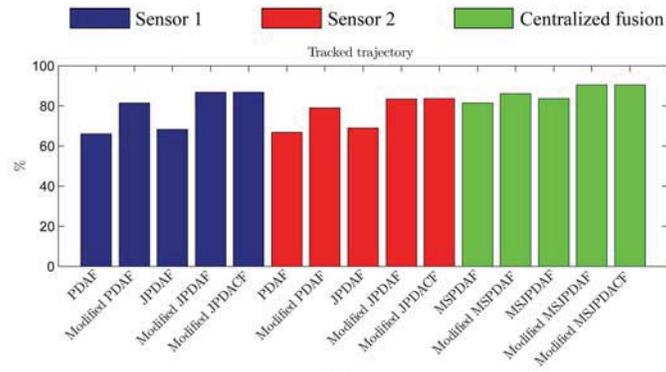


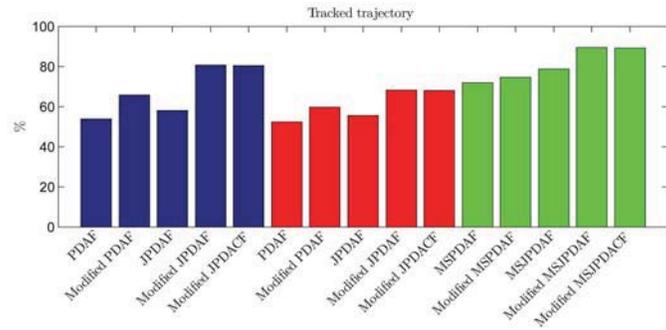
Fig. 8. Average percentage of swapped tracks in the three simulation scenarios. (a) Scenario 1: Crossing trajectories. (b) Scenario 2: Parallel trajectories. (c) Scenario 3: Sequential trajectories. (d) Scenario 4: Meeting trajectories.

8) *The position RMS error:* The last MOP in this analysis is the position RMS error, given in Fig. 14. The RMS error is based only on the true tracks in the simulation scenarios. In all scenarios the position RMS error is larger for the standard filters than for the

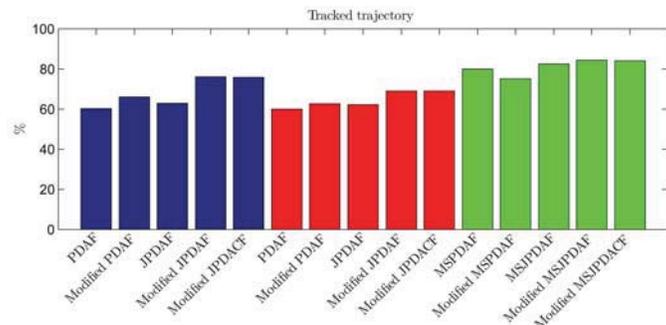
modified filters. This is because the standard filters do not consider the wake-originated measurements like the modified filters do, and the state estimate is therefore likely to be drawn into the wake. It can also be seen that the RMS error, at least for the modified JPDAF



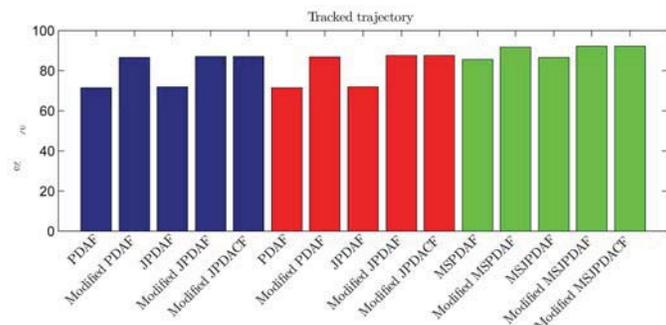
(a)



(b)



(c)



(d)

Fig. 9. Average percentage of tracked trajectory in the four simulation scenarios. (a) Scenario 1: Crossing trajectories. (b) Scenario 2: Parallel trajectories. (c) Scenario 3: Sequential trajectories. (d) Scenario 4: Meeting trajectories.

and JPDACF, is slightly reduced in the centralized tracking.

In the two first scenarios (crossing and parallel), the error increases during the periods when the targets are close to each other. For the crossing scenario this

is seen as a “jump” in the error when the targets are crossing between 80 s and 120 s. In the parallel scenario, this jump starts at about 60 s and ends at 140 s, which are the period the targets are moving in parallel. In these situations the estimate for one target

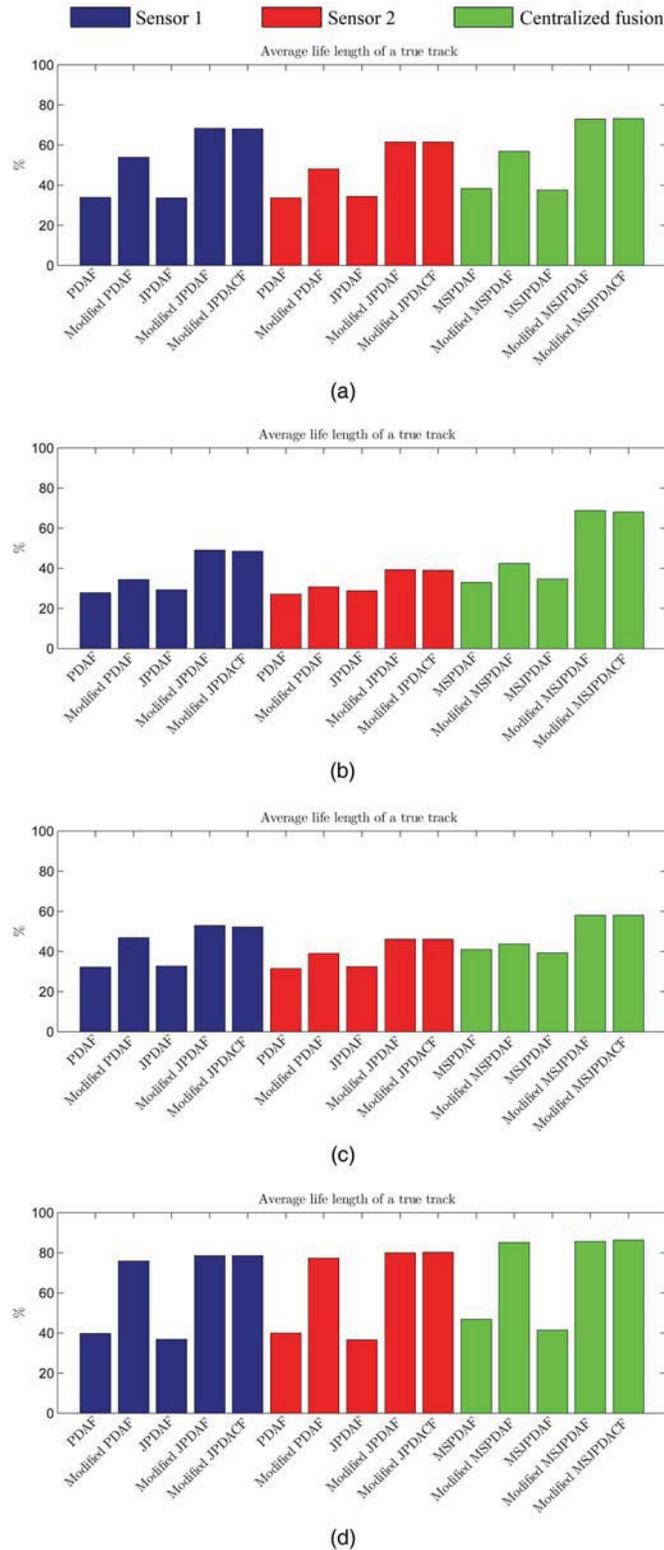


Fig. 10. Average life length of a true track relative to its true target's life length in the four simulation scenarios. (a) Scenario 1: Crossing trajectories. (b) Scenario 2: Parallel trajectories. (c) Scenario 3: Sequential trajectories. (d) Scenario 4: Meeting trajectories.

will be drawn towards the other target, also known as track coalescence [8]. Among the modified filters, this is most problematic for the single target tracking algorithm because it accounts for the wake behind its own target, but has no information about the nearby

target which also has a wake behind it. The modified multitarget filters perform similarly, and their RMS errors are almost constant throughout the run.

In the meeting scenario only a small tendency of the jump phenomenon is noticeable shortly after the pass-

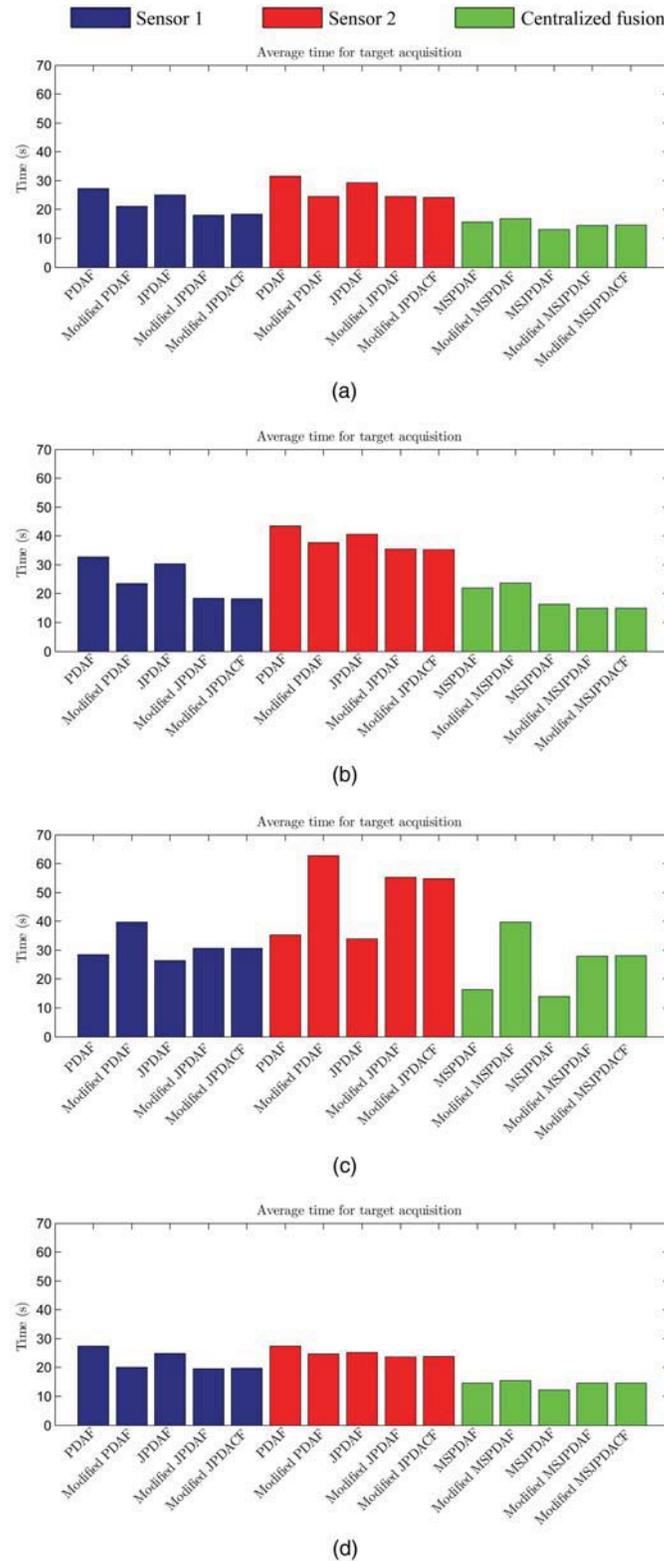
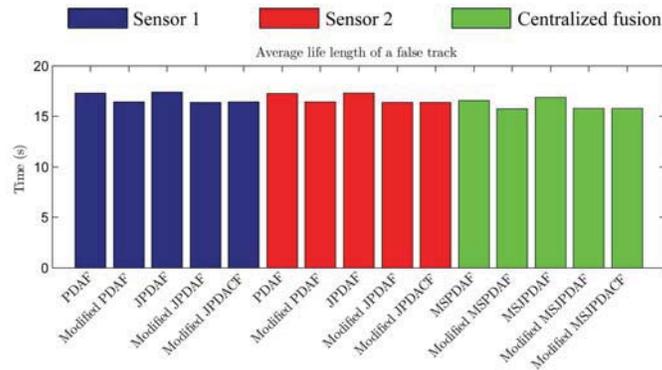


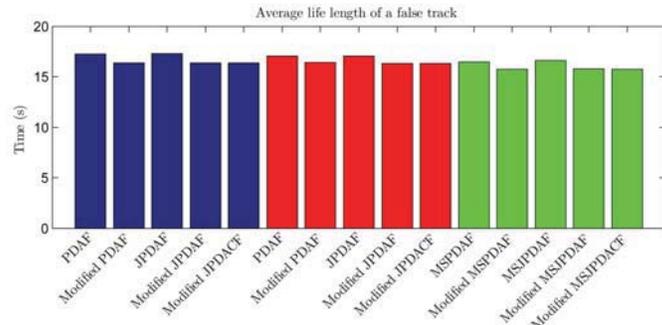
Fig. 11. Average time for target acquisition in the four simulation scenarios. (a) Scenario 1: Crossing trajectories. (b) Scenario 2: Parallel trajectories. (c) Scenario 3: Sequential trajectories. (d) Scenario 4: Meeting trajectories.

ing. As discussed above, the totally opposite velocities of the two targets and the high wake density in the whole joint validation region, make the targets' passing relatively easy.

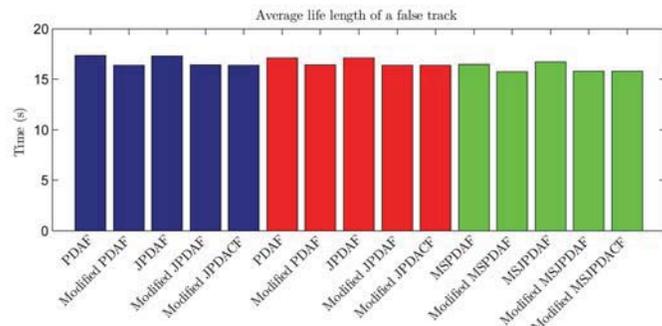
In the sequential scenario the targets are never closer than about 5 m, so the RMS error does not increase much during the period the tracks are following after each other. In this scenario, the modified single target



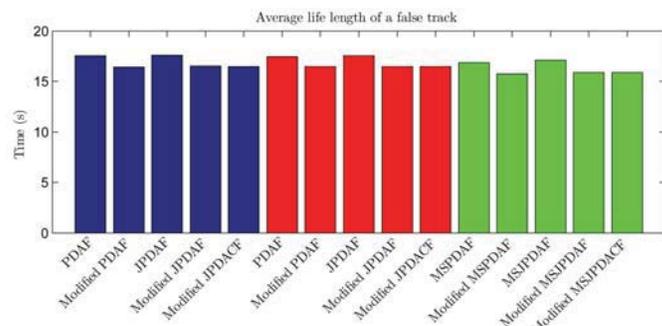
(a)



(b)



(c)

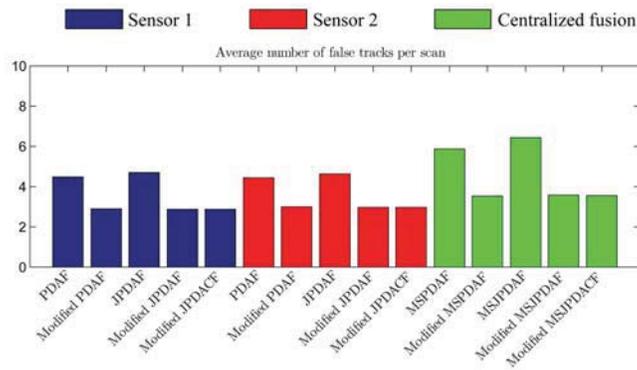


(d)

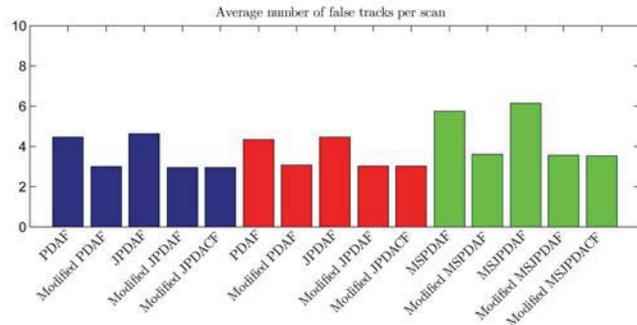
Fig. 12. Average track length for a false track in the four simulation scenarios. (a) Scenario 1: Crossing trajectories. (b) Scenario 2: Parallel trajectories. (c) Scenario 3: Sequential trajectories. (d) Scenario 4: Meeting trajectories.

tracking filter performs better than the modified multi-target tracking filters. The reason for this is because the RMS error is measured only among the true tracks, not when they become lost. In this scenario, where the targets are following after each other, the estimation error

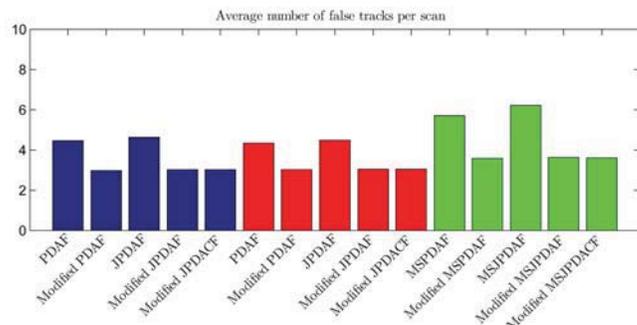
is larger for the target behind the one in front, because it is surrounded by wakes. From the percentage of lost tracks in Fig. 7, the modified single target filter will lose the target more often than the modified multitarget filters, and it is most likely that the lost target is the



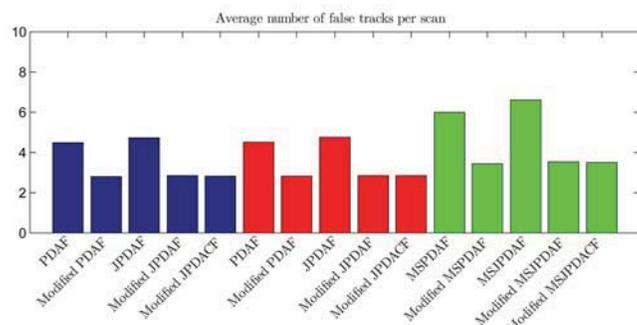
(a)



(b)



(c)



(d)

Fig. 13. Average number of false tracks per scan in the four simulation scenarios. (a) Scenario 1: Crossing trajectories. (b) Scenario 2: Parallel trajectories. (c) Scenario 3: Sequential trajectories. (d) Scenario 4: Meeting trajectories.

one with largest estimation error. Therefore, when the RMS error is calculated, the modified multitarget filters are based on tracks with larger estimation error than what the modified single target filter is based on, only because these tracks were not lost.

