

Journal of Advances in Information Fusion

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

Regular Papers

	Page
Fully Decentralized Estimation Using Square-Root Decompositions	3
<i>Susanne Radtke, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany</i>	
<i>Benjamin Noack, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany</i>	
<i>Uwe D. Hanebeck, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany</i>	
Analysis of Costs for the GNP Problem	17
<i>Mark Levedahl, Raytheon Technologies, Waltham, MA, USA</i>	
<i>John D. Glass, Raytheon Technologies, Waltham, MA, USA</i>	
A Constrained POMDP Formulation and Algorithmic Solution for Radar Resource Management in Multi-Target Tracking	31
<i>Max Ian Schöpe, Delft University of Technology, CD Delft, The Netherlands</i>	
<i>Hans Driessen, Delft University of Technology, CD Delft, The Netherlands</i>	
<i>Alexander G. Yarovoy, Delft University of Technology, CD Delft, The Netherlands</i>	
Improvement of Proportional Conflict Redistribution Rules of Combination of Basic Belief Assignments	48
<i>Théo Dezert, Gustave Eiffel University, Bouguenais, France</i>	
<i>Jean Dezert, The French Aerospace Lab (ONERA/DTIS), Palaiseau, France</i>	
<i>Florentin Smarandache, University of New Mexico, Gallup, NM, USA</i>	

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: June 2021

Editor-In-Chief	Stefano Coraluppi	Systems & Technology Research, USA; +1 781-305-4055; stefano.coraluppi@ieee.org
Associate	David Crouse	4555 Overlook Ave., SW. Washington, D.C., 20375; +1 (202) 404-1859; david.crouse@nrl.navy.mil
Administrative Editor	David W. Krout	University of Washington, USA; +1 206-616-2589; dkrout@apl.washington.edu

EDITORS FOR TECHNICAL AREAS

Tracking	Paolo Braca	NATO Science & Technology Organization, Centre for Maritime Research and Experimentation, Italy; +39 0187 527 461; paolo.braca@cmre.nato.int
Associate	Florian Meyer	University of California at San Diego, USA, +1 858-246-5016; flmeyer@ucsd.edu
Detection	Ruixin Niu	Virginia Commonwealth University, Richmond, Virginia, USA; +1 804-828-0030; rniu@vcu.edu
Fusion Applications	Ramona Georgescu	United Technologies Research Center, East Hartford, Connecticut, USA; 860-610-7890; georgera@utrc.utc.com
Image Fusion	Ting Yuan	Mercedes Benz R&D North America, USA; +1 669-224-0443; dr.ting.yuan@ieee.org
High-Level Fusion	Lauro Snidaro	Università degli Studi di Udine, Udine, Italy; +39 0432 558444; lauro.snidaro@uniud.it
Fusion Architectures and Management Issues	Marcus Baum	Karlsruhe Institute of Technology (KIT), Germany; +49-721-608-46797; marcus.baum@kit.edu
Classification, Learning, Bayesian and Other Reasoning Methods	Nageswara S. V. Rao	Oak Ridge National Laboratory, USA; +1 865-574-7517;
Associate	Jean Dezert	ONERA, Palaiseau, 91120, France; +33 180386564; jean.dezert@onera.fr
	Anne-Laure Joussetme	CMRE, Italy

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

Paulo Costa, <i>President</i>	Lance Kaplan, <i>Vice President Conferences</i>
Simon Maskell, <i>President-elect</i>	Anne-Laure Joussetme, <i>Vice President Membership</i>
Simon Maskell, <i>Secretary</i>	Darin Dunham, <i>Vice President Working Groups</i>
Chee Chong, <i>Treasurer</i>	Stefano Coraluppi, <i>JAIF EIC</i>
Dale Blair, <i>Vice President Publications</i>	Roy Streit, <i>Perspectives EIC</i>
David W. Krout, <i>Vice President Communications</i>	

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© 2021 ISIF, Inc.

Fully Decentralized Estimation Using Square-Root Decompositions

SUSANNE RADTKE
BENJAMIN NOACK
UWE D. HANEBECK

Networks consisting of several spatially distributed sensor nodes are useful in many applications. While distributed information processing can be more robust and flexible than centralized filtering, it requires careful consideration of dependencies between local state estimates. This paper proposes an algorithm to keep track of dependencies in decentralized systems where no dedicated fusion center is present. Specifically, it addresses double-counting of measurement information due to intermediate fusion results and correlations due to common process noise and common prior information. To limit the necessary amount of data, this paper introduces a method to partially bound correlations, leading to a more conservative fusion result than the optimal reconstruction while reducing the necessary amount of data. Simulation studies compare the performance and convergence rate of the proposed algorithm to other state-of-the-art methods.

Manuscript received August 28, 2020; revised January 15, 2021; released for publication June 30, 2021.

Refereeing of this contribution was handled by Chee-Yee Chong. This work was supported by the German Research Foundation (DFG) under grant HA 3789/14-1. The authors are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS) and with the Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany (e-mail: susanne.radtke@kit.edu; benjamin.noack@ieee.org; uwe.hanebeck@ieee.org).

1557-6418/21/\$17.00 © 2021 JAIF

COMMENT: RELATION TO PRIOR VERSIONS OF THIS PAPER

This paper is an extended version of [1], which won the Best Paper Award in the general category during the 23rd Conference on Information Fusion. Sections II, III, and IV have been improved to provide more clarity. Furthermore, Section V.B has been updated with an improved implementation of the previously used consensus algorithms, and the resulting implications are discussed.

I. INTRODUCTION

Considered problem: Sensor networks consist of several spatially distributed sensor nodes that can cooperatively perform a variety of different tasks [2], e.g., tracking a moving target using a network of cameras. In this paper, we consider the problem of fusing several state estimates in discrete-time linear Gaussian systems with multiple completely synchronized sensors with linear Gaussian observations. While centralized processing of measurements can be done optimally, network topology and communication bandwidth often forbid processing measurements in a central processing unit since nodes can only communicate with their closest neighbors. Distributed estimation allows the processing of measurements in a local processing unit. This local information is then communicated and fused with information from neighboring sensor nodes. It has been shown that the distributed processing of sensor data can be more robust, flexible, and scalable [3]. However, it introduces dependencies that need to be addressed carefully to ensure consistent fusion results.

State-of-the-art: Within the past 40 years, many algorithms [4] have been proposed to address the problems arising in distributed estimation. This includes using the information form of the Kalman filter [5]–[7] or formulating an optimally distributed Kalman filter [8]–[10]. Other approaches propose to use local Kalman filters and fuse their respective state estimates. Several publications address the correlations due to common process noise and common prior information [11]–[14]. When neglecting dependencies [15], fused estimates tend to become inconsistent as the uncertainty is underestimated. Covariance intersection [16]–[18] aims to find a conservative fusion rule to always ensure consistent results. As these are often too conservative, other approaches try to find closer bounds, e.g., inverse covariance intersection [19], [20]. Specifically for different network topologies, other algorithms such as the channel filter (ChF) [3], the information graph approach [21], or the information matrix fusion [22], [23] were proposed.

Another class of algorithms aims to converge to a global estimate by iteratively exchanging information between neighboring nodes. Prominent representatives include consensus on measurements [24],

consensus on information [25], [26], or hybrid approaches [27], [28]. Consensus methods can be regarded as suboptimal fusion rules [29] where the averaging of the information does not represent the actual information in the network and also does not consider redundant information systematically. For simpler network topologies, several approaches trying to reconstruct the cross-covariance matrix between state estimates using ensembles, e.g., the common past invariant Ensemble Kalman filter (CPI-EnKF) [30], or using samples [31]–[33] have been proposed. Furthermore, a reconstruction of cross-covariance matrices using square-root decompositions was proposed by [34] and [35]. The reconstruction of cross-covariances has advantageous properties as it allows optimal fusion with consistent fusion results that are generally more accurate and do not over- or underestimate the uncertainty. Yet, it requires the communication of additional information between sensor nodes, leading to a trade-off between optimality and network capacity.

Contribution: The square-root decomposition as initially proposed in [35] considers fusion in network topologies with only one dedicated fusion center. In this paper, we apply the decompositions to decentralized estimation tasks, where each node may sporadically serve as a fusion center. Nodes can exchange their estimates and fuse their local estimates with the received information. For this purpose, each node must keep track of correlations during its local processing steps. Not only common process noise needs to be encoded in the square-root decompositions, but also double-counting of information poses a problem in decentralized network topologies and needs to be tracked. Due to the storage requirements and communication load associated with the square-root decompositions, the nodes can reach a compromise between fusion quality and resource demands by introducing partial bounds on the correlations.

Outline: This paper is structured as follows. In Section II, we first discuss the problem of fusing several state estimates with correlated estimation errors. In Section III, we revisit the previously proposed square-root decomposition method [35] to reconstruct the cross-covariance matrix between estimates in centralized sensor networks with only one dedicated fusion center. Decentralized network topologies in the absence of a dedicated fusion center are studied in Section IV. The evaluation in Section V studies different scenarios and also provides a comparison with consensus methods. Section VII concludes this paper.