5.6 Usage of the Wake Model on Targets Without Wakes

In this section the erroneous use of wake models on targets without wakes is considered. The crossing sce-

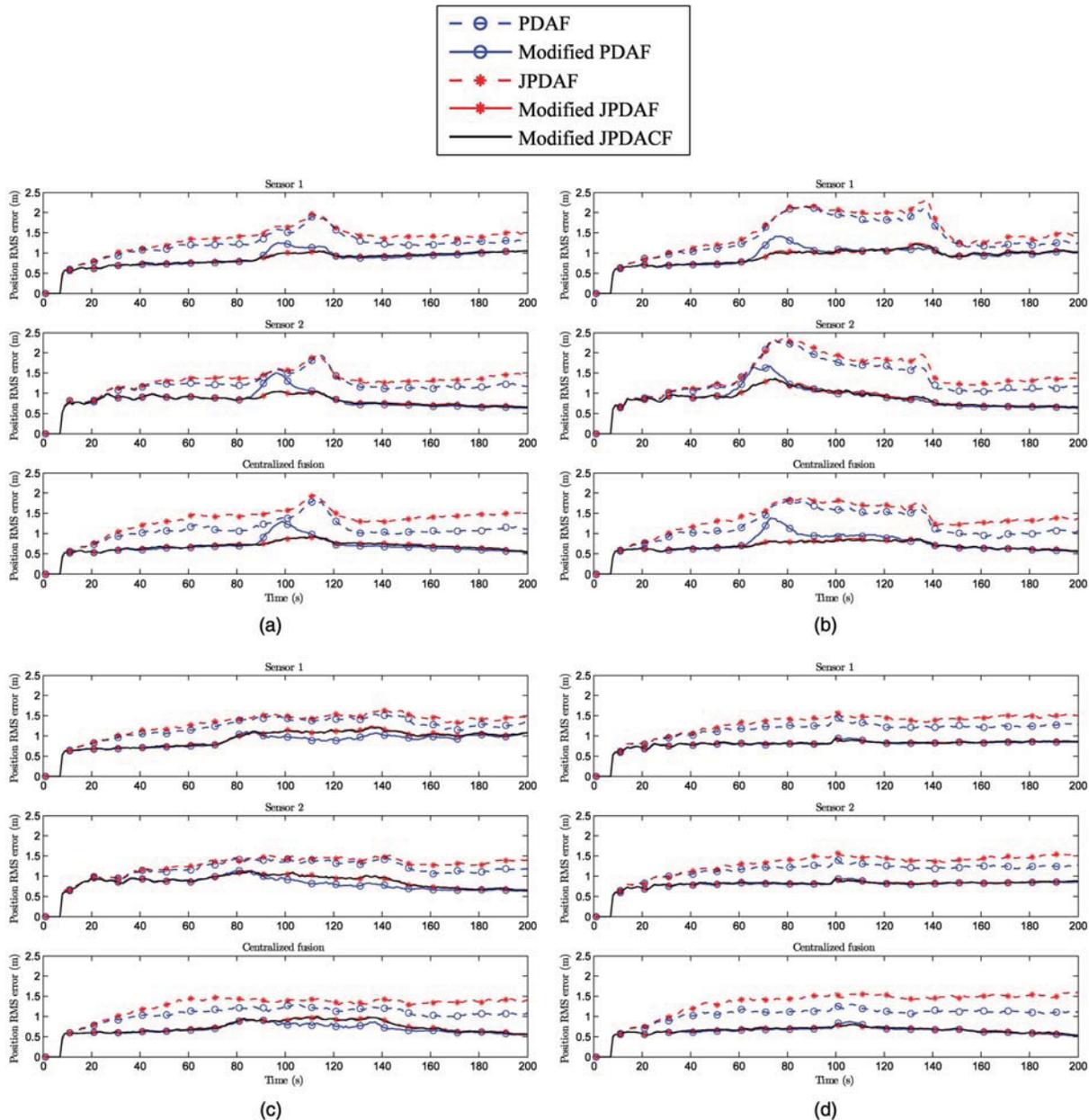


Fig. 14. Position RMS error from 500 Monte Carlo runs in: (a) Scenario 1: Crossing trajectories, (b) Scenario 2: Parallel trajectories, (c) Scenario 3: Sequential trajectories, (d) Scenario 4: Meeting trajectories.

nario (see Fig. 4) is used as before, but without wakes behind the targets.² Each target is simulated as a point-target (only one measurement) with detection probability $P_D = 0.7$, independently across time. The results after 500 Monte Carlo runs are shown in Fig. 15, and the performance is clearly better than in the wake-scenario due to the fact that each target is never simulated by more than one detection at a time. This shows that the scattering effect in real sensors, due to the beamforming, makes the tracking problem considerably harder and is an important element in further research.

²This would correspond to “closed breathing system” scuba divers or mechanical underwater vehicles.

It is also interesting to see that even though the modified PDAF performs worse than the standard PDAF, the modified multitarget algorithms perform almost the same as the standard JPDAF. This indicates that applying the modified JPDAF or JPDAF on targets without wakes will not degrade the tracking performance. For tracking in environments with different kinds of targets, with and without wakes, this is a desirable property.

Another issue worth mentioning is the increasing trend of the position RMS error for Sensor 1, and the decreasing trend for Sensor 2. This is due to the fact that the estimated position error increases with the distance to the sensors, and the targets are starting close to Sensor 1, and moving towards Sensor 2. Notice how

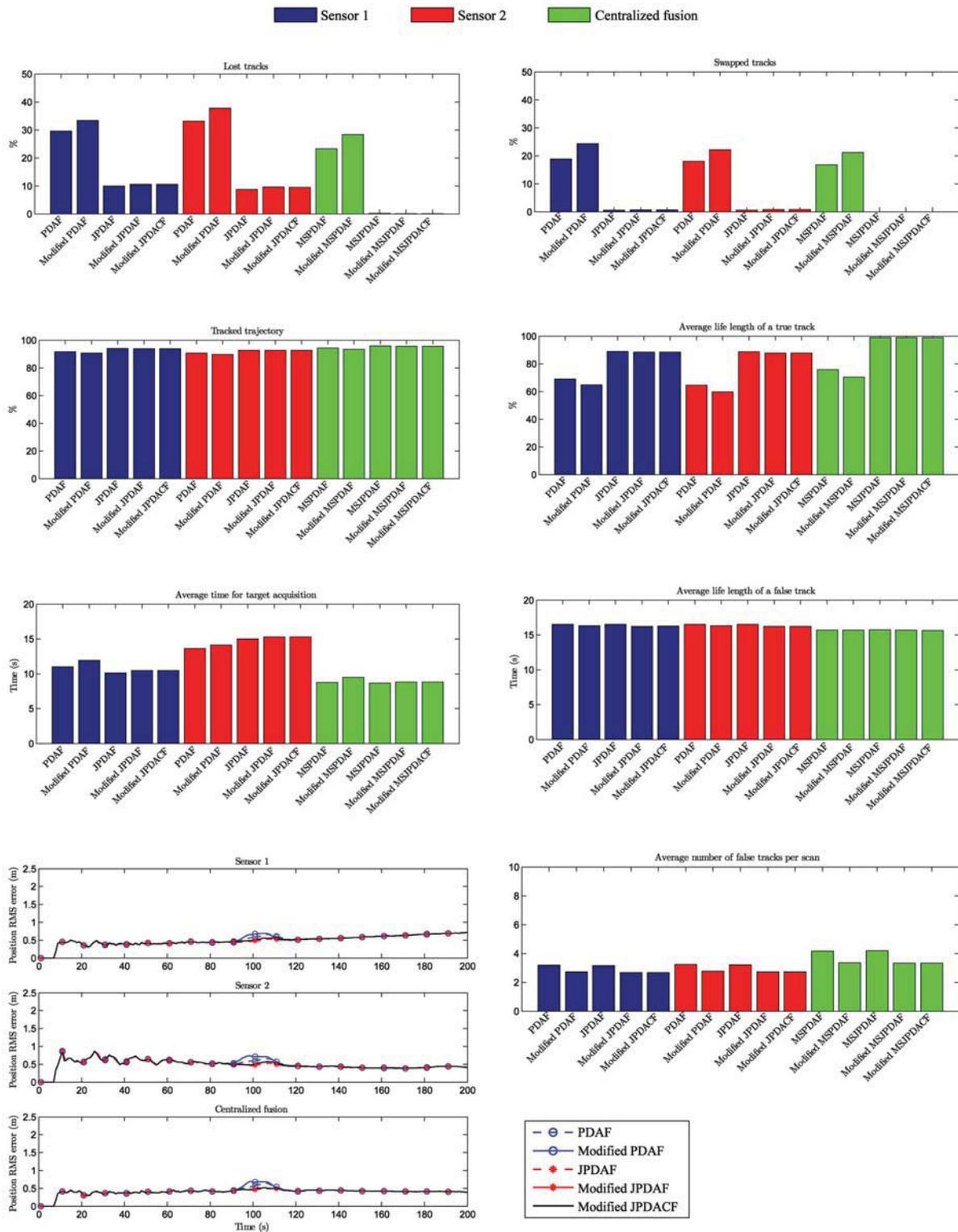


Fig. 15. Simulation results from the crossing scenario where the probabilistic wake model is erroneously applied to targets without wakes. Each target is simulated as point detections with detection probability $P_D = 0.7$, independent across time. The different features discussed in Section 5.5 are shown.

these trends in the single sensors are averaged out in the centralized tracking where the sensors' data are fused.

It is also shown that the use of multiple sensors is more effective in preventing lost tracks in this special

case than in the previous cases where the targets had wakes. This gives another justification of using multiple sensors when tracking targets in the presence of wakes, because a target could be mistaken for having a wake even when it does not have one.

6. CONCLUSIONS

An important factor in a multitarget tracking system is to correctly associate each measurement received from a detector to its origin. The JPDAF has been a solution to this problem due to its effectiveness and low computational demand. In the JPDAF all false measurements are assumed due to i.i.d. uniformly spatially distributed noise or clutter. This assumption is not adequate for targets that generate wakes, because detections originating from the wake are not uniformly distributed and may result in a lost track if they are not properly modelled. The solution presented incorporates a model of the wakes behind the targets in a multitarget environment. The purpose of this wake model is to weight wake-originated measurements lower than in a regular JPDAF to avoid the tracks following these measurements and therefore be forced to turn into the wake. To achieve this, we presented a model formed by the sum of single models each linearly increasing behind their associated targets.

A systematic comparison of the standard data association filters (PDAF and JPDAF) and their corresponding modified versions are presented in a multitarget multisensor environment. Four different simulation scenarios are examined where two targets in the presence of wakes are crossing, moving in parallel to each other, one following after another, and finally meeting and then passing each other. It is shown that the wake model presented is a useful modification of the JPDAF in all four scenarios. The only stated drawback using the wake model is when a target is moving after another one, surrounded by the wake from the target in front. In that case, if the rear target is lost, it is harder to reacquire the track because the measurements are assumed originating from the wake and not the true target.

This paper also presents the coupled version of the JPDAF, called JPDACF, and a modified JPDACF (with a wake model) is developed and tested. The simulations show that the modified JPDACF is not improving the performance compared to the simpler modified JPDAF, indicating that there is no significant correlation between the targets' estimation errors.

The simulation scenarios consider two sensors, and the data association filters at the local sensors are compared with multisensor (MS) filters in a centralized tracking configuration. A sequential state updating scheme is used in the multisensor filters, and the results show that the data fusion provides significant improvement in the tracking performance.

This paper also examines the effect of applying the wake model on point-targets without wakes. The results show that the modified JPDAF and JPDACF perform almost the same as the standard JPDAF. This makes the modification practical for real systems where both targets with wakes and targets without wakes are operating in the same environment.

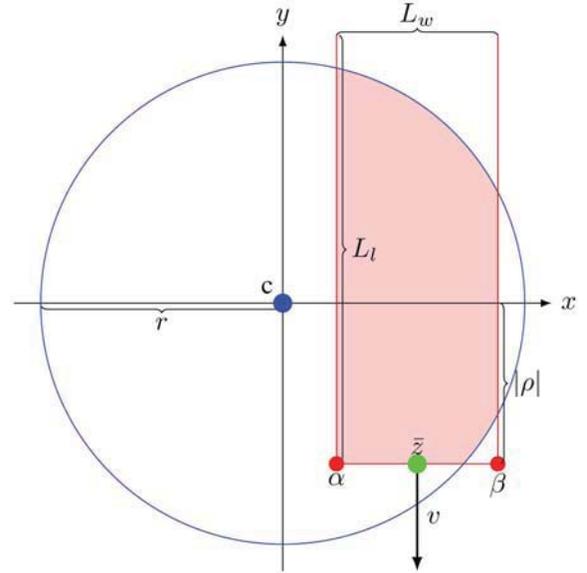


Fig. 16. Specification of variables for integration of the wake model, with length L_l and width L_w , inside the joint validation region with center c and radius r . The wake has front corners $[\alpha \ \rho]$ and $[\beta \ \rho]$ and is oriented behind the target with position \bar{z} and velocity v .

APPENDIX A. SPECIFICATION OF THE JOINT WAKE MODEL

In this appendix the joint wake model $p_W(z_k)$ introduced in Section 3 is presented, and an analytical expression for the probability P_{GW} is derived. The joint wake model is the sum of all N_T single wake models $p_W^t(z_k)$ behind each target t under consideration

$$p_W(z_k) = \frac{1}{N_T} \sum_{t=1}^{N_T} p_W^t(z_k). \quad (54)$$

Next, consider the single wake model $p_W^t(z_k)$ of target t , and let \bar{z} and v be the predicted position and velocity of the target, respectively. Reference to Fig. 16 may be helpful in the following. The single wake model is assumed linearly increasing with length L_l behind the predicted position of the target, i.e., in the direction opposite to v , and uniform with width L_w in the direction perpendicular to the target's velocity v . This model can be expressed by defining the independent variables l and w as the respective distances behind and sideways (relative to v) to the target. From the above assumption, l and w have the following densities

$$\begin{aligned} p_l(l) &= \frac{2l}{L_l^2} & 0 \leq l \leq L_l \\ p_w(w) &= \frac{1}{L_w} & 0 \leq w \leq \frac{L_w}{2} \end{aligned} \quad (55)$$

which yields

$$p_W^t(z_k) = p_l(l)p_w(w) = \frac{2l}{L_l^2 L_w}. \quad (56)$$

Notice that even though a current estimate of the velocity v is available in the filter, a better way in practice is to use an average of the latest estimates since the wake will not change direction as rapidly as the current target velocity estimate. In the simulations an average of the latest 6 estimates is used.

The joint validation region containing all candidate measurements in the multitarget environment is defined as a circle with radius r and center c . The center c is calculated as the average between all the predicted target positions, and the radius r is defined as the distance to the farthest validated measurement.

The probability P_{GW} in (25), used to restrict the density of the joint wake model $p_W(z_k)$ to the joint validation region, has to be calculated for each scan by integration of $p_W(z_k)$ inside the region. Since $p_W(z_k)$ is the sum of all single wake models $p_W^t(z_k)$, P_{GW} is obtained by calculating P_{GW}^t for each target t and then summing them up

$$P_{GW} = \sum_{t=1}^{N_T} P_{GW}^t. \quad (57)$$

The calculation of P_{GW}^t is derived next. Assume a Cartesian coordinate system with origin at position c and y -axis parallel to v but in the opposite direction, see Fig. 16. Define the two front corners of the wake model with elements α and β for the x -axis, and ρ for the y -axis

$$\begin{aligned} \rho &= (c - \bar{z})^T v / |v| \\ \alpha &= \sqrt{|c - \bar{z}|^2 - \rho^2} - w/2 \\ \beta &= \sqrt{|c - \bar{z}|^2 - \rho^2} + w/2. \end{aligned} \quad (58)$$

The integration depends on if the front corners $[\alpha \ \rho]^T$ and $[\beta \ \rho]^T$ are inside or outside the joint validation region (circle), and will be broken into one, two or three parts. To do this, define three binary variables δ_ρ , δ_α and δ_β as follows:

$$\delta_\rho = \begin{cases} 1 & \text{if } \rho < 0 \\ 0 & \text{otherwise} \end{cases} \quad (59)$$

$$\delta_\alpha = \begin{cases} 1 & \text{if } \sqrt{\alpha^2 + \rho^2} > r \\ 0 & \text{otherwise} \end{cases} \quad (60)$$

$$\delta_\beta = \begin{cases} 1 & \text{if } \sqrt{\beta^2 + \rho^2} > r \\ 0 & \text{otherwise} \end{cases}. \quad (61)$$

Then the integral can be written as

$$\begin{aligned} P_{GW}^t &= \frac{2}{L_t^2 L_w} \left\{ \delta_\rho \delta_\alpha \int_{\max(\alpha, -r)}^{-\sqrt{r^2 - \rho^2}} \int_{-\sqrt{r^2 - x^2}}^{\sqrt{r^2 - x^2}} (y - \rho) dy dx \right. \\ &\quad + \int_{\alpha(1 - \delta_\alpha) - \delta_\alpha \sqrt{r^2 - \rho^2}}^{\beta(1 - \delta_\beta) + \delta_\beta \sqrt{r^2 - \rho^2}} \int_{\rho}^{\sqrt{r^2 - x^2}} (y - \rho) dy dx \\ &\quad \left. + \delta_\rho \delta_\beta \int_{\sqrt{r^2 - \rho^2}}^{\min(\beta, r)} \int_{-\sqrt{r^2 - x^2}}^{\sqrt{r^2 - x^2}} (y - \rho) dy dx \right\}. \end{aligned} \quad (62)$$

For simplicity we substitute the limits of integration along the x -axis as follows:

$$\begin{aligned} a &= \max(\alpha, -r) \\ b &= -\sqrt{r^2 - \rho^2} \\ c &= \alpha(1 - \delta_\alpha) - \delta_\alpha \sqrt{r^2 - \rho^2} \\ d &= \beta(1 - \delta_\beta) + \delta_\beta \sqrt{r^2 - \rho^2} \\ e &= \sqrt{r^2 - \rho^2} \\ f &= \min(\beta, r) \end{aligned} \quad (63)$$

which yields

$$\begin{aligned} P_{GW}^t &= \frac{2}{L_t^2 L_w} \left\{ \delta_\rho \delta_\alpha \int_a^b \int_{-\sqrt{r^2 - x^2}}^{\sqrt{r^2 - x^2}} (y - \rho) dy dx \right. \\ &\quad + \int_c^d \int_{\rho}^{\sqrt{r^2 - x^2}} (y - \rho) dy dx \\ &\quad \left. + \delta_\rho \delta_\beta \int_e^f \int_{-\sqrt{r^2 - x^2}}^{\sqrt{r^2 - x^2}} (y - \rho) dy dx \right\} \\ &= \frac{1}{L_t^2 L_w} \left\{ \rho r^2 \left(\arcsin \frac{c}{r} - \arcsin \frac{d}{r} \right) \right. \\ &\quad + \frac{c^3 - d^3}{3} + 2\rho \delta_\alpha \delta_\rho \left(a\sqrt{r^2 - a^2} - b\sqrt{r^2 - b^2} \right) \\ &\quad - c(\rho^2 + r^2) + 2\rho \delta_\beta \delta_\rho \left(e\sqrt{r^2 - e^2} - f\sqrt{r^2 - f^2} \right) \\ &\quad + d(\rho^2 + r^2) + 2\rho \delta_\alpha \delta_\rho r^2 \left(\arcsin \frac{a}{r} - \arcsin \frac{b}{r} \right) \\ &\quad + \rho c \sqrt{r^2 - c^2} + 2\rho \delta_\beta \delta_\rho r^2 \left(\arcsin \frac{e}{r} - \arcsin \frac{f}{r} \right) \\ &\quad \left. - \rho d \sqrt{r^2 - d^2} \right\}. \end{aligned} \quad (64)$$

APPENDIX B. COVARIANCE UPDATE IN THE MODIFIED JPDACF

In this section the updated stacked covariance for the JPDACF in (47) is derived. The updated covariance $P_{k|k}^S$, conditioned on all measurements up to time k , Z^k , is

$$P_{k|k}^S = E\{[x_k^S - \hat{x}_{k|k}^S][x_k^S - \hat{x}_{k|k}^S]^T | Z^k\}. \quad (65)$$

This can be expressed as a weighted sum of all joint association event conditioned estimation error covariances by using the total probability theorem

$$\begin{aligned} P_{k|k}^S &= \sum_{\Theta_k} P\{\Theta_k | Z^k\} \\ &\quad \times E\{[x_k^S - \hat{x}_{k|k}^S][x_k^S - \hat{x}_{k|k}^S]^T | Z^k, \Theta_k\}. \end{aligned} \quad (66)$$

Let $\hat{x}_{k|k}^S(\Theta)$ be the state estimate conditioned on the joint association event Θ_k . By using this, (66) can be

rewritten as

$$\begin{aligned}
P_{k|k}^S &= \sum_{\Theta_k} P\{\Theta_k | Z^k\} \\
&\quad \times E \left\{ [(x_k^S - \hat{x}_{k|k}^S(\Theta) + \hat{x}_{k|k}^S(\Theta) - \hat{x}_{k|k}^S)] \right. \\
&\quad \left. \times [(x_k^S - \hat{x}_{k|k}^S(\Theta) + \hat{x}_{k|k}^S(\Theta) - \hat{x}_{k|k}^S)]^T | Z^k, \Theta_k \right\} \\
&= \sum_{\Theta_k} P\{\Theta_k | Z^k\} \\
&\quad \times E \left\{ (x_k^S - \hat{x}_{k|k}^S(\Theta))(x_k^S - \hat{x}_{k|k}^S(\Theta))^T \right. \\
&\quad + (x_k^S - \hat{x}_{k|k}^S(\Theta))(\hat{x}_{k|k}^S(\Theta) - \hat{x}_{k|k}^S)^T \\
&\quad + (\hat{x}_{k|k}^S(\Theta) - \hat{x}_{k|k}^S)(x_k^S - \hat{x}_{k|k}^S(\Theta))^T \\
&\quad \left. + (\hat{x}_{k|k}^S(\Theta) - \hat{x}_{k|k}^S)(\hat{x}_{k|k}^S(\Theta) - \hat{x}_{k|k}^S)^T | Z^k, \Theta_k \right\} \\
&= \sum_{\Theta_k} P\{\Theta_k | Z^k\} \\
&\quad \times E \{(x_k^S - \hat{x}_{k|k}^S(\Theta))(x_k^S - \hat{x}_{k|k}^S(\Theta))^T | Z^k, \Theta_k\} \\
&\quad + \sum_{\Theta_k} P\{\Theta_k | Z^k\} \\
&\quad \times (\hat{x}_{k|k}^S(\Theta) - \hat{x}_{k|k}^S)(\hat{x}_{k|k}^S(\Theta) - \hat{x}_{k|k}^S)^T \\
&= \sum_{\Theta_k} P\{\Theta_k | Z^k\} \\
&\quad \times \underbrace{E \{(x_k^S - \hat{x}_{k|k}^S(\Theta))(x_k^S - \hat{x}_{k|k}^S(\Theta))^T | Z^k, \Theta_k\}}_{P_\Theta} \\
&\quad + \underbrace{\sum_{\Theta_k} P\{\Theta_k | Z^k\} \hat{x}_{k|k}^S(\Theta) \hat{x}_{k|k}^S(\Theta)^T - \hat{x}_{k|k}^S(\hat{x}_{k|k}^S)^T}_{\tilde{P}} \\
&= \sum_{\Theta_k} P\{\Theta_k | Z^k\} P_\Theta + \tilde{P} \tag{67}
\end{aligned}$$

where the identity

$$\sum_{\Theta_k} P\{\Theta_k | Z^k\} = 1 \tag{68}$$

is used together with the fact that

$$\hat{x}_{k|k}^S(\Theta) = E\{x_k^S | Z^k, \Theta_k\} \tag{69}$$

$$\hat{x}_{k|k}^S = \sum_{\Theta_k} P\{\Theta_k | Z^k\} \hat{x}_{k|k}^S(\Theta). \tag{70}$$

Next, P_Θ in the first term of (67) will be derived

$$P_\Theta = E\{(x_k^S - \hat{x}_{k|k}^S(\Theta))(x_k^S - \hat{x}_{k|k}^S(\Theta))^T | Z^k, \Theta\} \tag{71}$$

where the conditioned estimation error is

$$\begin{aligned}
\tilde{x}_k^S(\Theta) &= x_k^S - \hat{x}_{k|k}^S(\Theta) \\
&= F^S x_{k-1}^S + v_{k-1}^S - F^S \hat{x}_{k-1|k-1}^S - I_\Theta^x W_k^S I_\Theta^z \nu_k^S(\Theta). \tag{72}
\end{aligned}$$

Substituting (under the assumption that all targets are observed)

$$\begin{aligned}
\nu_k^S(\Theta) &= z_k^S(\Theta) - H^S F^S \hat{x}_{k-1|k-1}^S \\
&= H^S F^S x_{k-1}^S + H^S v_{k-1}^S + w_k^S - H^S F^S \hat{x}_{k-1|k-1}^S \\
&= H^S F^S \tilde{x}_{k-1}^S + H^S v_{k-1}^S + w_k^S \tag{73}
\end{aligned}$$

in (72) yields

$$\begin{aligned}
\tilde{x}_k^S(\Theta) &= (I - I_\Theta^x W_k^S I_\Theta^z H^S) F^S \tilde{x}_{k-1}^S \\
&\quad + (I - I_\Theta^x W_k^S I_\Theta^z H^S) v_{k-1}^S - I_\Theta^x W_k^S I_\Theta^z w_k^S. \tag{74}
\end{aligned}$$

Using this in (71) gives

$$\begin{aligned}
P_\Theta &= (I - I_\Theta^x W_k^S I_\Theta^z H^S) P_{k|k-1}^S (I - I_\Theta^x W_k^S I_\Theta^z H^S)^T \\
&\quad + I_\Theta^x W_k^S I_\Theta^z R^S I_\Theta^z W_k^S I_\Theta^x \tag{75}
\end{aligned}$$

where

$$P_{k|k-1}^S = F^S P_{k-1|k-1}^S F^{ST} + Q^S. \tag{76}$$

In (75) the assumptions in (2) are used together with the following independence assumptions between the estimation error and the two noises

$$E\{\tilde{x}_k^S v_k^{ST} | Z^k, \Theta_k\} = 0 \tag{77}$$

$$E\{\tilde{x}_{k-1}^S w_k^{ST} | Z^k, \Theta_k\} = 0. \tag{78}$$

The last term \tilde{P} in (67) is

$$\begin{aligned}
\tilde{P} &= \sum_{\Theta_k} P\{\Theta_k | Z^k\} \hat{x}_{k|k}^S(\Theta) \hat{x}_{k|k}^S(\Theta)^T - \hat{x}_{k|k}^S(\hat{x}_{k|k}^S)^T \\
&= \sum_{\Theta_k} P\{\Theta_k | Z^k\} (\hat{x}_{k|k-1}^S + I_\Theta^x W_k^S I_\Theta^z \nu_k^S(\Theta)) \\
&\quad \times (\hat{x}_{k|k-1}^S + I_\Theta^x W_k^S I_\Theta^z \nu_k^S(\Theta))^T \\
&\quad - \left(\hat{x}_{k|k-1}^S + \sum_{\Theta_k} P\{\Theta_k | Z^k\} I_\Theta^x W_k^S I_\Theta^z \nu_k^S(\Theta) \right) \\
&\quad \times \left(\hat{x}_{k|k-1}^S + \sum_{\Theta_k} P\{\Theta_k | Z^k\} I_\Theta^x W_k^S I_\Theta^z \nu_k^S(\Theta) \right)^T \tag{79}
\end{aligned}$$

which after cancellations becomes the spread of innovations

$$\begin{aligned}
\tilde{P} &= \sum_{\Theta_k} P\{\Theta_k | Z^k\} I_\Theta^x W_k^S I_\Theta^z \nu_k^S(\Theta) \nu_k^S(\Theta)^T I_\Theta^z W_k^S I_\Theta^x \\
&\quad - \left(\sum_{\Theta_k} P\{\Theta_k | Z^k\} I_\Theta^x W_k^S I_\Theta^z \nu_k^S(\Theta) \right) \\
&\quad \times \left(\sum_{\Theta_k} P\{\Theta_k | Z^k\} I_\Theta^x W_k^S I_\Theta^z \nu_k^S(\Theta) \right)^T. \tag{80}
\end{aligned}$$

Using (80) and (75) in (67) yields the updated stacked covariance

$$\begin{aligned}
P_{k|k}^S &= \sum_{\Theta_k} P\{\Theta_k | Z^k\} \\
&\times \left\{ I_{\Theta}^x W_k^S I_{\Theta}^z (\nu_k^S(\Theta)) \nu_k^S(\Theta)^T + R^S \right\} I_{\Theta}^z W_k^S I_{\Theta}^x \\
&\quad + \left(I - I_{\Theta}^x W_k^S I_{\Theta}^z H^S \right) P_{k|k-1}^S \left(I - I_{\Theta}^x W_k^S I_{\Theta}^z H^S \right)^T \Big\} \\
&- \left(\sum_{\Theta_k} P\{\Theta_k | Z^k\} I_{\Theta}^x W_k^S I_{\Theta}^z \nu_k^S(\Theta) \right) \\
&\times \left(\sum_{\Theta_k} P\{\Theta_k | Z^k\} I_{\Theta}^x W_k^S I_{\Theta}^z \nu_k^S(\Theta) \right)^T. \quad (81)
\end{aligned}$$

REFERENCES

- [1] M. Athans, R. H. Whiting and M. Gruber
A suboptimal estimation algorithm with probabilistic editing for false measurements with applications to target tracking with wake phenomena.
IEEE Transactions on Automatic Control, **22**, 3 (June 1977), 372–384.
- [2] Y. Bar-Shalom, K. C. Chang and H. A. P. Blom
Tracking of splitting targets in clutter using an interacting multiple model joint probabilistic data association filter.
In *Proceedings of the 30th IEEE Conference on Decision and Control*, Brighton, UK, Dec. 1991, 2043–2048.
- [3] Y. Bar-Shalom and X. R. Li
Multitarget-Multisensor Tracking: Principles and Techniques. Storrs, CT: YBS Publishing, 1995.
- [4] Y. Bar-Shalom, X. R. Li and T. Kirubarajan
Estimation with Application to Tracking and Navigation. New York: Wiley-Interscience, 2001.
- [5] Y. Bar-Shalom
Tracking methods in a multitarget environment.
IEEE Transactions on Automatic Control, **23**, 4 (1978), 618–626.
- [6] D. K. Barton
Modern Radar System Analysis. Norwood, MA: Artech House, 1988.
- [7] S. Blake and S. Watts
A multitarget track-while-scan filter.
In *Proceedings of the IEE Radar 87 Conference*, London, England, Oct. 1987.
- [8] E. A. Bloem and H. A. P. Blom
Joint probabilistic data association methods avoiding track coalescence.
In *Proceedings of the 34th IEEE Conference on Decision and Control*, vol. 3, New Orleans, LA, Dec. 1995, 2752–2757.
- [9] H. A. P. Blom and E. A. Bloem
Probabilistic data association avoiding track coalescence.
IEEE Transactions on Automatic Control, **45**, 2 (Feb. 2000), 247–259.
- [10] H. Chen, T. Kirubarajan and Y. Bar-Shalom
Comparison of centralized and distributed tracking algorithms using air-to-air scenarios.
In *Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets*, vol. 4048, July 2000, 440–451.
- [11] H. Chen, T. Kirubarajan and Y. Bar-Shalom
Performance limits of track-to-track fusion versus centralized estimation: Theory and application [sensor fusion].
IEEE Transactions on Aerospace and Electronic Systems, **39**, 2 (Apr. 2003), 386–400.
- [12] H. Chen, K. Zhang and X. R. Li
Optimal data compression for multisensor target tracking with communication constraints.
In *Proceedings of the 43rd IEEE Conference on Decision and Control*, vol. 3, Dec. 2004, 2650–2655.
- [13] R. J. Fitzgerald
Development of practical PDA logic for multitarget tracking by microprocessor.
In Y. Bar-Shalom (Ed.), *Multitarget Multisensor Tracking*. Norwood, MA: Artech House, 1990, 1–23.
- [14] T. E. Fortmann, Y. Bar-Shalom and M. Scheffe
Sonar tracking of multiple targets using joint probabilistic data association.
IEEE Journal of Oceanic Engineering, **8**, 3 (July 1983), 173–184.
- [15] P. P. Gandhi and S. A. Kassam
Analysis of CFAR processors in homogeneous background.
IEEE Transactions on Aerospace and Electronic Systems, **24**, 4 (July 1988), 427–445.
- [16] X. Lurton
An Introduction to Underwater Acoustics. New York: Springer, 2002.
- [17] D. Musicki and R. Evans
Joint integrated probabilistic data association: JIPDA.
IEEE Transactions on Aerospace and Electronic Systems, **40**, 3 (July 2004), 1093–1099.
- [18] D. Musicki, R. Evans and S. Stankovic
Integrated probabilistic data association.
IEEE Transactions on Automatic Control, **39**, 6 (June 1994), 1237–1241.
- [19] L. Y. Pao and C. W. Frei
A comparison of parallel and sequential implementations of a multisensor multitarget tracking algorithm.
In *Proceedings of the American Control Conference*, vol. 3, Seattle, WA, June 1995, 1683–1687.
- [20] A. Papoulis and S. U. Pillai
Probability, Random Variables and Stochastic Processes 4th Edition. New York: McGraw-Hill, 2002.
- [21] A. Rødningsby and Y. Bar-Shalom
Tracking of divers using a probabilistic data association filter with a bubble model.
IEEE Transactions on Aerospace and Electronic Systems, 2009.
- [22] A. Rødningsby, Y. Bar-Shalom, O. Hallingstad and J. Glattetre
Multitarget tracking in the presence of wakes.
In *Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, 2008, 1536–1543.
- [23] J. P. Serra
Image Analysis and Mathematical Morphology. London, UK: Academic Press, 1982.
- [24] W. Smith
Modern Optical Engineering: The Design of Optical Systems (2nd ed.). New York: McGraw-Hill, 1990.
- [25] S. M. Tonissen and Y. Bar-Shalom
Maximum likelihood track-before-detect with fluctuating target amplitude.
IEEE Transactions on Aerospace and Electronic Systems, **34**, 3 (July 1998), 796–808.
- [26] R. Urick
Principles of Underwater Sound (3rd ed.). New York: McGraw-Hill, 1983.



Anders Rødningsby was born on May 19, 1979 in Kongsberg, Norway. He received the B.E. degree from Gjøvik University College, Norway, in electrical engineering and the M.S. degree from the University of Oslo, Norway, in applied mathematics, mechanics and numerical physics, in 2003 and 2005, respectively.

He is currently a research scientist at the Norwegian Defence Research Establishment (FFI), and a Ph.D. candidate at the Norwegian University of Science and Technology. His research interests include detection theory, navigation and target tracking.

Yaakov Bar-Shalom was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion, Israel Institute of Technology, in 1963 and 1967 and the Ph.D. degree from Princeton University in 1970, all in electrical engineering.

From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, CA. Currently he is Board of Trustees Distinguished Professor in the Dept. of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering at the University of Connecticut. He is also Director of the ESP (Estimation and Signal Processing) Lab.

His current research interests are in estimation theory and target tracking and has published over 370 papers and book chapters in these areas and in stochastic adaptive control. He coauthored the monograph *Tracking and Data Association* (Academic Press, 1988), the graduate texts *Estimation and Tracking: Principles, Techniques and Software* (Artech House, 1993), *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction* (Wiley, 2001), the advanced graduate text *Multitarget-Multisensor Tracking: Principles and Techniques* (YBS Publishing, 1995), and edited the books *Multitarget-Multisensor Tracking: Applications and Advances* (Artech House, Vol. I, 1990; Vol. II, 1992; Vol. III, 2000).

He has been elected Fellow of IEEE for “contributions to the theory of stochastic systems and of multitarget tracking.” He has been consulting to numerous companies and government agencies, and originated the series of Multitarget-Multisensor Tracking short courses offered via UCLA Extension, at Government Laboratories, private companies and overseas.

During 1976 and 1977 he served as Associate Editor of the IEEE Transactions on Automatic Control and from 1978 to 1981 as Associate Editor of Automatica. He was Program Chairman of the 1982 American Control Conference, General Chairman of the 1985 ACC, and Co-Chairman of the 1989 IEEE International Conference on Control and Applications. During 1983–87 he served as Chairman of the Conference Activities Board of the IEEE Control Systems Society and during 1987–89 was a member of the Board of Governors of the IEEE CSS. He was a member of the Board of Directors of the International Society of Information Fusion (1999–2004) and served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002 and Vice President for Publications in 2004–08.

In 1987 he received the IEEE CSS Distinguished Member Award. Since 1995 he is a Distinguished Lecturer of the IEEE AESS and has given numerous keynote addresses at major national and international conferences. He is corecipient of the M. Barry Carlton Award for the best paper in the IEEE Transactions on Aerospace and Electronic Systems in 1995 and 2000 and the 1998 University of Connecticut AAUP Excellence Award for Research. In 2002 he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. He is the recipient of the 2008 IEEE Dennis J. Picard Medal for Radar Technologies and Applications.





Oddvar Hallingstad was born on October 17, 1945 in Hol, Norway. He received the M.E. degree from the Norwegian Technical University (NTH) in 1971 and his Ph.D. in 1978 also from NTH. His thesis covered “Transient stability models: Parameter estimation and model reduction.”

Oddvar joined the Norwegian Defence Research Establishment (FFI) in 1977 where he worked on the inertial navigation system for the new coastal vessels and the inertial navigation system for the Penguin air to sea missile. Since 1989 he has been at The Graduate University Center at Kjeller lecturing estimation theory and being an advisor for master and Ph.D. students. He is also an adjunct professor at The Norwegian University of Science and Technology (NTNU) and a part-time employee at FFI.



John Glattetre was born on November 7, 1944 in Brooklyn, NY and moved to Norway in 1957. He received the B.E. from Oslo Technical College in 1971, the M.E. degree from the Norwegian Technical University in 1977 and his Ph.D. from the Norwegian Technical University in 1986. His thesis covered “Interaction between water borne waves and seismic waves in the ocean bottom: The forward- and inverse problem.”

John was an authorized aircraft radio mechanic in SAS. He performed flight calibration of aeronautical navigation systems (VOR, ILS, VDF) for the Norwegian Directorate of Civil Aviation. After a short period with micro electronics, he joined the Norwegian Defence Research Establishment (FFI) as a research scientist and worked with sound propagation and signal processing in underwater acoustics. John presently does R & D for Kongsberg Maritime AS where his emphasis is on active sonar, sound propagation, design and signal processing including tracking. He is co-advisor for two Ph.D. students. John is a member of the Acoustical Society of America's Technical Committee on Underwater Acoustics.

Track-to-Track Fusion Configurations and Association in a Sliding Window

XIN TIAN

YAAKOV BAR-SHALOM

Track-to-track fusion (T2TF) is very important in distributed tracking systems. Compared to the centralized measurement fusion (CMF), T2TF can be done at a lower rate and thus has potentially lower communication requirements. In this paper we investigate the optimal T2TF algorithms under linear Gaussian (LG) assumption, which can operate at an arbitrary rate for various information configurations. It is also assumed that the tracking system is synchronized. Namely, all the trackers obtain measurements and do track updates simultaneously and there are no communication delays between local trackers and the fusion center (FC). The algorithms presented in this paper can be generalized to asynchronous scenarios. First, the algorithms for T2TF without memory (T2TFwoM) are presented for three information configurations: with no, partial and full information feedback from the FC to the local trackers. As one major contribution of this paper, the impact of information feedback on fusion accuracy is investigated. It is shown that using only the track estimates at the fusion time (T2TFwoM), information feedback will have a negative impact on the fusion accuracy. Then, the algorithms for T2TF with memory (T2TFwM), which are optimal at an arbitrary rate, are derived for configurations with no, partial and full information feedback. It is shown that, operating at full rate, T2TFwM is equivalent to the CMF regardless of information feedback. However, at a reduced rate, a certain amount of degradation in fusion accuracy is unavoidable. In contrast to T2TFwoM, T2TFwM benefits from information feedback.