II. PROBLEM FORMULATION

We consider a discrete-time linear time-variant stochastic dynamic system with time index k and state transition matrix \mathbf{A}_k , state vector $\mathbf{x}_k \in \mathbb{R}^N$ of state dimension N , and zero-mean white Gaussian system noise

\mathbf{w}_k with noise dimension $W = N$ and covariance matrix \mathbf{Q}_k , i.e.,

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{w}_k, \text{ with } \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k). \quad (1)$$

The system is observed by a network of N_s sensor nodes. The processing and sensing times of the sensor nodes are synchronized. Each individual node i receives measurements using the observation model \mathbf{C}^i and covariance \mathbf{R}_k^i according to

$$\mathbf{z}_k^i = \mathbf{C}^i \mathbf{x}_k + \mathbf{v}_k^i, \text{ with } \mathbf{v}_k^i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k^i). \quad (2)$$

Furthermore, we assume that the measurement noise and the process noise are mutually independent. Each node i computes a state estimate $\hat{\mathbf{x}}_{k|k}^i$ with error covariance matrix $\mathbf{P}_{k|k}^i = E[(\hat{\mathbf{x}}_{k|k}^i - \mathbf{x}_k)(\hat{\mathbf{x}}_{k|k}^i - \mathbf{x}_k)^T]$.

A. Fusion of Estimates

Without loss of generality, we confine ourselves to the fusion of two estimates as multiple estimates can be fused sequentially. In the following discussions, we also omit the time index k for the sake of clarity.

The fusion of two state estimates $\hat{\mathbf{x}}^i$ and $\hat{\mathbf{x}}^j$ can take place at an arbitrary time step k and is a linear combination with the fusion gains \mathbf{F}^i and \mathbf{F}^j . Depending on the chosen fusion algorithm, the gains can be determined according to the Bar-Shalom/Campo formulas but can also be fixed weighting matrices. The fused estimate becomes

$$\hat{\mathbf{x}}^f = \mathbf{F}^i \hat{\mathbf{x}}^i + \mathbf{F}^j \hat{\mathbf{x}}^j, \quad (3)$$

with $\mathbf{F}^i + \mathbf{F}^j = \mathbf{I}$ and the corresponding error covariance matrix

$$\begin{aligned} \mathbf{P}^f &= \mathbf{F}^i \mathbf{P}^i (\mathbf{F}^i)^T + \mathbf{F}^i \mathbf{P}^{i,j} (\mathbf{F}^j)^T + \mathbf{F}^j \mathbf{P}^{j,i} (\mathbf{F}^i)^T + \mathbf{F}^j \mathbf{P}^j (\mathbf{F}^j)^T \\ &= [\mathbf{F}^i \ \mathbf{F}^j] \mathbf{J} [\mathbf{F}^i \ \mathbf{F}^j]^T. \end{aligned} \quad (4)$$

The joint error covariance matrix is

$$\mathbf{J} = \begin{bmatrix} \mathbf{P}^i & \mathbf{P}^{i,j} \\ \mathbf{P}^{j,i} & \mathbf{P}^j \end{bmatrix},$$

where $\mathbf{P}^{i,j} = E[(\hat{\mathbf{x}}^i - \mathbf{x})(\hat{\mathbf{x}}^j - \mathbf{x})^T] = (\mathbf{P}^{j,i})^T$ denote the cross-covariances and characterize the correlated estimation errors between the state estimates. Typically, the fusion gains \mathbf{F}^i and \mathbf{F}^j are computed to minimize the estimation error $E[(\hat{\mathbf{x}}^f - \mathbf{x})^T(\hat{\mathbf{x}}^f - \mathbf{x})]$. In this case, we refer to $\hat{\mathbf{x}}^f$ as the optimal fusion result. As discussed, e.g., in [37], the optimal fusion result can also be represented as a weighted least-squares (WLS) estimate

$$\hat{\mathbf{x}}^{\text{WLS}} = \arg \min_{\hat{\mathbf{x}}} [\hat{\mathbf{m}} - \mathbf{H} \hat{\mathbf{x}}]^T \mathbf{J}^{-1} [\hat{\mathbf{m}} - \mathbf{H} \hat{\mathbf{x}}], \quad (5)$$

with $\hat{\mathbf{m}} = [\hat{\mathbf{x}}^i \ \hat{\mathbf{x}}^j]^T$ and the matrix $\mathbf{H} = [\mathbf{I} \ \mathbf{I}]^T$, which determines how the local states map into the global state vector. The solution to formula (5) is a gain matrix according to

$$\mathbf{K} = [\mathbf{F}^i \ \mathbf{F}^j] = (\mathbf{H}^T \mathbf{J}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{J}^{-1}.$$

For two sensor nodes, the fusion gains can be calculated according to the Bar-Shalom/Campo formulas [11] by

$$\mathbf{F}^j = (\mathbf{P}^i - \mathbf{P}^{i,j})(\mathbf{P}^i + \mathbf{P}^j - \mathbf{P}^{i,j} - \mathbf{P}^{i,i})^{-1}$$

and $\mathbf{F}^i = \mathbf{I} - \mathbf{F}^j$.