For nonlinear distributed tracking systems, an approximate implementation of the T2TF algorithms is proposed. It requires less communications between the FC and the local trackers, which allows the algorithms to be implemented in distributed tracking systems with low communication capacity. Simulation results show that the proposed approximate implementation is consistent and has practically no loss in fusion accuracy due to the approximation. For the sensors-target geometry considered, it can meet the performance bound of the CMF at the fusion times.

The problem of track-to-track association (T2TA) is also investigated. The sliding window test for T2TA, which uses track estimates within a time window, is derived. It accounts exactly for the crosscovariances among the track estimates and yields false alarm rates that match the theoretical values. To evaluate the test power when using more data frames, a comparison between the single time association test and the sliding window test is performed. Counter-intuitively, it is shown that the belief “the longer the window, the greater the test power” is not always correct.

Manuscript received January 4, 2008; revised January 2, 2009; released for publication September 4, 2009.

Refereeing of this contribution was handled by Chee-Yee Chong.

Authors' address: Dept. of Electrical and Computer Engineering, University of Connecticut, Storrs CT 06269-2157, E-mail: ({xin.tian, ybs}@engr.uconn.edu).

1557-6418/09/\$17.00 © 2009 JAIF

1. INTRODUCTION

In a multisensor tracking system, the fusion center (FC) is meant to gather and process information from local sensors or trackers. There are generally two approaches for this purpose. One is the centralized measurement fusion (CMF), in which the local measurements are sent directly to the FC, where the central tracker performs measurement to track association and track update. The other approach is track-to-track fusion (T2TF) in which local tracks are sent to the FC where tracks of the same target are fused for improved accuracy. In this paper, each track is assumed to be generated by a Kalman filter, which is optimal under linear Gaussian (LG) assumption and the same as the linear minimum mean square error (LMMSE) estimator for linear systems without the Gaussian assumption. It is also assumed that the tracking system is synchronized. Namely, all the trackers obtain measurements and perform track updates simultaneously and there are no communication delays between local trackers and the FC. The T2TF algorithms presented in this paper can be generalized to asynchronous cases which will be discussed in [26]. Although the CMF approach produces the best results, it requires constant and reliable communication links between local sensors and the FC. Lags in the communication links will result in out-of-sequence measurements (OOSM), thus requiring special algorithms [2]. If the communication links become saturated because there are too many measurements to transmit, the sensor network will lose information and might fail. The T2TF approach is more attractive for practical implementations. It allows the local trackers to communicate with the FC once in a while, sending local tracks for T2TF and possibly receiving as feedback the fused tracks from the FC. The major benefit is that there is no restriction on when and how often the local tracks should be transmitted. This can reduce the requirements on the capacity of the communication links.

For the problem of T2TF, the crosscorrelation among tracks of the same target due to common process noises was first observed in [1], where a formula for the calculation of the crosscovariance was also presented. Based on the formula in [1], the algorithm for the one-scan T2TF, i.e., T2TF without memory (T2TFwoM) was studied in [5], which derived the algorithm for T2TFwoM without information feedback (T2TFwoMnf) at an arbitrary rate (see also [17, 18, 19, 20, 21, 22, 27]). Another type of T2TF algorithm—the information matrix fusion (IMF)—was proposed in [13, 25]. Note that, unlike T2TFwoM, the IMF belongs to the class of fuser with memory, since it uses track estimates from the previous fusion. The IMF is equivalent to the CMF when the fuser is operating at full rate [9, 13]. Comparisons between the IMF and the T2TFwoMnf can be found in [8, 9], where it is shown that, operating at full rate, the T2TFwoMnf is not as accurate as the IMF, and it was concluded that the suboptimality of

T2TFwoMnf is because it is optimal only in ML sense.¹ However, the IMF is not optimal when the fuser is operating at reduced rate and, as reported in [10], it causes inconsistency and even divergence. A simulation based comparison on existing fusion algorithms can also be found in [23].

In this paper, the T2TF algorithms that can operate at an arbitrary rate are investigated for various information configurations:

1. T2TFwoMnf (T2TFwoM with no information feedback)
2. T2TFwoMpf (T2TFwoM with partial information feedback)
3. T2TFwoMff (T2TFwoM with full information feedback)
4. T2TFwMnf (T2TFwM with no information feedback)
5. T2TFwMpf (T2TFwM with partial information feedback)
6. T2TFwMff (T2TFwM with full information feedback)

Except for T2TFwoMnf presented in [5], the results for all the other configurations are new. The impact of information feedback and memory on the accuracy of T2TF is thoroughly examined.

For T2TFwoM, depending on the existence of information feedback, the three information configurations [6, 15] are illustrated in Fig. 1. Suppose there are two tracks (that pertain to the same target) which are fused at certain times. The first configuration is the T2TFwoM without information feedback (T2TFwoMnf) [3], designated as Type IIa configuration for multisensor tracking in [6]. As indicated in Fig. 1(a), the two local tracks evolve independently without the information from each other, thus the improved accuracies are achieved only at the fusion times at the FC. The second configuration is the T2TFwoM with partial information feedback (T2TFwoMpf) which belongs to the Type IIb configuration in [6]. In this case, as shown in Fig. 1(b), track 1 is fused with track 2 and continues with the fused track (feedback) from the FC. However, track 2 does not receive the fused track in view of the partial information feedback. The third configuration is the T2TFwoM with full information feedback (T2TFwoMff), which also belongs to the Type IIb configuration in [6]. As shown in Fig. 1(c), both local trackers receive and continue with the fused track.

As shown in this paper T2TFwoM has a degradation in fusion accuracy compared to the CMF, and information feedback has a negative impact on the fusion accuracy of T2TFwoM. However the degradation in fusion accuracy of T2TFwoM can be recovered by using also the track estimates from the previous fusion. Accordingly,

¹The actual reason that T2TFwoMnf is (slightly) inferior to IMF operating at full rate (when it is algebraically equivalent to CMF, see [6] Section 8.6) is the lack of memory of T2TFwoMnf. This is discussed in detail in Section 2.4 and Section 3.4.

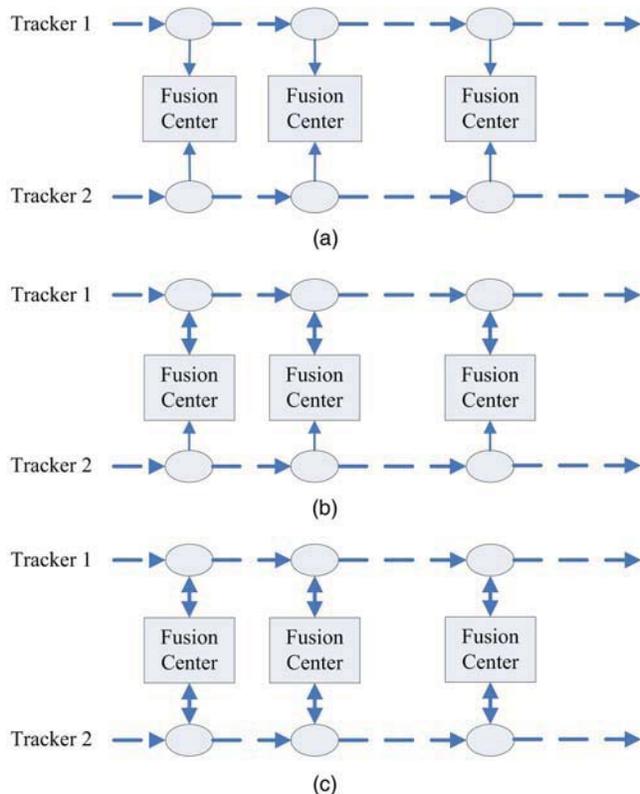


Fig. 1. Information configurations for T2TFwoM (horizontal axis is time). (a) T2TFwoM with no feedback. (b) T2TFwoM with partial feedback (Fusion Center to Tracker 1). (c) T2TFwoM with full feedback (Fusion Center to Tracker 1 and Tracker 2).

the algorithms for T2TFwM are derived for configurations with no, partial and full information feedback. It is shown that, when operating at full rate, T2TFwM is equivalent to the CMF (which is the global optimum) regardless of information feedback. However, at reduced rate, a certain amount of loss in fusion accuracy is unavoidable, and in contrast to the case of T2TFwoM, information feedback improves the fusion accuracy of T2TFwM. Furthermore, unlike the IMF, the T2TFwM algorithms derived in this paper are optimal at any rate.

As pointed out in [8], the major difficulty for the practical implementation of the optimal T2TF algorithm is that it requires all the local filter gains and observation matrices since the last fusion. In nonlinear distributed tracking systems, the local information is not directly available at the FC. In view of this, we propose an approximate implementation of the T2TF algorithms. It is based on the idea of reconstructing local information at the FC with minimum amount of information from the local trackers, which has much less communication requirements than the transmission of those local matrices [12]. Simulation results show that this approximate implementation is consistent and has practically no loss in accuracy due to the approximation.

Studies on the problem of track-to-track association (T2TA) can be found in [4, 24], in which the tests for T2TA are made based on a single frame of data. In [16], it is claimed that the test based on an average of

TABLE I
List of Acronyms

CMF	Centralized Measurement Fusion	configuration IV
IMF	Information Matrix Fusion	special configuration II ^M
T2TF	Track-to-Track Fusion	
T2TFwoM	Track-to-Track Fusion without memory	configuration II
T2TFwoMnf	T2TFwoM with no information feedback	configuration II _{nf}
T2TFwoMpf	T2TFwoM with partial information feedback	configuration II _{pf}
T2TFwoMff	T2TFwoM with full information feedback	configuration II _{ff}
T2TFwM	Track-to-Track Fusion with memory	configuration II ^M
T2TFwMnf	T2TFwM with no information feedback	configuration II _{nf} ^M
T2TFwMpf	T2TFwM with partial information feedback	configuration II _{pf} ^M
T2TFwMff	T2TFwM with full information feedback	configuration II _{ff} ^M

the single time tests within a time window has improved performance over the single time test. However, this conclusion was drawn ignoring the state errors' cross-correlation in time [7]. In this paper the *chi-square based sliding window test for T2TA*, which uses track estimates within a time window, is derived. It accounts exactly for all the crosscovariances among the track estimates and yields false alarm rates that match the theoretical values. To evaluate the test power when using more data frames (longer window), a comparison between the single time test and the sliding window test is performed. Counterintuitively, it is shown that *the belief "the longer the window, the more the power" is not necessarily correct*. This is because the power of the test depends on both the noncentrality parameter and the degrees of freedom of the (chi-square) test statistic. When the multiple frames of data selected for T2TA are strongly correlated, which happens for motion with low process noises (because the filter has "longer memory" in this case), the gain in the noncentrality parameter by using more data frames is too small to overcome the negative impact of the increased degrees of freedom on the power of the test. Thus, the sliding window test may be counterproductive and has *lower power* than the single time test.

All the results presented are optimal under the LG assumption.² The paper is organized as follows. Section 2 discusses the algorithms for T2TFwoM with no, partial and full information feedback. Section 3 derives the algorithms for T2TFwM with no, partial and full information feedback and shows the impact of information feedback on T2TFwM. In Section 4, the approximate implementation of the T2TF algorithms is proposed and evaluated in a tracking scenario with a nonlinear measurement model. The problem of T2TA test is investigated in Section 5, where the sliding window test is derived and compared with the single time test. Section 6 summarizes the paper with conclusions. For the convenience of readers, Table I lists the acronyms used in this paper and extends the configurations discussed in [6].

2. TRACK-TO-TRACK FUSION WITHOUT MEMORY (T2TFwoM) AT AN ARBITRARY RATE

²In the case of fusion the algorithms presented constitute the LMMSE fuser, also called BLUE in, e.g., [18].

This section investigates the algorithms of T2TFwoM (configuration II [6]) at an arbitrary rate. Section 2.1 formulates the problem. In Section 2.2 the T2TFwoM algorithms are presented for information configurations: with no, partial and full information feedback. Section 2.3 presents the simulation results that compare the fusion accuracies of T2TFwoMnf, T2TFwoMpf, T2TFwoMff and CMF. This leads to the observation that, in T2TFwoM, information feedback will cause a degradation of the fusion accuracy. This phenomenon is further explained in Section 2.4.

2.1. Problem Formulation: T2TFwoM

Consider the basic scenario with two local trackers (designated as 1 and 2) at different locations. Each tracker obtains measurements with its local sensor and maintains local tracks of the targets. For the sake of simplicity, it is assumed that the system operates in a synchronous fashion, where all the trackers obtain measurements and do local track updates simultaneously with sampling interval T . Communication links, which have no delay in time, are available between the FC and the local trackers. Each local tracker is allowed to communicate with the FC once in a while, sending its tracks to the FC and possibly receiving the fused tracks (when there is information feedback). At the FC, the fusion of the tracks of a target from trackers 1 and 2 is formulated as follows.³ Let $\hat{x}_1(k|k)$, $P_1(k|k)$ and $\hat{x}_2(k|k)$, $P_2(k|k)$ represent the two local tracks at the fusion time. Assuming the crosscovariance of the two tracks $P_{12}(k|k)$ is available at the FC, T2TFwoM should be performed, so that

$$\begin{aligned} & [\hat{x}_c(k|k), P_c(k|k)] \\ & = \mathbf{f}[\hat{x}_1(k|k), P_1(k|k), \hat{x}_2(k|k), P_2(k|k), P_{12}(k|k)] \end{aligned} \quad (1)$$

where $\hat{x}_c(k|k)$ and $P_c(k|k)$ represent the fused track. After the fusion, the local tracks and their crosscovariance should also be updated to $\hat{x}_1^*(k|k)$, $P_1^*(k|k)$, $\hat{x}_2^*(k|k)$,

³Association (see [6] Section 8.4) is assumed to have been already performed.

$P_2^*(k | k)$ and $P_{12}^*(k | k)$ according to the information configuration of the fusion (possible feedback from the FC). Throughout the paper, superscript “*” is used to indicate post-fusion tracks. Note that (1) implies that only the local track estimates at the fusion time are used for T2TFwoM, i.e., this is a fuser without memory of fused and local track estimates from the previous fusion time.⁴

2.2. The Algorithms for T2TFwoM

If the local tracks $\hat{x}_1(k | k)$, $P_1(k | k)$, $\hat{x}_2(k | k)$, $P_2(k | k)$ and their crosscovariance $P_{12}(k | k)$ are available at the FC, the optimal⁵ T2TFwoM can be done according to Eqs. (8.4.4-4)–(8.4.4-5) in [6], namely,

$$\begin{aligned} \hat{x}_c(k | k) &= \hat{x}_1(k | k) + [P_1(k | k) - P_{12}(k | k)] \\ &\quad \cdot [P_1(k | k) + P_2(k | k) - P_{12}(k | k) - P_{21}(k | k)]^{-1} \\ &\quad \cdot [\hat{x}_2(k | k) - \hat{x}_1(k | k)] \\ &= \hat{x}_1(k | k) + K_{12}(k)[\hat{x}_2(k | k) - \hat{x}_1(k | k)] \end{aligned} \quad (2)$$

$$\begin{aligned} P_c(k | k) &= P_1(k | k) - [P_1(k | k) - P_{12}(k | k)] \\ &\quad \cdot [P_1(k | k) + P_2(k | k) - P_{12}(k | k) - P_{21}(k | k)]^{-1} \\ &\quad \cdot [P_1(k | k) - P_{21}(k | k)] \end{aligned} \quad (3)$$

where

$$P_{12}(k | k) = P_{21}(k | k)' = \text{Cov}[\tilde{x}_1(k | k), \tilde{x}_2(k | k)]. \quad (4)$$

To calculate $P_{12}(k | k)$, suppose the previous fusion was performed at discretized time l , after which one has the errors

$$\tilde{x}_1^*(l | l) = \hat{x}_1^*(l | l) - x(l) \quad (5)$$

$$\tilde{x}_2^*(l | l) = \hat{x}_2^*(l | l) - x(l) \quad (6)$$

where $x(l)$ denotes the true state of the target at l . Let

$$P_1^*(l | l) = \text{Cov}[\tilde{x}_1^*(l | l), \tilde{x}_1^*(l | l)] \quad (7)$$

$$P_2^*(l | l) = \text{Cov}[\tilde{x}_2^*(l | l), \tilde{x}_2^*(l | l)] \quad (8)$$

$$P_{12}^*(l | l) = \text{Cov}[\tilde{x}_1^*(l | l), \tilde{x}_2^*(l | l)]. \quad (9)$$

From Eq. (8.4.2-2) in [6], one has

$$\begin{aligned} \tilde{x}_s(l + 1 | l + 1) &= [I - K_s(l + 1)H_s(l + 1)]F(l)\tilde{x}_s(l | l) \\ &\quad - [I - K_s(l + 1)H_s(l + 1)]v(l) \\ &\quad + K_s(l + 1)w_s(l + 1) \quad s = 1, 2, \end{aligned} \quad (10)$$

⁴A fuser with memory (of the previous track estimates) uses the track estimates from the previous fusion time, which, as shown in Section 3.4, improves fusion accuracy.

⁵MMSE if all the estimation errors are Gaussian and LMMSE otherwise [7].

Using (10) recursively for both the central and local tracks from discrete time l to k , it follows that

$$\begin{aligned} \tilde{x}_s(k | k) &= W_s^e(k, l)\tilde{x}_s^*(l | l) + \sum_{i=l+1}^k W_s^v(k, i-1)v(i-1) \\ &\quad + \sum_{i=l+1}^k W_s^w(k, i)w_s(i), \quad s = 1, 2 \end{aligned} \quad (11)$$

where the weights are defined as

$$W_s^e(k, l) = \prod_{i=0}^{k-l-1} [I - K_s(k-i)H_s(k-i)]F(k-i-1) \quad (12)$$

$$\begin{aligned} W_s^v(k, i-1) &= - \left\{ \prod_{j=0}^{k-i-1} [I - K_s(k-j)H_s(k-j)]F(k-j-1) \right\} \\ &\quad \cdot [I - K_s(i)H_s(i)] \end{aligned} \quad (13)$$

and

$$W_s^w(k, i) = \left\{ \prod_{j=0}^{k-i-1} [I - K_s(k-j)H_s(k-j)]F(k-j-1) \right\} K_s(i) \quad (14)$$

in which $K_s(i)$, $i = l + 1, \dots, k$ are the Kalman filter gains and $H_s(i)$ are the observation matrices at local tracker s and $F(i-1)$ are the state transition matrices. Eq. (11) is the expression of the errors of the tracks from Kalman filters as weighted sums of the previous error at a certain point and the intervening process and measurement noises. The significance of this expression is that it shows explicitly all the sources of uncertainty and provides the general tool for the derivations of the T2TF and T2TA algorithms in the absence or presence of memory and feedback.

From (11) and the whiteness assumption of the measurement noises and the process noises, the crosscovariance $P_{12}(k | k)$, required by the T2TFwoM given in (2)–(3), can be calculated as

$$\begin{aligned} P_{12}(k | k) &= W_1^e(k, l)P_{12}^*(l | l)W_2^e(k, l)' \\ &\quad + \sum_{i=l+1}^k W_1^v(k, i-1)Q(i-1)W_2^v(k, i-1)' \end{aligned} \quad (15)$$

where $Q(i)$ is the covariance of the process noises at time i . Similarly to (11), the error of the fused track (2) can be expressed as

$$\begin{aligned} \tilde{x}_c(k | k) &= \tilde{x}_1(k | k) + K_{12}(k)[\tilde{x}_2(k | k) - \tilde{x}_1(k | k)] \\ &= [I - K_{12}(k)]\tilde{x}_1(k | k) + K_{12}(k)\tilde{x}_2(k | k). \end{aligned} \quad (16)$$

After the fusion, local tracks 1 and 2 and their cross-covariance should be updated according to the information configuration.