(6)

Then, the fusion rule can be written as

$$\mathbf{P}^f = (\mathbf{H}^T \mathbf{J}^{-1} \mathbf{H})^{-1},$$
(7)

$$\hat{\mathbf{x}}^f = \mathbf{K} \hat{\mathbf{m}} = \mathbf{P}^f \mathbf{H}^T \mathbf{J}^{-1} \hat{\mathbf{m}}.$$
(8)

Furthermore, from (5), it follows that the fusion result is unbiased. This formula can only be solved optimally if the joint covariance matrix \mathbf{J} is available. The entries on the main diagonal are the covariances of the local filters and thus known. The entries on the off-diagonals, on the other hand, are caused by the dependent information shared between the individual sensor nodes, and they are usually hard to keep track of.

There are several sources of correlated estimation errors in distributed state estimation problems [13], namely

- 1) common prior information,
- 2) common process noise, and
- 3) common measurement information.

Common prior information occurs when the local KFs are initialized with the same information, e.g., the same prior state estimate and the same prior covariance matrix. But even with independent initialization of local filters, every sensor node is affected by the same process noise, which leads to correlated estimation errors between state estimates. The local KFs assume conditional independence of measurements, which are then incorporated into the local state estimates. Due to the spread of information throughout the network and further processing, measurement information can be incorporated into several state estimates. This double-counting of sensor data causes additional correlations. Only proper treatment of these correlations allows correct and consistent fusion results.

Optimal fusion is an essential aspect of distributed estimation, and several authors discussed the optimality of the fusion of state estimates, e.g., [36] and [37]. However, the fusion of state estimates is not equal to the minimum mean squared error (MMSE) sense in which a central KF can utilize measurements. Therefore, we want to distinguish between a central KF and the optimal centralized fusion in this paper.

B. Correlations Due to Common Process Noise and Common Prior Information

In systems with a central fusion node [see Fig. 1(a)], state estimates are correlated due to common process noise and common prior information. When all processing steps are known, the cross-covariances between state estimates can be calculated recursively [11]. During the time update, the process noise is incorporated and the cross-covariance matrix is updated, leading to the recursive formula

$$\begin{aligned} \mathbf{P}_{k|k-1}^{i,j} &= E[(\hat{\mathbf{x}}_{k|k-1}^i - \mathbf{x}_k)(\hat{\mathbf{x}}_{k|k-1}^j - \mathbf{x}_k)^T] \\ &= E\left[(\mathbf{A}_k \hat{\mathbf{x}}_{k-1|k-1}^i - (\mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{w}_k)) \right. \\ &\quad \left. \times (\mathbf{A}_k \hat{\mathbf{x}}_{k-1|k-1}^j - (\mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{w}_k))^T\right] \\ &= \mathbf{A}_k E\left[(\hat{\mathbf{x}}_{k-1|k-1}^i - \mathbf{x}_{k-1})(\hat{\mathbf{x}}_{k-1|k-1}^j - \mathbf{x}_{k-1})^T\right] \mathbf{A}_k^T \\ &\quad + E[\mathbf{w}_k(\mathbf{w}_k)^T] \\ &= \mathbf{A}_k \mathbf{P}_{k-1|k-1}^{i,j} \mathbf{A}_k^T + \mathbf{Q}_k, \end{aligned}$$
(9)

where $\mathbf{P}_{k-1|k-1}^{i,j}$ for time step $k = 1$ is the common prior covariance $\mathbf{P}_{0|0}$. During the measurement update, the cross-covariance is updated using the KF gain \mathbf{K}_k^i by

$$\begin{aligned} \mathbf{P}_{k|k}^{i,j} &= E[(\hat{\mathbf{x}}_{k|k}^i - \mathbf{x}_k)(\hat{\mathbf{x}}_{k|k}^j - \mathbf{x}_k)^T] \\ &= E\left[(\hat{\mathbf{x}}_{k|k-1}^i + \mathbf{K}_k^i \mathbf{z}_k^i - \mathbf{x}_k)(\hat{\mathbf{x}}_{k|k-1}^j + \mathbf{K}_k^j \mathbf{z}_k^j - \mathbf{x}_k)^T\right] \\ &= E\left[(\hat{\mathbf{x}}_{k|k-1}^i + \mathbf{K}_k^i(\mathbf{v}_k^i - \mathbf{C}_k^i \hat{\mathbf{x}}_{k|k-1}^i) - \mathbf{x}_k) \right. \\ &\quad \left. \times (\hat{\mathbf{x}}_{k|k-1}^j + \mathbf{K}_k^j(\mathbf{v}_k^j - \mathbf{C}_k^j \hat{\mathbf{x}}_{k|k-1}^j) - \mathbf{x}_k)^T\right] \end{aligned}$$

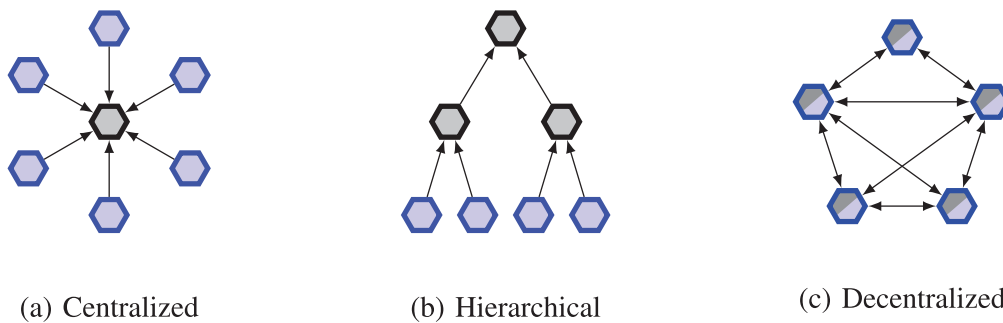


Fig. 1. Different network topologies with sensor nodes (blue), nodes only dedicated to fusion (gray), and sensor nodes with fusion capabilities (blue and gray).

$$\begin{aligned}
&= (\mathbf{I} - \mathbf{K}_k^i \mathbf{C}_k^i) E \left[(\hat{\mathbf{x}}_{k|k-1}^i - \mathbf{x}_k) (\hat{\mathbf{x}}_{k|k-1}^j - \mathbf{x}_k)^\top \right] \\
&\quad \times (\mathbf{I} - \mathbf{K}_k^j \mathbf{C}_k^j)^\top + E \left[\mathbf{v}_k^i (\mathbf{v}_k^j)^\top \right] \\
&= \mathbf{L}_k^i \mathbf{P}_{k|k-1}^{i,j} (\mathbf{L}_k^j)^\top, \tag{10}
\end{aligned}$$

where $\mathbf{L}_k^i = \mathbf{I} - \mathbf{K}_k^i \mathbf{C}_k^i$ and $E[\mathbf{v}_k^i (\mathbf{v}_k^j)^\top] = \mathbf{0}$ because the measurement noises are mutually independent. This recursive formulation can also be rewritten explicitly as a sum of the covariances:

$$\mathbf{P}_{k|k}^{i,j} = \mathbf{T}_{0,k}^i \mathbf{P}_{0|0} (\mathbf{T}_{0,k}^j)^\top + \sum_{\tau=1}^k \mathbf{T}_{\tau,k}^i \mathbf{Q}_\tau (\mathbf{T}_{\tau,k}^j)^\top, \tag{11}$$

where at every time step τ we include the new process noise \mathbf{Q}_τ . The matrix $\mathbf{T}_{\tau,k}$ denote the individual matrix transformations that are a result of the local KFs [see equations (9) and (10)]. In large sensor networks, keeping track of these correlations can be cumbersome and often infeasible as it requires full communication of all processing steps. Therefore, the methods in [34] and [35] propose the use of square-root decompositions to keep track of correlated estimation errors.

III. SQUARE-ROOT DECOMPOSITION OF COMMON PROCESS NOISE

The following section revisits our previous work about the square-root decomposition algorithm. It was originally formulated only for the fusion in centralized sensor networks with only one dedicated fusion center. The basic idea is a sliding window mechanism for a square-root decomposition of the track correlations. Every node updates and saves its history of processing steps in a matrix containing all square-root decompositions of common prior information and common process noise. During the fusion step, every node transmits its state estimate, covariance matrix, and square-root matrix. The square-root matrix allows us to reconstruct the joint covariance matrix to fuse the local estimates according to (7) and (8). The recursive formula of (11) is reformulated as a square-root decomposition as

$$\begin{aligned}
\mathbf{P}_{k|k}^{i,j} &= \mathbf{T}_{0,k}^i \sqrt{\mathbf{P}_{0|0}} (\sqrt{\mathbf{P}_{0|0}})^\top (\mathbf{T}_{0,k}^j)^\top \\
&\quad + \sum_{\tau=1}^k \mathbf{T}_{\tau,k}^i \sqrt{\mathbf{Q}_\tau} (\sqrt{\mathbf{Q}_\tau})^\top (\mathbf{T}_\tau^j)^\top \\
&= \sum_{\tau=0}^k \boldsymbol{\Sigma}_{\tau,Q}^i (\boldsymbol{\Sigma}_{\tau,Q}^j)^\top.
\end{aligned}$$