In configuration T2TFwoMnf (see Fig. 1(a)), one has

$$\hat{x}_1^*(k|k) = \hat{x}_1(k|k) \quad (17)$$

$$P_1^*(k|k) = P_1(k|k) \quad (18)$$

$$\hat{x}_2^*(k|k) = \hat{x}_2(k|k) \quad (19)$$

$$P_2^*(k|k) = P_2(k|k) \quad (20)$$

$$P_{12}^*(k|k) = P_{12}(k|k) \quad (21)$$

where $P_{12}(k|k)$ is given in (15).

In configuration T2TFwoMpf (see Fig. 1(b))

$$\hat{x}_1^*(k|k) = \hat{x}_c(k|k) \quad (22)$$

$$P_1^*(k|k) = P_c(k|k) \quad (23)$$

$$\hat{x}_2^*(k|k) = \hat{x}_2(k|k) \quad (24)$$

$$P_2^*(k|k) = P_2(k|k) \quad (25)$$

and according to (16)

$$P_{12}^*(k|k) = [I - K_{12}(k)]P_{12}(k|k) + K_{12}(k)P_2(k|k). \quad (26)$$

In configuration T2TFwoMff (see Fig. 1(c))

$$\hat{x}_2^*(k|k) = \hat{x}_1^*(k|k) = \hat{x}_c(k|k) \quad (27)$$

$$P_2^*(k|k) = P_1^*(k|k) = P_c(k|k) \quad (28)$$

$$P_{12}^*(k|k) = P_c(k|k). \quad (29)$$

The algorithm of T2TFwoM is summarized as follows:

- At the FC, the local tracks are fused according to (2)–(3).
- The fusion can be done exactly, if the following data are available:
 - (i) The local tracks to be fused: $\hat{x}_1(k|k)$, $P_1(k|k)$ and $\hat{x}_2(k|k)$, $P_2(k|k)$
 - (ii) The covariances and crosscovariance from the previous fusion at time l : $P_1^*(l|l)$, $P_2^*(l|l)$, $P_{12}^*(l|l)$ (see (7)–(8))—needed for the calculation of the current crosscovariance.
 - (iii) The local weights (12)–(13).
- Depending on the information configuration, the local tracks are updated accordingly using (17)–(21) for T2TFwoMnf, or (22)–(26) for T2TFwoMpf, or (27)–(29) for T2TFwoMff.

The algorithm of T2TFwoM has no theoretical limit on the number of the local trackers. Only the crosscovariances among all the tracks of the same target need to be properly calculated. See [11] for the n -sensors version of the fusion equations (2)–(3). The use of the results from [11] in the general case requires Eqs. (11)–(13) for each sensor.

TABLE II
Fuser Variances (at fusion times) in Steady State
(fusion interval: 5 s)

Fusion Type	FC Track at Fusion Time	
	Pos	Vel
T2TFwoMff	133	6.29
T2TFwoMpf	131	6.30
T2TFwoMnf	125	6.30
CMF	119	6.03

2.3. Comparison of the T2TFwoM Algorithms and the CMF

The algorithms for T2TFwoM are evaluated first in the following tracking scenario. The target state is defined as $[x \dot{x}]'$. The target motion is modeled as the discrete white noise acceleration (DWNA) model in [7], Section 6.3.2. It is assumed that two sensors obtain position measurements of the target with a sampling interval of $T = 1$ s. The standard deviation of the measurement noise is $\sigma_w = 30$ m and the process noise variance is $q = 1 \text{ m}^2/\text{s}^4$. T2TFwoM takes place every 5 s, i.e., at a reduced rate.

All the fusers are consistent in the simulations (their covariance calculations are exact). In view of the consistency, the performance comparison can be made using the calculated covariances. Table II shows the steady state variances of position and velocity at the FC. All the fused tracks are more accurate than the single-sensor (local) tracks without fusion, which have steady state variances as 205 in position and 7.26 in velocity. Note that at the fusion time the position estimates of all the fused tracks have a small degradation compared to the CMF: 5% for T2TFwoMnf, 10% for T2TFwoMpf, 12% for T2TFwoMff. This shows that T2TFwoM has a degradation in fusion accuracy compared to the CMF and the degradation increases in the presence of information feedback. This apparently counterintuitive result is further discussed in the next subsection.

2.4. The Impact of Information Feedback on T2TFwoM

To show the impact of information feedback on T2TFwoM, consider the following two-step scalar estimation problem. At time 0, two local estimators have independent prior information: $\bar{x}_1 \sim N(x(0), P_1)$ at estimator 1 and $\bar{x}_2 \sim N(x(0), P_2)$ at estimator 2. At time 1, $x(1) = x(0) + v_1$ where the process noise $v_1 \sim N(0, Q)$. The estimators have independent measurements of the state: $z_1 = x(1) + w_1$ ($w_1 \sim N(0, R_1)$) at estimator 1 and $z_2 = x(1) + w_2$ ($w_2 \sim N(0, R_2)$) at estimator 2. The errors in the prior information, the process noise and the measurement noises are all independent. For the sake of simplicity it is assumed that $P_1 = P_2 = R_1 = R_2 = 1$ and $Q = 1/2$. Consider the fusion at time 1 and the fuser uses only the track estimates at time 1, namely T2TFwoM. Fig. 2 shows the information flow of the centralized

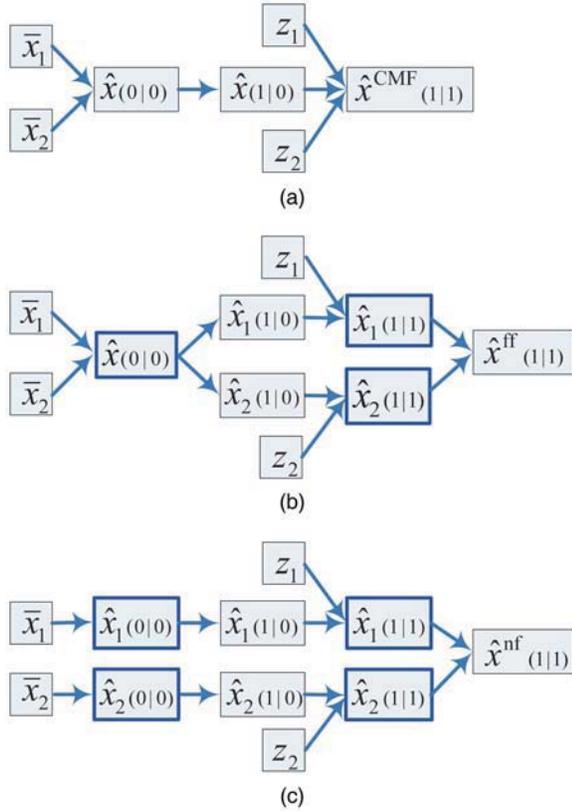


Fig. 2. Information flow (CMF and T2TFwoM). (a) CMF (configuration IV). (b) T2TFwoMff (configuration IIff). (c) T2TFwoMnf (configuration IIInf).

measurement fusion (CMF), T2TFwoM with full information feedback (T2TFwoMff) and T2TFwoM with no information feedback (T2TFwoMnf). Note that in T2TFwoMff the information feedback (sharing) occurs at time 0 in this example.

Using CMF (Configuration IV in Section 8.2.5 of [6]), one has

$$\hat{x}^{\text{CMF}}(1|1) = \frac{1}{6}\bar{x}_1 + \frac{1}{6}\bar{x}_2 + \frac{1}{3}z_1 + \frac{1}{3}z_2 \quad (30)$$

with error (using the weighted sum form)

$$\tilde{x}^{\text{CMF}}(1|1) = \frac{1}{6}\tilde{x}_1 + \frac{1}{6}\tilde{x}_2 - \frac{1}{3}v_1 + \frac{1}{3}w_1 + \frac{1}{3}w_2 \quad (31)$$

where \tilde{x}_1 and \tilde{x}_2 denote the errors of \bar{x}_1 and \bar{x}_2 . It is easy to calculate the covariance

$$\text{Cov}[\hat{x}^{\text{CMF}}(1|1)] = \frac{1}{3}. \quad (32)$$

In T2TFwoMff,⁶ one has

$$\hat{x}^{\text{ff}}(1|1) = \frac{1}{4}\bar{x}_1 + \frac{1}{4}\bar{x}_2 + \frac{1}{4}z_1 + \frac{1}{4}z_2 \quad (33)$$

with error

$$\tilde{x}^{\text{ff}}(1|1) = \frac{1}{4}\tilde{x}_1 + \frac{1}{4}\tilde{x}_2 - \frac{1}{2}v_1 + \frac{1}{4}w_1 + \frac{1}{4}w_2 \quad (34)$$

⁶Note that, following Section 8.2.3 of [6], Configuration IIb does not have a memory of past estimates at the FC. It is T2TFwoM.

and

$$\text{Cov}[\hat{x}^{\text{ff}}(1|1)] = \frac{3}{8}. \quad (35)$$

In T2TFwoMnf (Configuration IIa in Section 8.2.3 of [6]), it follows that

$$\hat{x}^{\text{nf}}(1|1) = \frac{1}{5}\bar{x}_1 + \frac{1}{5}\bar{x}_2 + \frac{3}{10}z_1 + \frac{3}{10}z_2 \quad (36)$$

with error

$$\tilde{x}^{\text{nf}}(1|1) = \frac{1}{5}\tilde{x}_1 + \frac{1}{5}\tilde{x}_2 - \frac{2}{5}v_1 + \frac{3}{10}w_1 + \frac{3}{10}w_2 \quad (37)$$

and

$$\text{Cov}[\hat{x}^{\text{nf}}(1|1)] = \frac{17}{50}. \quad (38)$$

Then one has

$$\text{Cov}[\hat{x}^{\text{CMF}}(1|1)] < \text{Cov}[\hat{x}^{\text{nf}}(1|1)] < \text{Cov}[\hat{x}^{\text{ff}}(1|1)]. \quad (39)$$

There are losses in accuracy in T2TFwoMff and T2TFwoMnf compared to the CMF, although they are relatively small, due to the large process noise in the example. Comparing (31), (34) and (37), it can be seen that the weights of the measurements are lower in T2TFwoMnf than in the CMF. They become even lower in T2TFwoMff due to the information feedback, which leads to the further loss in fusion accuracy.

3. THE ALGORITHMS FOR T2TFwM AT AN ARBITRARY RATE

The results in Section 2.4 show that, at full rate, T2TFwoM is less accurate than the CMF and information feedback is detrimental to T2TFwoM. However, the IMF is equivalent to the CMF when operating at full rate [13]. In this case, T2TFwoM is inferior to the IMF. This is because T2TFwoM uses only local estimates at the fusion time, which contain most but not all of the information for T2TF. In contrast, the IMF belongs to the class of T2TFwM. However, at *reduced rate*, the IMF algorithm is not optimal anymore.

To account for the information from the fused and local track estimates from the previous fusion time,⁷ the algorithm for T2TFwM at an *arbitrary rate* is derived in the next three subsections for configurations with no, partial and full information feedback, designated as T2TFwMnf, T2TFwMpf and T2TFwMff respectively. Fig. 3 shows the information flow of the three configurations.

3.1. The Algorithm for T2TFwMnf

As shown in Fig. 3(a), for T2TFwMnf, at fusion time k , the track estimates to be fused are local track

⁷This implies one-step memory. In fact, there is no need for more, since one-step memory summarizes, under the LG (or LMMSE) assumptions, all the information from the previous track estimates. This is confirmed by the simulation results in Section 3.4, which show that, at full rate, T2TFwM (using one-step memory) yields the globally optimal fusion results.

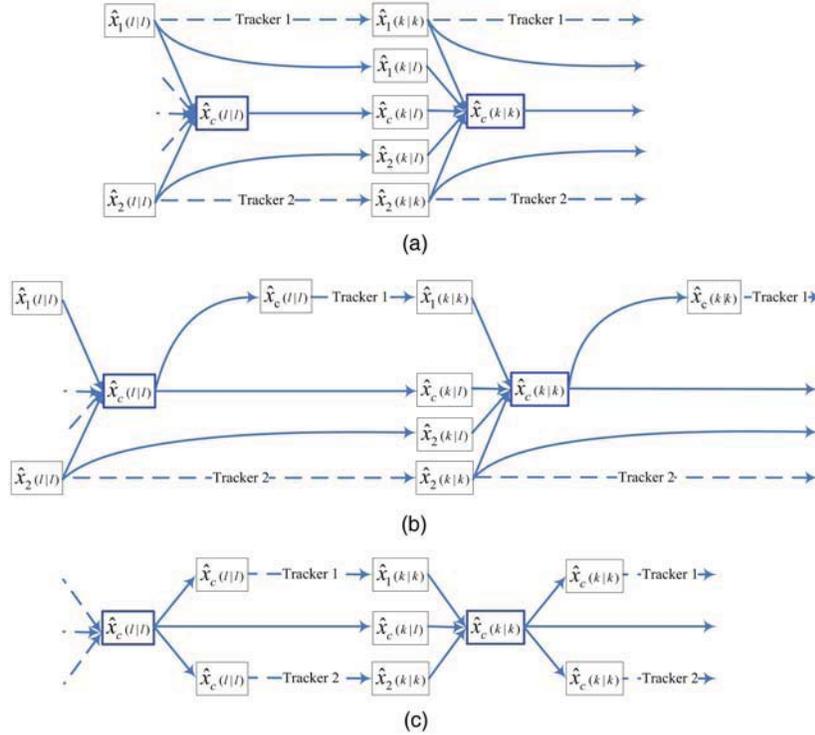


Fig. 3. Information flow: T2TFwM at an arbitrary rate. (a) T2TFwM with no information feedback (one cycle: from fusion time l to the next fusion time k). (b) T2TFwM with partial information feedback (one cycle: from fusion time l to the next fusion time k). (c) T2TFwM with full information feedback (one cycle: from fusion time l to the next fusion time k).

estimates $\hat{x}_1(k|k)$, $\hat{x}_2(k|k)$ and the predicted track estimates $\hat{x}_1(k|l)$, $\hat{x}_2(k|l)$ and $\hat{x}_c(k|l)$ from the previous fusion, where subscript “c” indicates the (fused) track at the FC. Stacking the estimates and the predicted estimates as a vector, one has

$$\mu = [\hat{x}_1(k|k) \quad \hat{x}_2(k|k) \quad \hat{x}_c(k|l) \quad \hat{x}_1(k|l) \quad \hat{x}_2(k|l)]'. \quad (40)$$

The fusion of these track estimates requires the covariance of μ , denoted as $\text{Cov}(\mu)$. To obtain this covariance, a slight modification of (11) gives

$$\tilde{x}_s(k|l) = W_0^e(k,l)\tilde{x}_s^*(l|l) + \sum_{i=l+1}^k W_0^v(k,i-1)v(i-1) \quad (41)$$

where $s = 1, 2, c$ and

$$W_0^e(k,l) = \prod_{i=0}^{k-l-1} F(k-i-1) \quad (42)$$

$$W_0^v(k,i-1) = - \prod_{j=0}^{k-i-1} F(k-j-1) \quad (43)$$

which are obtained by substituting the filter gains $K_s(i)$ in (12) and (13) by zero matrices, since the filter gains are zero in the predicted track estimates. From (12), (13), (42) and (43), $\text{Cov}(\mu)$ can be easily calculated with linear algebra.

For T2TFwMnf, define the following difference matrix

$$M = \begin{bmatrix} I & 0 & -I & 0 & 0 \\ 0 & I & -I & 0 & 0 \\ 0 & 0 & -I & I & 0 \\ 0 & 0 & -I & 0 & I \end{bmatrix} \quad (44)$$

where I denotes identity matrix of appropriate dimension, such that

$$\nu = \begin{bmatrix} \hat{x}_1(k|k) - \hat{x}_c(k|l) \\ \hat{x}_2(k|k) - \hat{x}_c(k|l) \\ \hat{x}_1(k|l) - \hat{x}_c(k|l) \\ \hat{x}_2(k|l) - \hat{x}_c(k|l) \end{bmatrix} = M\mu. \quad (45)$$

Using the standard MMSE estimator from [7], the fused estimate is given by

$$\hat{x}_c(k|k) = \hat{x}_c(k|l) + \text{Cov}[x(k), \nu] \text{Cov}(\nu)^{-1} \nu \quad (46)$$

where $x(k)$ is the true state of the target at time k , and

$$\text{Cov}[x(k), \nu] = -[\text{Cov}(\mu)]_{(3,:)} M' \quad (47)$$

$$\text{Cov}(\nu) = M \text{Cov}(\mu) M' \quad (48)$$

where $[\text{Cov}(\mu)]_{(i,:)}$ denotes the i th row of matrix $\text{Cov}(\mu)$. In (46) $\hat{x}_c(k|l)$ plays the role of the prior at the FC and the other elements of μ play the role of the observations.

From (45)–(48), the fused estimate is

$$\begin{aligned}\hat{x}_c(k|k) &= \hat{x}_c(k|l) - [\text{Cov}(\mu)]_{(3,:)} M' (M \text{Cov}(\mu) M')^{-1} M \mu \\ &= \hat{x}_c(k|l) + K_\mu \mu\end{aligned}\quad (49)$$

where

$$K_\mu \triangleq -[\text{Cov}(\mu)]_{(3,:)} M' (M \text{Cov}(\mu) M')^{-1} M. \quad (50)$$

The fused covariance is

$$\begin{aligned}P_c(k|k) &= P_c(k|l) - \text{Cov}[x(k), \nu] \text{Cov}(\nu)^{-1} \text{Cov}[x(k), \nu]' \\ &= P_c(k|l) - [\text{Cov}(\mu)]_{(3,:)} M' (M \text{Cov}(\mu) M')^{-1} M \\ &\quad \cdot [[\text{Cov}(\mu)]_{(3,:)}]' \\ &= P_c(k|l) + K_\mu [[\text{Cov}(\mu)]_{(3,:)}]'.\end{aligned}\quad (51)$$

For T2TFwMnf the crosscovariances between the fused track and the tracks from trackers 1 and 2 can be obtained from (49) as

$$\begin{aligned}P_{1c}(k|k) &\triangleq \text{Cov}[\hat{x}_1(k|k), \hat{x}_c(k|k)] \\ &= [\text{Cov}(\mu)]_{(1,3)} + [\text{Cov}(\mu)]_{(1,:)} K_\mu'\end{aligned}\quad (52)$$

$$\begin{aligned}P_{2c}(k|k) &\triangleq \text{Cov}[\hat{x}_2(k|k), \hat{x}_c(k|k)] \\ &= [\text{Cov}(\mu)]_{(2,3)} + [\text{Cov}(\mu)]_{(2,:)} K_\mu'\end{aligned}\quad (53)$$

where $[\text{Cov}(\mu)]_{(i,j)}$ is element (i, j) of $\text{Cov}(\mu)$.