Each sensor node stores its square-root terms in the matrix

$$\mathbf{S}_{k,Q}^i = \left[\boldsymbol{\Sigma}_{0,Q}^i, \boldsymbol{\Sigma}_{1,Q}^i, \dots, \boldsymbol{\Sigma}_{k,Q}^i \right],$$

which includes all noise terms until the current time step k and has the dimension $M = N \times D = N \times (N + (k -$

$1) \times W)$. The calculation of this matrix can be done recursively. At time step $k = 0$, it is initialized with

$$\mathbf{S}_{0,Q}^i = \boldsymbol{\Sigma}_{0,Q}^i = \sqrt{\mathbf{P}_0},$$

and the matrix is then linearly transformed by the time update, and a new noise term $\boldsymbol{\Sigma}_{k,Q}^i = \sqrt{\mathbf{Q}_k}$ is included. Furthermore, the matrix is then updated using the gain matrix of the KF update $\mathbf{L}_k^i = \mathbf{I} - \mathbf{K}_k^i \mathbf{C}_k^i$

$$\mathbf{S}_{k,Q}^i = \mathbf{L}_k^i [\mathbf{A}_k^i \mathbf{S}_{k-1,Q}^i, \boldsymbol{\Sigma}_{k,Q}^i].$$

When the fusion step is reached, the cross-covariance matrix between node i and node j can be reconstructed as

$$\mathbf{P}_{k,Q}^{i,j} = \sum_{m=0}^k \boldsymbol{\Sigma}_{m,Q}^i (\boldsymbol{\Sigma}_{m,Q}^j)^\top = \mathbf{S}_{k,Q}^i (\mathbf{S}_{k,Q}^j)^\top. \tag{12}$$

By including a new process noise term at every time update, the square-root decomposition matrix $\mathbf{S}_{k,Q}^i$ will continue to grow linearly in size. Since communication bandwidth is limited in sensor networks, we need to find a trade-off between the optimal decomposition of correlated estimation errors and the communication capacity.

A. Limiting the Number of Square-Root Decomposition Terms for Process Noise and Common Prior Information

In order to keep the number of entries in the square-root decomposition matrix constant, the square-root matrix is decomposed [35] into two parts:

$$\mathbf{S}_k^i = [\mathbf{S}_{k,\mathcal{T}_Q}^i, \mathbf{S}_{k,\Omega}^i],$$

where $\mathbf{S}_{k,\mathcal{T}_Q}^i$ is a moving horizon square-root decomposition matrix

$$\mathbf{S}_{k,\mathcal{T}_Q}^i = \left[\boldsymbol{\Sigma}_{k-\mathcal{T}_Q+1}^i, \boldsymbol{\Sigma}_{k-\mathcal{T}_Q+2}^i, \dots, \boldsymbol{\Sigma}_k^i \right], \tag{13}$$

which will only include the dependent noise terms up to a user-defined time horizon \mathcal{T}_Q . The remaining noise terms will be removed from the square-root matrix and summarized in a residual $\mathbf{S}_{k,\Omega}^i$. This residual has to be bounded in order to obtain a consistent fusion result. To formulate the fusion rule, we consider the optimal joint covariance matrix

$$\mathbf{J}_k = \begin{bmatrix} \mathbf{P}_k^i & \mathbf{P}_k^{i,j} \\ \mathbf{P}_k^{j,i} & \mathbf{P}_k^j \end{bmatrix}.$$

We can now decompose this matrix into a part $\mathbf{P}_{k,\mathcal{T}_Q}^{i,j}$ that we can reconstruct and a part $\mathbf{P}_{k,\Omega}^{i,j}$ that is correlated but whose exact correlation we cannot reconstruct anymore, i.e.,

$$\mathbf{J}_k = \begin{bmatrix} \mathbf{P}_k^i & \mathbf{P}_{k,\mathcal{T}_Q}^{i,j} + \mathbf{P}_{k,\Omega}^{i,j} \\ \mathbf{P}_{k,\mathcal{T}_Q}^{j,i} + \mathbf{P}_{k,\Omega}^{j,i} & \mathbf{P}_k^j \end{bmatrix}.$$

This residual can be calculated recursively and includes all correlated noise terms not included in the square-root

matrix $\mathbf{S}_{k,\mathcal{T}}^i$. With the residual, we obtain

$$\mathbf{S}_{k,\Omega}^i (\mathbf{S}_{k,\Omega}^i)^\top = \mathbf{\Omega}_{k,Q}^i. \quad (14)$$