Since there is no information feedback, both local tracks are not changed after the fusion is performed at the FC. One has

$$\hat{x}_1^*(k|k) = \hat{x}_1(k|k) \quad (54)$$

$$P_1^*(k|k) = P_1(k|k) \quad (55)$$

$$\hat{x}_2^*(k|k) = \hat{x}_2(k|k) \quad (56)$$

$$P_2^*(k|k) = P_2(k|k) \quad (57)$$

$$P_{12}^*(k|k) = P_{12}(k|k) \quad (58)$$

$$P_{1c}^*(k|k) = P_{1c}(k|k) \quad (59)$$

$$P_{2c}^*(k|k) = P_{2c}(k|k). \quad (60)$$

3.2. The Algorithm for T2TFwMpf

Unlike T2TFwMnf, in T2TFwMpf, one has $\hat{x}_1(k|l) = \hat{x}_c(k|l)$. Consequently, the elements $\hat{x}_1(k|l)$ in (40) should be removed. In this case, redefine μ in (40) and M in (44) as

$$\mu = [\hat{x}_1(k|k) \quad \hat{x}_2(k|k) \quad \hat{x}_c(k|l) \quad \hat{x}_2(k|l)]' \quad (61)$$

and

$$M \triangleq \begin{bmatrix} I & 0 & -I & 0 \\ 0 & I & -I & 0 \\ 0 & 0 & -I & I \end{bmatrix}. \quad (62)$$

It follows that

$$\nu = \begin{bmatrix} \hat{x}_1(k|k) - \hat{x}_c(k|l) \\ \hat{x}_2(k|k) - \hat{x}_c(k|l) \\ \hat{x}_2(k|l) - \hat{x}_c(k|l) \end{bmatrix} = M\mu. \quad (63)$$

Then, similarly to T2TFwMnf, the fused estimate $\hat{x}_c(k|k)$ is obtained using (49) and the fused covariance $P_c(k|k)$ follows from (51) using the modified definitions (61) and (62).

After the fusion, the local tracks and the track cross-covariances are updated as follows

$$\hat{x}_1^*(k|k) = \hat{x}_c(k|k) \quad (\text{feedback}) \quad (64)$$

$$P_1^*(k|k) = P_c(k|k) \quad (\text{feedback}) \quad (65)$$

$$P_{1c}^*(k|k) = P_c(k|k) \quad (\text{feedback}) \quad (66)$$

$$\hat{x}_2^*(k|k) = \hat{x}_2(k|k) \quad (\text{no feedback}) \quad (67)$$

$$P_2^*(k|k) = P_2(k|k) \quad (\text{no feedback}) \quad (68)$$

$$P_{12}^*(k|k) = P_{2c}(k|k) \quad (69)$$

$$P_{2c}^*(k|k) = P_{2c}(k|k) \quad (70)$$

where $P_{2c}(k|k)$ is given by (53) using the modified definitions (61) and (62).

3.3. The Algorithm for T2TFwMff

In contrast to T2TFwMnf, in T2TFwMff, one has $\hat{x}_1(k|l) = \hat{x}_2(k|l) = \hat{x}_c(k|l)$. Consequently, the elements $\hat{x}_1(k|l)$ and $\hat{x}_2(k|l)$ in (40) should be removed. In this case, redefine μ in (40) and M in (44) as

$$\mu = [\hat{x}_1(k|k) \quad \hat{x}_2(k|k) \quad \hat{x}_c(k|l)]' \quad (71)$$

and

$$M = \begin{bmatrix} I & 0 & -I \\ 0 & I & -I \end{bmatrix}. \quad (72)$$

It follows that

$$\nu = \begin{bmatrix} \hat{x}_1(k|k) - \hat{x}_c(k|l) \\ \hat{x}_2(k|k) - \hat{x}_c(k|l) \end{bmatrix} = M\mu. \quad (73)$$

Then, similarly to T2TFwMnf, the fused estimate is obtained using (49) and the fused covariance follows from (51) using the modified definitions (71) and (72).

Due to full information feedback, the local tracks and the track crosscovariances are updated as

$$\hat{x}_1^*(k|k) = \hat{x}_c(k|k) \quad (\text{feedback}) \quad (74)$$

$$P_1^*(k|k) = P_c(k|k) \quad (\text{feedback}) \quad (75)$$

$$P_{1c}^*(k|k) = P_c(k|k) \quad (\text{feedback}) \quad (76)$$

$$\hat{x}_2^*(k|k) = \hat{x}_c(k|k) \quad (\text{feedback}) \quad (77)$$

$$P_2^*(k|k) = P_2(k|k) \quad (\text{feedback}) \quad (78)$$

$$P_{12}^*(k|k) = P_{2c}^*(k|k) = P_c(k|k) \quad (\text{feedback}). \quad (79)$$

Note that in T2TFwMff the crosscovariances between the fused track and the local tracks after the information feedback are the same with the fused covariance (51). This is different from the updated crosscovariances in T2TFwMnf, i.e., (59) and (60).

TABLE III
Fuser and Tracker 1 Calculated Variances at Fusion Times for $N_f = 1$ (full rate), $q = 0.3$, $R_1 = R_2 = 1$

Time		1	2	3	4	5	6
T2TFwMnf	Tracker 1	1.0000	0.5652	0.4639	0.4331	0.4230	0.4196
	Fuser	0.5000	0.3077	0.2743	0.2673	0.2658	0.2654
T2TFwMpf	Fuser	0.5000	0.3077	0.2743	0.2673	0.2658	0.2654
T2TFwMff	Fuser	0.5000	0.3077	0.2743	0.2673	0.2658	0.2654
CMF		0.5000	0.3077	0.2743	0.2673	0.2658	0.2654

TABLE IV
Fuser and Tracker 1 Calculated Variances at Fusion Times for $N_f = 3$ (reduced rate), $q = 0.3$, $R_1 = R_2 = 1$

Time		1	3	6	9	12	15
T2TFwMnf	Tracker1	1.0000	0.4639	0.4196	0.4180	0.4179	0.4179
	Fuser	0.5000	0.2772	0.2698	0.2694	0.2694	0.2694
T2TFwMpf	Fuser	0.5000	0.2763	0.2690	0.2688	0.2688	0.2688
T2TFwMff	Fuser	0.5000	0.2755	0.2683	0.2682	0.2682	0.2682
CMF		0.5000	0.2743	0.2654	0.2653	0.2653	0.2653

3.4. Performance Comparison: T2TFwMnf vs. T2TFwMff and CMF

To evaluate the performance of the optimal T2TF with memory (T2TFwM) at an arbitrary rate, consider the following tracking scenario. The state of the target (taken as a scalar for simplicity) evolves according to

$$x(k) = x(k-1) + v(k), \quad k = 2, 3, \dots \quad (80)$$

where $v(k)$ is the process noise with variance q .

There are two trackers, 1 and 2, taking measurements of the target with measurement noises w_1 and w_2 , namely,

$$z_i(k) = x(k) + w_i(k), \quad i = 1, 2 \quad (81)$$

where $w_i(k)$ are zero-mean Gaussian noises with variance R_i . The two trackers calculate tracks of the target with their own measurements using a Kalman filter. Each local track is initialized at time 1 with the first local measurement. The first T2TF happens at time 1. Then T2TFwM occurs every N_f sampling times.

Table III shows the fuser- and tracker-calculated variances of the errors of the track estimates when the fuser is operating at full rate. It can be seen that, at full rate, the fuser with memory (with no, partial and full information feedback) is equivalent to the CMF.

Table IV shows the fuser- and tracker-calculated variances when the fuser (with memory) is operating at reduced rate. In this case, T2TFwM, with or without information feedback, has a small loss in fusion accuracy compared to the CMF. This is because, in T2TF, information from the common process noises and the common prior information (due to information feedback) appears simultaneously in different local tracks,

which causes their weights (and, consequently, also the weights of the new measurements) in the fused track to deviate from the global optimum. This is similar to the T2TFwoM example discussed in Section 2.4. When T2TFwM is done at full rate, these deviations are fully corrected by fusing the previous track estimates (see [6], Section 8.6). However, at a lower rate, the deviations can not be fully corrected, thus, a certain amount of degradation in fusion accuracy is unavoidable. Also note that, in contrast to the case of T2TFwoM, T2TFwMff is more accurate than T2TFwMpf and T2TFwMnf, namely information feedback *improves* fusion accuracy in T2TFwM (as expected). While it is too involved to provide a theoretical proof of this result, simulations in different settings confirm this.

Another fusion algorithm that also uses the previous track estimates (i.e., it has memory) is the IMF [13, 25]. When operating at full rate, the IMF is algebraically equivalent to the CMF (see [6], Section 8.6) and also to the algorithms for T2TFwM presented in this section. However, at a lower rate, the IMF is not an optimal algorithm. As reported in [10], this may even lead to divergence. In contrast, the algorithms for T2TFwMnf, T2TFwMpf and T2TFwMff are optimal at any rate.

3.5. Summary of the Various T2TF Configurations

To summarize the discussion in Section 2 and Section 3:

- For T2TFwoM (which uses only the track estimates at the fusion time), information feedback will cause a degradation in fusion accuracy (see Section 2.4). This is because the local trackers use locally optimal but globally suboptimal filter gains, which leads to lower gains for the new measurements in the fused track.

TABLE V
Data Required from Local Tracker 2 for each Fusion with Fusion Interval of M

	Data Required	The Total Amounts (bits)
Exact algorithm	$W_2^e(k, l), W_2^v(k, i-1), \hat{x}_2, P_2$	$[(M+1)n_x^2 + n_x + 0.5(n_x+1)n_x]n_{acc}$
Approximate algorithm	Measurement time stamps, \hat{x}_2, P_2	$Mn_T + [n_x + 0.5(n_x+1)n_x]n_{acc}$

n_{acc} : The number of the bits for each element in the state and covariance (accuracy)
 n_x : Dimension of the state n_T : The number of bits for each measurement time stamp

Information feedback will lower the gains further, and cause more degradation in fusion accuracy.

- The algorithms for T2TFwM (which uses also the track estimates from the previous fusion) at an arbitrary rate for different information configurations (with and without information feedback) are derived for the first time. When operating at full rate, T2TFwM, with no, partial or full information feedback, is equivalent to the CMF (global optimum). When operating at a reduced rate, T2TFwMnf, T2TFwMpf and T2TFwMff all have a small degradation in fusion accuracy compared to the CMF. However, unlike the case of T2TFwoM, the degradation is smaller for T2TFwMff than for T2TFwMpf and T2TFwMnf. Namely information feedback improves fusion accuracy in T2TFwM.

4. THE APPROXIMATE IMPLEMENTATION OF THE T2TF ALGORITHMS

The T2TF algorithms at an arbitrary rate require the local weight matrices (12)–(13) at the FC. However, transmission of these matrices might not be affordable in practical distributed tracking systems due to limit in communication capacity. The idea of the approximate implementation is to approximately reconstruct the local information required by the T2TF algorithms at the FC using a minimum amount of information from the local trackers. Section 4.1 presents the approximate implementation which has significantly lower communication requirements than the original algorithms. Simulation results in Section 4.2 show that this implementation is consistent and has practically no loss in the fusion accuracy due to the approximation.

4.1. The Approximate Algorithm

Note that the local weighting matrices (12)–(13) are functions of $F(i-1), H_s(i), K_s(i), i = l+1, \dots, k$. At the FC, if estimates of the target positions (i.e., an approximate trajectory of the target) are available, using also the locations of the local sensors and the times when the local measurements are taken, the observation matrices used by the local tracker and the related track updates can be approximately reconstructed, yielding $H_s(i), K_s(i), i = l+1, \dots, k$. Note that these reconstructions do not need the actual measurements due to the

nature of the Kalman filter that the covariance updates do not depend on the actual measurements. Thus the approximate evaluation of (12)–(13) is given by⁸

$$\bar{W}_s^e(k, l) = \prod_{i=0}^{k-l-1} [I - \bar{K}_s(k-i)\bar{H}_s(k-i)]F(k-i-1) \quad (82)$$

$$\bar{W}_s^v(k, i-1) = - \left\{ \prod_{j=0}^{k-i-1} [I - \bar{K}_s(k-j)\bar{H}_s(k-j)]F(k-j-1) \right\} \cdot [I - \bar{K}_s(i)\bar{H}_s(i)] \quad (83)$$

where overbar denotes the approximate values. For T2TF, (15) can be approximated by

$$\begin{aligned} \bar{P}_{12}(k|k) &= \bar{W}_1^e(k, l)P_{12}^*(l|l)\bar{W}_2^e(k, l)' \\ &+ \sum_{i=l+1}^k \bar{W}_1^v(k, i-1)Q(i-1)\bar{W}_2^v(k, i-1)' \end{aligned} \quad (84)$$

Thus, to approximately calculate the local weights (12)–(13) required in the T2TF algorithms, the FC needs to know the following: (i) the locations of the local sensors, (ii) the time when the local measurements are obtained (considering the possibility of missed detections in practical systems) and (iii) an approximate trajectory of the target. Note that, at the FC, (iii) is available when there is a tracker running at the FC or it can be obtained by interpolation between the track estimates at the fusion times. Although extra computations are required at the FC to obtain the approximate weights at the local trackers, these computations are affordable due to the simplicity of the expressions involved. A brief analysis of the savings in communication is given below.

In distributed tracking systems with sensors at fixed locations, there is no need to transmit the locations. Assume further that tracker 1 is collocated with the FC and, thus, the approximate track trajectory can be obtained from its track estimates. Table V compares the amount of data that needs to be transmitted from local tracker 2 to the FC in the exact algorithm and in

⁸If the measurements are linear (as in [18]) there is no need for this approximation. In the problem considered later, the measurements are nonlinear (range and azimuth); consequently, this approximation will be needed.

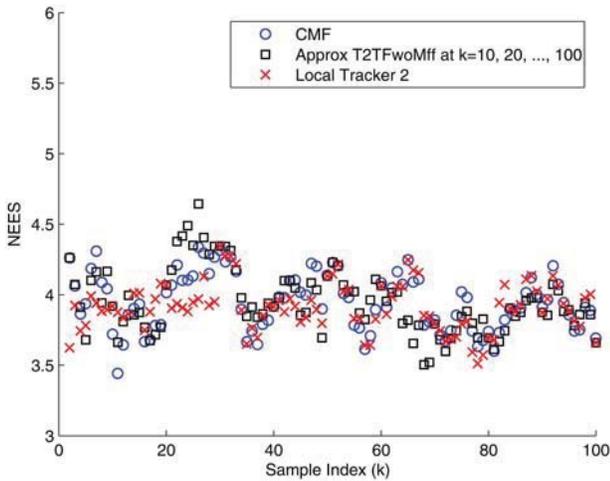


Fig. 4. Filter consistency test: NEES (T2TFwoMff).

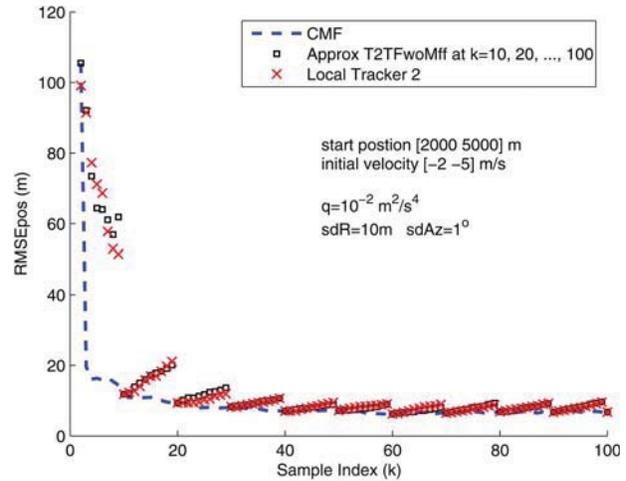


Fig. 5. RMS position errors (T2TFwoMff).

the approximate implementation. Given that transmission of measurement time stamps is much less expensive than transmission of the local weights, namely, $(M + 1)n_x^2 n_{acc} \gg Mn_T$, the savings in communication using the approximate algorithm is significant.

4.2. Simulation Results

The approximate implementation of the T2TF algorithms is tested in a 2-D multisensor tracking scenario, in which one target is tracked by two trackers. The target motion is modeled as a DWNA process [7] with process noise variance q . The target state is defined as $x = [\xi \ \dot{\xi} \ \zeta \ \dot{\zeta}]'$. Tracker 1 is located at [0 0] m and tracker 2 is at [0 10000] m. The local sensors (radars) obtain position measurements of the target in their polar coordinates, namely, range and azimuth, every $T = 1$ s. The filter used by the trackers is the converted measurement Kalman filter (CMKF) [7]. It is assumed that tracker 1 is collocated with the FC, i.e., the FC has access to the estimates of tracker 1, and tracker 2 sends its track $(\hat{x}_2(k|k), P_2(k|k))$ to the FC every 10 sampling intervals. The simulation results are obtained from 100 Monte-Carlo (MC) runs.

4.2.1. T2TFwoM with full Information Feedback (T2TFwoMff)

Figs. 4–5 show the performance of the approximate implementation in the case of T2TFwoMff. As shown in Fig. 4, the fused tracks are always consistent (the normalized estimation error squared (NEES)) are in their 95% probability region [3.46 4.69]). In Fig. 5, significant improvement in track accuracies from both trackers is observed at the fusion times $k = 10, 20, \dots, 100$. The results of the CMF are also presented as the bound of the tracking performance. The root mean square (RMS) position errors indicate that the fused track practically meets the performance bound of the CMF at the fusion times. Between the

fusion times the errors increase because each tracker is on its own. The loss of accuracy because of the information feedback in the fused track (discussed in Section 2.3) becomes insignificant due to the geometric diversity of the two tracks.⁹ The results for velocity are similar. Since the approximate implementation performs practically as well as the CMF, there is no need in this case to evaluate the exact algorithm for T2TFwoMff.

4.2.2. T2TFwoM with Partial Information Feedback (T2TFwoMpf)

Figs. 6–7 show the performance of the approximate implementation in the case of T2TFwoMpf for the same tracking scenario with the following modification: tracker 2 does not receive the fused track from the FC. Fig. 6 shows that the fused track is consistent. In terms of achieved accuracy, it also meets the performance bound of the CMF at the fusion points.

4.2.3. The Effect of Ignoring the Crosscovariance in T2TFwoMff

Figs. 8–9 show the statistics from the same tracking scenario, but the T2TFwoMff is performed assuming that the two tracks are uncorrelated. The divergence of the filter—very rapid in NEES, slower but noticeable in RMSE—demonstrates the importance of taking into account the crosscovariances. This happens because the excessive optimism (fused covariances that are too small) due to ignoring the crosscovariances.

⁹With the two sensors widely separated (10000 m apart in the simulation), and the measurements much more accurate in the range direction than the crossrange direction, the two tracks are geometrically complementary to each other in the scenario considered. In such cases, the improvements of accuracy of the fused track are very significant, and they are less affected by the crosscorrelation between the tracks.

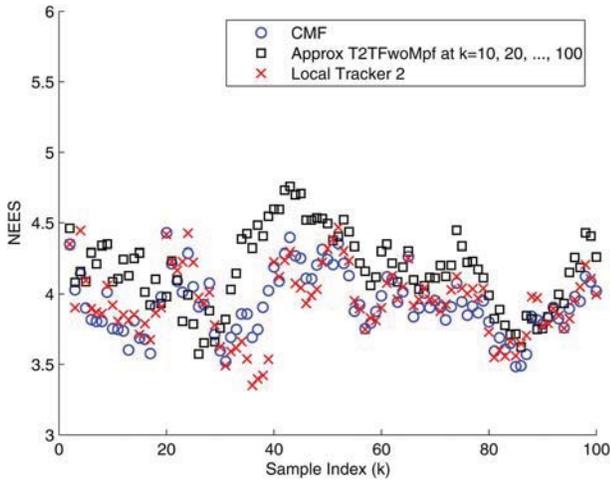


Fig. 6. Filter consistency test: NEES (T2TFwoMpf).

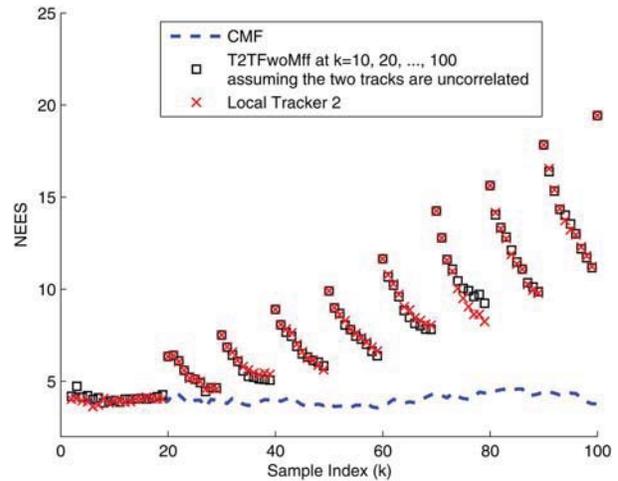


Fig. 8. Filter consistency test: NEES (T2TFwoMff). Ignoring the crosscovariances in T2TFwoMff leads to divergence.

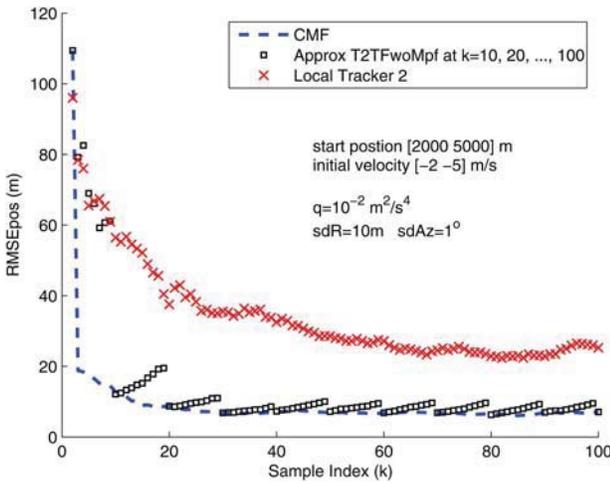


Fig. 7. RMS position errors (T2TFwoMpf).

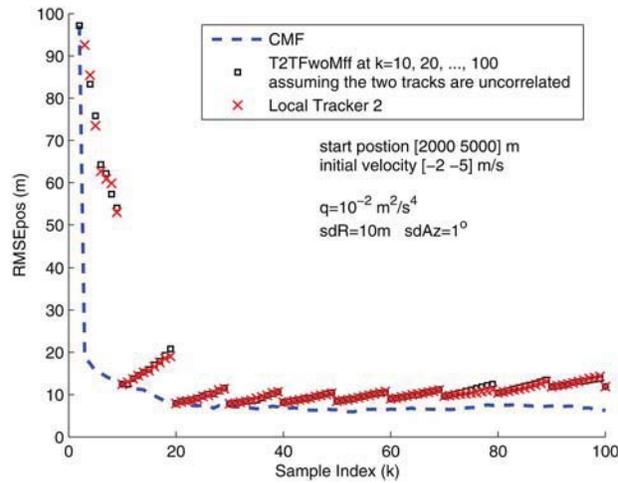


Fig. 9. RMS position errors (T2TFwoMff).

5. SLIDING WINDOW TEST FOR T2TA

The chi-square based track-to-track association (T2TA) test is investigated in this section. Note that the chi-square based T2TA test is based only on the likelihood function (LF) under H_0 (the two tracks belong to the same target). The optimal (likelihood ratio—LR) test can not be used for T2TA, since the exact LF under H_1 (the two tracks do not belong to the same target) is not available. Although a diffuse prior may be used to calculate the LF under H_1 , which leads to a LR test that is the same as the chi-square based test. The test still has virtually no information about H_1 .¹⁰ The test can be done based on a single frame of track estimates, or using track estimates at multiple times. Conventional belief is that, given the same false alarm rate, using multiple frames of data will yield higher power. Accordingly the sliding window test is proposed in Section 5.1, which is shown

¹⁰Also note that the chi-square based test uses only the Gaussian exponent (This has the disadvantage that large covariance leads to acceptance but it has low power). The assignment, however, will use the full LF, i.e., there is penalty for large covariance matrix.

to yield false alarm rates that match the theoretical values. Then we compare the power of the sliding window test to that of the single time test. Counterintuitively, it is observed that the sliding window test, which uses more data, does not necessarily have more power than the single time test. The reason for this phenomenon is discussed in Section 5.2.

5.1. The Algorithm of the Sliding Window Test for T2TA

Consider the basic T2TA test of whether two tracks originated from the same target. For the single time test at time k , the data includes the tracks $\hat{x}_1(k|k)$, $P_1(k|k)$ from tracker 1 and $\hat{x}_2(k|k)$, $P_2(k|k)$ from tracker 2, as well as their crosscovariance $P_{12}(k|k)$. Define

$$\Delta(k) = \hat{x}_1(k|k) - \hat{x}_2(k|k). \quad (85)$$

It follows that

$$P_{\Delta}(k) = P_1(k|k) + P_2(k|k) - P_{12}(k|k) - P_{12}(k|k)' \quad (86)$$

where $P_{12}(k | k)$ is calculated from (15). The test statistic is

$$T(k) = \Delta(k)' P_{\Delta}(k)^{-1} \Delta(k) \quad (87)$$

which, under H_0 (the two tracks are for the same target), is a random variable with a $\chi_{n_x}^2$ distribution (n_x is the dimension of the state). However, for data association at subsequent times, similar test statistics $T(g)$, $g > k$, are correlated with $T(k)$, thus the sum of single time test statistics (87) within a time window does not have a χ^2 distribution [4, 24].¹¹

The sliding window test based on the most recent N frames of data needs to account for the crosscovariances among data at different times. Without loss of generality, consider the T2TA that occurs at time t_m . Define

$$\Delta_N(t_m) = [\Delta(t_m) \quad \Delta(t_{m-1}) \dots \Delta(t_{m-N+1})]' \quad (88)$$

where subscript N is the window length and $\Delta(t_m), \dots, \Delta(t_{m-N+1})$ are from the N most recent track estimates received by the FC. The test statistic $T_N(t_m)$ for this sliding window of N times is

$$T_N(t_m) = \Delta_N(t_m)' \text{Cov}[\Delta_N(t_m)]^{-1} \Delta_N(t_m) \quad (89)$$

which, under H_0 , has a χ^2 distribution with Nn_x degrees of freedom.

Given $\text{Cov}[\Delta_N(t_{m-1})]$ from the previous sliding window, to obtain $\text{Cov}[\Delta_N(t_m)]$, the new diagonal term

$$\text{Cov}[\Delta(t_m)] = P_{\Delta}(t_m) \quad (90)$$

is calculated from (86). The new off-diagonal terms are

$$\begin{aligned} \text{Cov}[\Delta(t_m), \Delta(t_i)] &= \text{Cov}[\tilde{x}_1(t_m | t_m) - \tilde{x}_2(t_m | t_m), \tilde{x}_1(t_i | t_i) - \tilde{x}_2(t_i | t_i)] \\ &= \text{Cov}[\tilde{x}_1(t_m | t_m), \tilde{x}_1(t_i | t_i)] + \text{Cov}[\tilde{x}_2(t_m | t_m), \tilde{x}_2(t_i | t_i)] \\ &\quad - \text{Cov}[\tilde{x}_1(t_m | t_m), \tilde{x}_2(t_i | t_i)] - \text{Cov}[\tilde{x}_2(t_m | t_m), \tilde{x}_1(t_i | t_i)], \\ &\quad i = m - N + 1, \dots, m - 1. \end{aligned} \quad (91)$$

From (11), it follows that

$$\begin{aligned} \text{Cov}[\Delta(t_m), \Delta(t_i)] &= W_e^1(t_m, t_i) P_1(t_i | t_i) + W_e^2(t_m, t_i) P_2(t_i | t_i) \\ &\quad - W_e^1(t_m, t_i) P_{12}(t_i | t_i) - W_e^2(t_m, t_i) P_{21}(t_i | t_i) \end{aligned} \quad (92)$$

where $P_1(t_i | t_i)$, $P_2(t_i | t_i)$ are from the tracks at time t_i and $P_{12}(t_i | t_i)$ is calculated using (15).

5.2. The Sliding Window Test vs. the Single Time Test

The sliding window test using test statistics (89) and the single time test with test statistics (87) are compared in a 1-D multisensor tracking scenario as in Section 2.3. Target 1 starts at 5000 m with an initial velocity -3 m/s. Target 2 starts at 5030 m with the same initial velocity

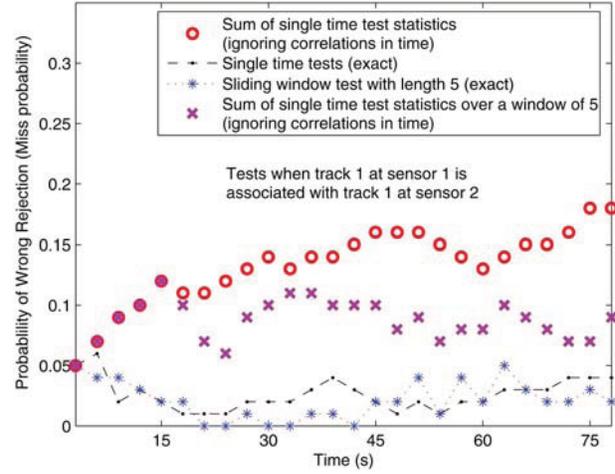


Fig. 10. Miss probability (wrong rejection) from 100 MC runs for 0.025 theoretical value (associations are done every 3 s).

(the initial target separation is 30 m).¹² Both targets have process noises with variance $q = 2 \cdot 10^{-2} \text{ m}^2/\text{s}^4$. These noises are independent across the targets, leading to their eventual separation. Two sensors, designated as 1 and 2, obtain position measurements of targets 1 and 2 every 1 s and maintain separate tracks for the targets. The T2TA tests are performed every 3 s, which is the time interval between two consecutive local track estimates used in the window test. The sliding window test uses a window of $N = 5$ times. For comparison, two tests based on the sum of the single time tests are also performed. One is the cumulative sum over all the previous single time tests; another is the sum of the single time tests within a sliding window of N as the approach proposed in [16, 24]. Both are not optimal because the correlations in time are ignored.

Fig. 10 shows the miss probability when the track of target 1 at sensor 1 is associated with the track of the same target at sensor 2. The theoretical miss probability is 2.5% (correct acceptance 97.5%). The “single time test” (based on single frame of data) and the “sliding window test” match the theoretical error probability. However, the miss probabilities of the other two tests based on the sum of the single time test statistics are significantly larger than the theoretical value. This is due to the fact that these tests ignore the correlation among the single time test statistics.

Fig. 11 shows the probability of correct rejection (power of the test) for 0.025 miss probability when the track of target 1 at sensor 1 is associated with the track of target 2 at sensor 2. Surprisingly, the sliding window test has lower power than the single time test. This counterintuitive phenomenon is further analyzed in the Section 5.3 with an illustrative example.

¹²This small separation is only for the purpose of comparing the power of different tests for T2T association. It is assumed that these closely spaced targets are resolved by the sensors.

¹¹This is contrary to the assertion in [16].

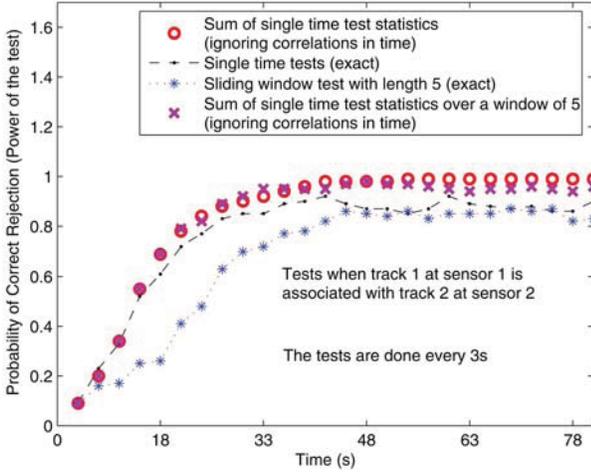


Fig. 11. Power of test for 0.025 miss probability from 100 MC runs.

5.3. The Effect of Window Length on the Power of the T2TA Test

Consider a scalar state estimation problem with two sensors/estimators. Estimator 1 has prior information $\hat{x}_1(0) \sim N(x_1(0), P_1)$, where $x_1(0)$ denotes the true state of the target corresponding to track 1 at time 0, P_1 is the variance of the estimate $\hat{x}_1(0)$. At time 1, this true state propagates to $x_1(1) = x_1(0) + v$ where $v \sim N(0, Q)$. A measurement is taken at time 1 as $z_1(1) = x_1(1) + w_1$, where $w_1 \sim N(0, R_1)$. Estimator 2 has prior information on the target corresponding to track 2, $\hat{x}_2(0) \sim N(x_2(0), P_2)$. At time 1, the state of this target evolves as $x_2(1) = x_2(0) + v$ and the measurement of sensor 2 is $z_2(1) = x_2(1) + w_2$, where $w_2 \sim N(0, R_2)$. It is assumed that the states of the two targets have the same process noise v , so the difference between their true states stays constant (they are moving in formation).¹³ When the two targets are the same, $x_1(t) - x_2(t) = 0$, $t = 0, 1$, otherwise the target separation is $|x_1(t) - x_2(t)| = d > 0$, $t = 0, 1$. It is assumed that the errors in the prior information and the measurement noises are all independent and for the sake of simplicity $P_1 = P_2 = R_1 = R_2 = \sigma^2$.

For the T2TA test based on the prior information at time 0, one has

$$\Delta(0) = \hat{x}_1(0) - \hat{x}_2(0) \quad (93)$$

$$\text{Var}[\Delta(0)] = P_1 + P_2 = 2\sigma^2 \quad (94)$$

$$T(0) = \text{Var}[\Delta(0)]^{-1} \Delta(0)^2 = \frac{1}{2\sigma^2} \Delta(0)^2 \quad (95)$$

under H_0 (the two tracks are from the same target), $E[\Delta(0)] = 0$; under H_1 (the two tracks are from two different targets), $E[\Delta(0)] = d$.

At time 1, with the measurements z_1 and z_2 , the updated estimates for the target (under H_0) or targets

¹³As later discussed, this assumption is necessary to obtain the actual theoretical performance of the test.

(under H_1) are

$$\begin{aligned} \hat{x}_1(1) &= \frac{R_1}{P_1 + Q + R_1} \hat{x}_1(0) + \frac{P_1 + Q}{P_1 + Q + R_1} z_1 \\ &= \frac{\sigma^2}{2\sigma^2 + Q} \hat{x}_1(0) + \frac{\sigma^2 + Q}{2\sigma^2 + Q} z_1 \end{aligned} \quad (96)$$

$$\begin{aligned} \hat{x}_2(1) &= \frac{R_2}{P_2 + Q + R_2} \hat{x}_2(0) + \frac{P_2 + Q}{P_2 + Q + R_2} z_2 \\ &= \frac{\sigma^2}{2\sigma^2 + Q} \hat{x}_2(0) + \frac{\sigma^2 + Q}{2\sigma^2 + Q} z_2. \end{aligned} \quad (97)$$

Thus

$$\begin{aligned} \Delta(1) &= \hat{x}_1(1) - \hat{x}_2(1) \\ &= \frac{\sigma^2}{2\sigma^2 + Q} (\hat{x}_1(0) - \hat{x}_2(0)) + \frac{\sigma^2 + Q}{2\sigma^2 + Q} (z_1 - z_2) \end{aligned} \quad (98)$$

$$\begin{aligned} \text{Var}[\Delta(1)] &= \frac{\sigma^4}{(2\sigma^2 + Q)^2} (P_1 + P_2) + \frac{(\sigma^2 + Q)^2}{(2\sigma^2 + Q)^2} (R_1 + R_2) \\ &= \frac{2Q^2\sigma^2 + 4Q\sigma^4 + 4\sigma^6}{(2\sigma^2 + Q)^2} \end{aligned} \quad (99)$$

$$\begin{aligned} T(1) &= \text{Var}[\Delta(1)]^{-1} \Delta(1)^2 \\ &= \frac{(2\sigma^2 + Q)^2}{2Q^2\sigma^2 + 4Q\sigma^4 + 4\sigma^6} \Delta(1)^2 \end{aligned} \quad (100)$$

under H_0 , $E[\Delta(1)] = 0$ and under H_1 , $E[\Delta(1)] = d$.

For the window test,

$$\Delta_2(1) = [\Delta(0) \quad \Delta(1)]' \quad (101)$$

$$\text{Cov}[\Delta_2(1)] = \begin{bmatrix} 2\sigma^2 & \frac{2\sigma^4}{2\sigma^2 + Q} \\ \frac{2\sigma^4}{2\sigma^2 + Q} & \frac{2Q^2\sigma^2 + 4Q\sigma^4 + 4\sigma^6}{(2\sigma^2 + Q)^2} \end{bmatrix} \quad (102)$$

$$T_2(1) = \Delta_2(1)' \{ \text{Cov}[\Delta_2(1)] \}^{-1} \Delta_2(1) \quad (103)$$

with $E[\Delta_2(1) | H_0] = [0 \quad 0]'$ and $E[\Delta_2(1) | H_1] = [d \quad d]'$.

Since the single time test is a special case of the window test with a window length of 1, $T(0)$ can also be denoted as $T_1(0)$ and $T(1)$ as $T_1(1)$. The tests statistics $T(0)$, $T(1)$ and $T_2(1)$, which are quadratic forms, have non-central Chi-square distributions with N degrees of freedom and noncentrality parameter λ [14]. The number of the degrees of freedom of the test statistic is the window length N and the noncentrality parameter λ is given by

$$\begin{aligned} \lambda_N(t) &= E[\Delta_N(t)]' \{ \text{Cov}[\Delta_N(t)] \}^{-1} E[\Delta_N(t)], \\ & \quad t = 0, 1 \end{aligned} \quad (104)$$

with the expectation taken conditioned on H_0 ("same target," i.e., $d = 0$) or H_1 ($d > 0$). Specifically,

$$T_N(t) \sim \chi^2(N, \lambda_N(t)), \quad t = 0, 1. \quad (105)$$

TABLE VI
Statistical Properties of the Test Statistics: Single Time Test vs. Sliding Window Test

Test statistic $\sim \chi^2(N, \lambda)$	Degrees of Freedom N	Noncentrality parameter λ	
		H_0	H_1
$T(0) = T_1(0)$	1	0	$\frac{1}{2\sigma^2}d^2$
$T(1) = T_1(1)$	1	0	$\frac{(2\sigma^2 + Q)^2}{2Q^2\sigma^2 + 4Q\sigma^4 + 4\sigma^6}d^2$
$T_2(1)$	2	0	$\frac{1}{\sigma^2}d^2$

Notice that (105) holds only when the covariance matrices of $\Delta_N(t)$ are the same under both H_0 and H_1 , which requires the targets to have the same process noise. This happens when the targets move in formation. However, in general, different targets do not necessarily have the same process noise. In such cases, the test statistic $T_N(t)$ does not follow a non-central χ^2 distribution under H_1 and the difference between the true states of the targets is nonstationary. Thus the power of the test can not be obtained theoretically.

The cumulative distribution function (cdf) of a $\chi^2(N, \lambda)$ random variable is given by [28]

$$P\{\chi^2(N, \lambda) \leq x\} = \sum_{j=0}^{\infty} e^{-\lambda/2} \frac{(\lambda/2)^j}{j!} \frac{\gamma(j + k/2, x/2)}{\Gamma(j + k/2)}. \quad (106)$$

Softwares are available for the calculation of (106).

The statistical properties of the above test statistics under H_0 and H_1 are shown in Table VI. Notice that, in this example, the noncentrality parameter of the window test $T_2(1)$ doesn't depend on the value of the process noise variance Q . This holds for this specific example but is not true in general. However, it is easy to show that the noncentrality parameter of the sliding window test is always greater than or equal to that of the single time test.