We now aim to find a bound according to

$$\begin{bmatrix} \frac{1}{\omega} \mathbf{\Omega}_{k,Q}^i & 0 \\ 0 & \frac{1}{1-\omega} \mathbf{\Omega}_{k,Q}^j \end{bmatrix} \geq \begin{bmatrix} \mathbf{\Omega}_{k,Q}^i & \mathbf{P}_{k,\Omega}^{i,j} \\ \mathbf{P}_{k,\Omega}^{j,i} & \mathbf{\Omega}_{k,Q}^j \end{bmatrix}.$$

Finally, we can now formulate the new suboptimal joint covariance matrix

$$\tilde{\mathbf{J}}_k = \begin{bmatrix} \mathbf{P}_{k,Q}^i - \mathbf{\Omega}_{k,Q}^i & \mathbf{P}_{k,\mathcal{T}_Q}^{i,j} \\ \mathbf{P}_{k,\mathcal{T}_Q}^{j,i} & \mathbf{P}_{k,Q}^j - \mathbf{\Omega}_{k,Q}^j \end{bmatrix} + \begin{bmatrix} \frac{1}{\omega} \mathbf{\Omega}_{k,Q}^i & 0 \\ 0 & \frac{1}{1-\omega} \mathbf{\Omega}_{k,Q}^j \end{bmatrix} \geq \mathbf{J}_k, \quad (15)$$

which we will use for the fusion step according to formulas (3) and (4). The weighting factors ω can be found by minimizing the fused covariance matrix according to formula (7). Alternatively, an approximate solution such as the one proposed by [34] and [38] can be used. Although suboptimal, we used the latter approach for its simple implementation and fast execution time. The weighting factor can be calculated by

$$\omega = \frac{1/\text{tr}(\mathbf{\Omega}_Q^i)}{1/\text{tr}(\mathbf{\Omega}_Q^i) + 1/\text{tr}(\mathbf{\Omega}_Q^j)}.$$

Afterwards, the formula given in (6) yields

$$\mathbf{F}^j = \left(\mathbf{P}^i + \frac{1}{\omega} \mathbf{\Omega}_Q^i - \mathbf{P}_{\mathcal{T}_Q}^{i,j} \right) \times \left(\mathbf{P}^i + \frac{1}{\omega} \mathbf{\Omega}_Q^i + \mathbf{P}^j + \frac{1}{1-\omega} \mathbf{\Omega}_Q^j - \mathbf{P}_{\mathcal{T}_Q}^{j,i} - \mathbf{P}_{\mathcal{T}_Q}^{i,i} \right)^{-1}.$$

Last, the fused covariance and fused state can be calculated according to equations (7) and (8).

IV. EXTENSION TO THE FUSION IN DECENTRALIZED SENSOR NETWORKS

The square-root decomposition enables the nodes to encode correlated process noise and correlated prior information in a distributed fashion. The central node in Fig. 1(a) does not need to keep track of the correlations, processing steps, or number of nodes as all the required information is provided by the nodes themselves. Modifications to the square-root decomposition are necessary when nodes are organized in hierarchical network topologies, as shown in Fig. 1(b), where intermediate fusion nodes exist. Each fusion step alters the correlation structure among the nodes, which has to be encoded properly and is discussed in Section IV.A. The decentralized network architecture depicted in Fig. 1(c) exhibits cycles that lead to double-counting of information. Section IV.B discusses how additional data structures

can be introduced to cover correlations due to double-counting of measurements and thus correlated measurement errors.

A. Hierarchical Fusion

In a hierarchical fusion architecture, nodes may fuse estimates and pass them to the upper layer for a subsequent fusion step. Hence, such intermediate fusion nodes have to take into account correlations for the fusion but simultaneously have to compute an updated square-root decomposition for the subsequent fusion steps. Each node i can fuse its estimate with an estimate received from node j by using the fusion formulas (3) and (4). The required cross-covariance matrices $\mathbf{P}^{i,j} = (\mathbf{P}^{j,i})^\top$ are obtained by the square-root decomposition, i.e., by using (12).