Assuming $\sigma^2 = 1$, Fig. 12 compares the noncentrality parameters for the window test ($N = 2$) and the single time test at time 1. It can be seen that, if $Q = 0$,

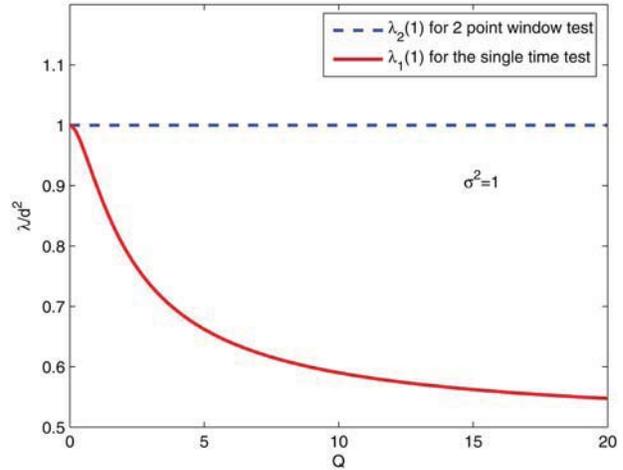


Fig. 12. The noncentrality parameters (normalized by the separation squared) vs. process noise variance.

then the noncentrality parameters of $T_2(1)$ and $T(1)$ are the same. However, $T_2(1)$ has 1 more degree of freedom than $T(1)$, thus the sliding window test $T_2(1)$ requires a higher threshold for the same miss probability of H_0 and, consequently, is less powerful than the single time test $T(1)$. As the variance of the process noise Q increases, the noncentrality parameter of $T_2(1)$ remains constant, and will be significantly larger than the noncentrality parameter of $T(1)$, which decreases with Q , as shown in Fig. 12. This compensates for the larger number of degrees of freedom of $T_2(1)$ and makes $T_2(1)$ eventually more powerful than $T(1)$.

Table VII compares the power of the tests under different process noise variances Q when $d = 3$ and $\sigma^2 = 1$. The “Threshold for rejection” and “Power of the test” are from the theoretical calculations. The “Miss probability” and “Correct rejection” are from Monte Carlo simulations and they match the theoretical values. The results show that when the process noise level is high ($Q = 6$), the window test has higher power than the single time test; however, counterintuitively, the window test has lower power than the single time test when the process noise level is low ($Q = 0.1$).

Note that the power of the test depends on (i) the number of degrees of freedom N of the chi-square

TABLE VII
Performance of the Tests ($d = 3$, $\sigma^2 = 1$ and Probability of Correct Acceptance $1 - \alpha = 0.975$). Q is the Process Noise Variance; N is the Degrees of Freedom of the Test Statistic and λ is the Noncentrality Parameter. The Miss Probability and Correct Rejection (power) are Obtained from 1000 Monte Carlo (MC) Runs.

Test Stat	Q	N	λ under H_1	Threshold (rejection of H_0 , $\alpha = 0.025$)	Theoretical power of the test	MC Miss Prob of H_0	MC Correct Rejection of H_0
$T(1)$	0.1	1	8.98	5.02	0.775	0.026	0.76
$T_2(1)$		2	9	7.38	0.678	0.030	0.69
$T(1)$	6	1	5.76	5.02	0.56	0.031	0.57
$T_2(1)$		2	9	7.38	0.678	0.029	0.66

test statistic (which determines the threshold) and (ii) the noncentrality parameter λ . Two explanations from different perspectives to this seemingly counterintuitive phenomenon are given next.

1. In this example, the power of the window test $T_2(1)$ remains the same over different process noise levels Q as a result of the constant noncentrality parameter, i.e., the window test is not sensitive to process noise levels. When the process noise level is low ($Q = 0.1$ in this example) the power of the single time test is higher than the window test's because it has almost the same noncentrality parameter as the window test but only one degree of freedom (i.e., lower threshold). However, with a high level of process noise (when $Q = 6$) the noncentrality parameter decreases and the power of the single time test drops below that of the window test.
2. At time 1, by incorporating the data from time 0, the window test has a noncentrality parameter larger than that of the single time test (for both cases when $Q = 0.1$ and $Q = 6$). However, the inclusion of the data from time 0 also increases the degrees of freedom of the test statistic (from 1 to 2), which has a negative impact on the power of the test (for the same false alarm rate, this raises the threshold to 7.38 from 5.02 for the single time test). When the crosscorrelation between the data at time 1 and time 0 is large (which happens for low process noise $Q = 0.1$), the increase in the noncentrality parameter is too small to overcome the negative effect of the increased degree of freedom. In such cases, the window test has lower power than the single time test.

The discussion above indicates that the advantage of the window is negated by the crosscorrelation in time, which is higher for low process noise.¹⁴ This suggests that, to enhance the power of the sliding window test, it is necessary to make sure that the multiple frames of data selected for the test are not strongly correlated. This can be accomplished by increasing the time difference between the selected frames.

To confirm this guideline, Fig. 13 shows the power of the tests under a theoretical false alarm rate of $\alpha = 0.025$ in the same simulation scenario as in Section 5.2 except that the tests for T2TA are done every 15 s (e.g., the time interval of two consecutive track estimates used in the window test) as opposed to 3 s and the length of the sliding window is set to 4. It is shown that, in this case, the sliding window test has more power than the single time test.

The phenomenon that using more data may lead to lower power in the classical chi-square test seems counterintuitive. However, it can be better understood given the fact that the test uses only the likelihood function (LF) under H_0 and the LF under H_1 is unknown. Thus

¹⁴This is because with lower process noise the “memory” of the filter is “longer.”

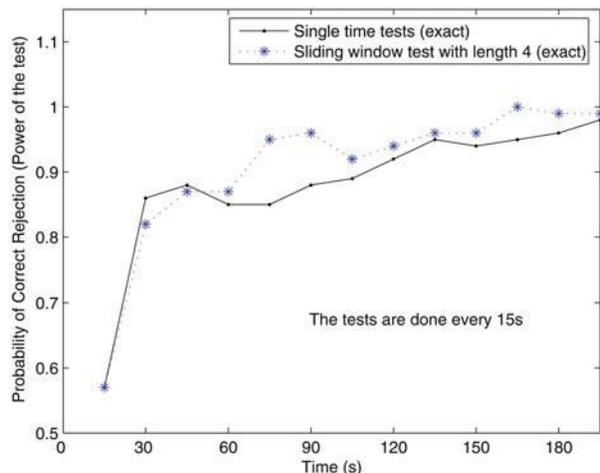


Fig. 13. Power of the test: sliding window test vs. single frame test with a increased testing interval of 15 s.

it is unclear how the available data should be combined to better differentiate the two hypothesis. The loss in power of the chi-square based window T2TA test is the result of the lack of information under H_1 . A likelihood ratio (LR) test would be more powerful but it requires knowledge of the target separation, i.e., it is not practical.

6. CONCLUSIONS

In this paper the optimal T2TF algorithms at an arbitrary rate are investigated for various information configurations. First algorithms for T2TF without memory (T2TFwoM: fuser uses only the local track estimates at the fusion time) are presented for three information configurations, namely, T2TFwoM with no, partial and full information feedback. It is shown that, for T2TFwoM, information feedback is detrimental to fusion accuracy. Then algorithms for T2TF with memory (T2TFwM: fuser uses also the fused and local track estimates from the previous fusion) at an arbitrary rate are derived for information configurations with no, partial and full information feedback. It is shown that, at full rate, T2TFwM, with or without information feedback, is equivalent to the centralized measurement fusion (CMF, which is the global optimum). However, when operating at a lower rate, a certain amount of loss in fusion accuracy (compared to the CMF) is unavoidable. In contrast to the case of T2TFwoM, it is shown that information feedback improves the fusion accuracy of T2TFwM. And, unlike the information matrix fusion (IMF) which is optimal (same as CMF) only at full rate, the algorithms for T2TFwM are optimal at any rate.

An approximate implementation of the T2TF algorithms is also proposed based on the reconstruction of local information at the fusion center (FC). For nonlinear distributed tracking systems, it has much lower communication requirements and practically no loss in fusion accuracy due to the approximation. Simulation results show that it is consistent and, for the sensors-

target geometry considered, it meets the performance bound of the centralized measurement fusion at the fusion points.

The hypothesis test for T2TA is also studied in this paper. The sliding window test for T2TA is presented. It uses track estimates in a time window and yields false alarm rates that match the theoretical values. The sliding window test was compared with the single time test and the results show that, counterintuitively, the sliding window test may have lower power than the single time test. This is because using more data also increases the degrees of freedom of the test statistics which has a negative impact on the power of the test. When the multiple frames of data selected for T2TA are strongly correlated, which happens for motion with low process noise, the increase in the noncentrality parameter is too small to overcome the negative effect of the increased degree of freedom. In such cases, the sliding window test may be counterproductive and has lower power than the single time test. In practice, this should be avoided by, e.g., increasing the time difference between the selected data frames.

REFERENCES

- [1] Y. Bar-Shalom
On the track-to-track correlation problem.
IEEE Transactions on Automatic Control, **26**, 2 (Apr. 1981), 571–572.
- [2] Y. Bar-Shalom
Update with out-of-sequence measurements in tracking: Exact solution.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 3 (July 2002), 769–778.
- [3] Y. Bar-Shalom
On hierarchical tracking for the real world.
IEEE Transactions on Aerospace and Electronic Systems, **42**, 3 (July 2006), 846–850.
- [4] Y. Bar-Shalom
On the sequential track correlation algorithm in a multisensor data fusion system.
IEEE Transactions on Aerospace and Electronic Systems, **43**, 1 (Jan. 2008), 396.
- [5] Y. Bar-Shalom and L. Campo
The effect of the common process noise on the two-sensor fused-track covariance.
IEEE Transactions on Aerospace and Electronic Systems, **22**, 6 (Nov. 1986), 803–804.
- [6] Y. Bar-Shalom and X. R. Li
Multitarget-Multisensor Tracking: Principles and Techniques.
YBS Publishing, 1995.
- [7] Y. Bar-Shalom, X. R. Li and T. Kirubarajan
Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction.
Wiley, 2001.
- [8] K. C. Chang, R. K. Saha, Y. Bar-Shalom and M. Alford
Performance evaluation of multisensor track-to-track fusion. In *Proceedings of 1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Dec. 1996, 627–632.
- [9] K. C. Chang, R. K. Saha, Y. Bar-Shalom
On optimal track-to-track fusion.
IEEE Transactions on Aerospace and Electronic Systems, **33**, 4 (Oct. 1997), 1271–1276.
- [10] K. C. Chang, Z. Tian, and R. Saha
Performance evaluation of track fusion with information matrix filter.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 2 (Apr. 2002), 455–466.
- [11] H. Chen, T. Kirubarajan and Y. Bar-Shalom
Performance limits of track-to-track fusion vs. centralized estimation.
IEEE Transactions on Aerospace and Electronic Systems, **39**, 2 (Apr. 2003), 386–400.
- [12] H. Chen and Y. Bar-Shalom
Track fusion with legacy track sources.
In *Proceedings of the 9th International Conference on Information Fusion*, Florence, Italy, July 2006.
- [13] C. Y. Chong, S. Mori, and K. C. Chang
Distributed multitarget multisensor tracking.
In Y. Bar-Shalom (Ed.), *Multitarget-Multisensor Tracking: Advanced Applications*, MA: Artech House, 1990, ch. 8.
- [14] D. K. Dey, N. Ravishanker
A First Course in Linear Model Theory.
Chapman & Hall/CRC, 2001.
- [15] O. E. Drummond
A Hybrid sensor fusion algorithm architecture and tracklets.
In *Proceedings of 1997 SPIE Conference on Signal and Data Processing of small Targets*, vol. 3163, July 1997, 512–524.
- [16] Y. He and J. W. Zhang
New track correlation algorithms in a multisensor data fusion system.
IEEE Transactions on Aerospace and Electronic Systems, **42**, 4 (Oct. 2006), 1359–1371.
- [17] X. R. Li and J. Wang
Unified optimal linear estimation fusion—Part II: Discussions and examples.
In *Proceedings of the Third International Conference on Information Fusion*, Paris, France, July 2000, MoC2.18–MoC2.25.
- [18] X. R. Li, Y. M. Zhu, J. Wang and C. Z. Han
Unified optimal linear estimation fusion—Part I: Unified model and fusion rules.
IEEE Transactions on Information Theory, **49**, 9 (Sept. 2003), 2192–2207.
- [19] X. R. Li and P. Zhang
Optimal linear estimation fusion—Part III: Crosscorrelation of local estimation errors.
In *Proceedings of the Fourth International Conference on Information Fusion*, Montreal, QC, Canada, Aug. 2001, WeB1.11–WeB1.18.
- [20] X. R. Li and K. S. Zhang
Optimal linear estimation fusion—Part IV: Optimality and efficiency of distributed fusion.
In *Proceedings of the Fourth International Conference on Information Fusion*, Montreal, QC, Canada, Aug. 2001, WeB1.19–WeB1.26.
- [21] X. R. Li, K. S. Zhang, J. Zhao, and Y. M. Zhu
Optimal linear estimation fusion—Part V: Relationships.
In *Proceedings of the Fifth International Conference on Information Fusion*, Annapolis, MD, July 2002, 497–504.
- [22] X. R. Li
Optimal linear estimation fusion—Part VII: Dynamic systems.
In *Proceedings of the Sixth International Conference on Information Fusion*, Cairns, Australia, July 2003, 445–462.
- [23] S. Mori, W. H. Barker, C. Y. Chong, and K. C. Chang
Track association and track fusion with nondeterministic target dynamics.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 2 (Apr. 2002), 659–668.

- [24] B. L. Scala and A. Farina
Effects of cross-covariance and resolution on track association.
In *Proceedings of the Third International Conference on Information Fusion*, vol. 2, Paris, France, July 2000, WED1/10–WED1/16.
- [25] J. L. Speyer
Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem.
IEEE Transactions on Automatic Control, **24**, 2 (Apr. 1979), 54–57.
- [26] X. Tian and Y. Bar-Shalom
The exact algorithm for asynchronous track-to-track fusion.
In *Proceedings of SPIE Conference on Signal and Data Processing of small Targets*, Orlando, FL, Apr. 2010.
- [27] K. S. Zhang, X. R. Li, P. Zhang, and H. F. Li
Optimal linear estimation fusion—Part VI: Sensor data compression.
In *Proceedings of the Sixth International Conference on Information Fusion*, Cairns, Australia, July 2003, 221–228.
- [28] Chi-Square Distribution
http://en.wikipedia.org/wiki/Noncentral_chi-square_distribution.



Xin Tian was born in 1980. He received the B.E. degree in 2002 and M.E. degree in 2005, both from the Department of Information and Communication Engineering, Xi'an Jiaotong University, China. He is now a Ph.D. candidate at the Department of Electrical and Computer Engineering, University of Connecticut.

His research interests include statistical signal processing, tracking and information fusion.

Yaakov Bar-Shalom was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion, Israel Institute of Technology, in 1963 and 1967 and the Ph.D. degree from Princeton University in 1970, all in electrical engineering.

From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, CA. Currently he is Board of Trustees Distinguished Professor in the Dept. of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering at the University of Connecticut. He is also Director of the ESP (Estimation and Signal Processing) Lab.

His current research interests are in estimation theory and target tracking and has published over 370 papers and book chapters in these areas and in stochastic adaptive control. He coauthored the monograph *Tracking and Data Association* (Academic Press, 1988), the graduate texts *Estimation and Tracking: Principles, Techniques and Software* (Artech House, 1993), *Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction* (Wiley, 2001), the advanced graduate text *Multitarget-Multisensor Tracking: Principles and Techniques* (YBS Publishing, 1995), and edited the books *Multitarget-Multisensor Tracking: Applications and Advances* (Artech House, Vol. I, 1990; Vol. II, 1992; Vol. III, 2000).

He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multitarget tracking." He has been consulting to numerous companies and government agencies, and originated the series of Multitarget-Multisensor Tracking short courses offered via UCLA Extension, at Government Laboratories, private companies and overseas.

During 1976 and 1977 he served as Associate Editor of the IEEE Transactions on Automatic Control and from 1978 to 1981 as Associate Editor of Automatica. He was Program Chairman of the 1982 American Control Conference, General Chairman of the 1985 ACC, and Co-Chairman of the 1989 IEEE International Conference on Control and Applications. During 1983–87 he served as Chairman of the Conference Activities Board of the IEEE Control Systems Society and during 1987–89 was a member of the Board of Governors of the IEEE CSS. He was a member of the Board of Directors of the International Society of Information Fusion (1999–2004) and served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002 and Vice President for Publications in 2004–08.

In 1987 he received the IEEE CSS Distinguished Member Award. Since 1995 he is a Distinguished Lecturer of the IEEE AESS and has given numerous keynote addresses at major national and international conferences. He is corecipient of the M. Barry Carlton Award for the best paper in the IEEE Transactions on Aerospace and Electronic Systems in 1995 and 2000 and the 1998 University of Connecticut AAUP Excellence Award for Research. In 2002 he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. He is the recipient of the 2008 IEEE Dennis J. Picard Medal for Radar Technologies and Applications.



INTERNATIONAL SOCIETY OF INFORMATION FUSION

ISIF Website: <http://www.isif.org>

2009 BOARD OF DIRECTORS*

2007–2009	2008–2010	2009–2011
Stefano Coraluppi	Steve Andler	Éloi Bossé
Jean Dezert	Subhash Challa	Uwe Hanebeck
Wolfgang Koch	Mahendra Mallick	Roy Streit

*Board of Directors are elected by the members of ISIF for a three year term.

PAST PRESIDENTS

Darko Musicki, 2008	Chee Chong, 2004	Yaakov Bar-Shalom, 2000
Erik Blasch, 2007	Xiao-Rong Li, 2003	Jim Llinas, 1999
Pierre Valin, 2006	Yaakov Bar-Shalom, 2002	Jim Llinas, 1998
W. Dale Blair, 2005	Pramod Varshney, 2001	

SOCIETY VISION

The International Society of Information Fusion (ISIF) is the premier professional society and global information resource for multidisciplinary approaches for theoretical and applied information fusion technologies.

SOCIETY MISSION

Advocate

To advance the profession of fusion technologies, propose approaches for solving real-world problems, recognize emerging technologies, and foster the transfer of information.

Serve

To serve its members and engineering, business, and scientific communities by providing high-quality information, educational products, and services.

Communicate

To create international communication forums and hold international conferences in countries that provide for interaction of members of fusion communities with each other, with those in other disciplines, and with those in industry and academia.

Educate

To promote undergraduate and graduate education related to information fusion technologies at universities around the world. Sponsor educational courses and tutorials at conferences.

Integrate

Integrate ideas from various approaches for information fusion, and look for common threads and themes—look for overall principles, rather than a multitude of point solutions. Serve as the central focus for coordinating the activities of world-wide information fusion related societies or organizations. Serve as a professional liaison to industry, academia, and government.

Disseminate

To propagate the ideas for integrated approaches to information fusion so that others can build on them in both industry and academia.

Call for Papers

The Journal of Advances in Information Fusion (JAIF) seeks original contributions in the technical areas of research related to information fusion. Authors of papers in one of the technical areas listed on the inside cover of JAIF are encouraged to submit their papers for peer review at <http://jaif.msubmit.net>.

Call for Reviewers

The success of JAIF and its value to the research community is strongly dependent on the quality of its peer review process. Researchers in the technical areas related to information fusion are encouraged to register as a reviewer for JAIF at <http://jaif.msubmit.net>. Potential reviewers should notify via email the appropriate editors of their offer to serve as a reviewer.