For the subsequent fusion layer, the square-root decomposition needs to encode the correlation structure of the fusion result $\hat{\mathbf{x}}^f$. The cross-covariance matrix for this fusion result $\hat{\mathbf{x}}^f$ and the estimate $\hat{\mathbf{x}}^l$ of a third node l yields

$$\begin{aligned} \mathbf{P}^{f,l} &= E[(\hat{\mathbf{x}}^f - \mathbf{x})(\hat{\mathbf{x}}^l - \mathbf{x})^\top] \\ &= E[(\mathbf{F}^i \hat{\mathbf{x}}^i + \mathbf{F}^j \hat{\mathbf{x}}^j - \mathbf{x})(\hat{\mathbf{x}}^l - \mathbf{x})^\top] \\ &= \mathbf{F}^i \mathbf{P}^{i,l} + \mathbf{F}^j \mathbf{P}^{j,l}. \end{aligned}$$

The dependencies $\mathbf{P}^{i,l}$ and $\mathbf{P}^{j,l}$ are given by the corresponding square-root decompositions, i.e.,

$$\mathbf{P}^{i,l} = \mathbf{S}_Q^i (\mathbf{S}_Q^l)^\top \text{ and } \mathbf{P}^{j,l} = \mathbf{S}_Q^j (\mathbf{S}_Q^l)^\top.$$

Hence, the fused square-root decomposition for the $\mathbf{P}^{f,l}$ has the form

$$\mathbf{S}_Q^f = \mathbf{F}^i \mathbf{S}_Q^i + \mathbf{F}^j \mathbf{S}_Q^j, \quad (16)$$

which gives $\mathbf{P}^{f,l} = \mathbf{S}_Q^f (\mathbf{S}_Q^l)^\top$ for any l .

For a finite horizon \mathcal{T}_Q , \mathbf{S}^f only partially covers the correlations, and the fusion node also has to update the residual term (14). According to the chosen weight ω in (15), the residual becomes

$$\begin{aligned} \mathbf{\Omega}_Q^f &= \frac{1}{\omega} \mathbf{F}^i \mathbf{\Omega}_Q^i (\mathbf{F}^i)^\top + \frac{1}{1-\omega} \mathbf{F}^j \mathbf{\Omega}_Q^j (\mathbf{F}^j)^\top \\ &\geq \mathbf{F}^i \mathbf{\Omega}_Q^i (\mathbf{F}^i)^\top + \mathbf{F}^j \mathbf{\Omega}_Q^{i,j} (\mathbf{F}^j)^\top \\ &\quad + \mathbf{F}^j \mathbf{\Omega}_Q^{j,i} (\mathbf{F}^i)^\top + \mathbf{F}^j \mathbf{\Omega}_Q^j (\mathbf{F}^j)^\top, \end{aligned} \quad (17)$$

which is a bound since any information about $\mathbf{\Omega}_Q^{i,j}$ has been discarded.

B. Double-Counting

Double-counting occurs when two nodes i and j fuse their estimates for a second time. In other words, the information sent out by node i circles back to this node over possibly multiple hops and processing steps. Not only common process noise then leads to correlations, but also measurements incorporated in the estimates

reappear at the nodes and introduce additional correlations. In the latter case, two estimates are to be fused that share the same information. The cross-covariance matrix between the fused estimate $\hat{\underline{x}}^f$ and the estimate $\hat{\underline{x}}^i$ of node i yields

$$\begin{aligned} \mathbf{P}^{f,i} &= E[(\hat{\underline{x}}^f - \underline{x})(\hat{\underline{x}}^i - \underline{x})^T] \\ &= E[(\mathbf{F}^i \hat{\underline{x}}^i + \mathbf{F}^j \hat{\underline{x}}^j - \underline{x})(\hat{\underline{x}}^i - \underline{x})^T] \\ &= \mathbf{F}^i \mathbf{P}^{i,i} + \mathbf{F}^j \mathbf{P}^{j,i}. \end{aligned}$$

The cross-covariance $\mathbf{P}^{j,i}$ can be calculated as discussed in Section II.B. The matrix $\mathbf{P}^{j,i}$ represents the correlated estimation errors of sensor node i and is equal to the covariance matrix

$$\begin{aligned} \mathbf{P}_{k|k}^i &= E[(\hat{\underline{x}}_{k|k}^i - \underline{x}_k)(\hat{\underline{x}}_{k|k}^i - \underline{x}_k)^T] \\ &= E\left[(\hat{\underline{x}}_{k|k-1}^i + \mathbf{K}_k^i \mathbf{z}_k^i - \underline{x}_k)(\hat{\underline{x}}_{k|k-1}^i + \mathbf{K}_k^i \mathbf{z}_k^i - \underline{x}_k)^T\right] \\ &= E\left[(\hat{\underline{x}}_{k|k-1}^i + \mathbf{K}_k^i(\mathbf{v}_k^i - \mathbf{C}_k^i \hat{\underline{x}}_{k|k-1}^i) - \underline{x}_k)\right. \\ &\quad \left. \times (\hat{\underline{x}}_{k|k-1}^i + \mathbf{K}_k^i(\mathbf{v}_k^i - \mathbf{C}_k^i \hat{\underline{x}}_{k|k-1}^i) - \underline{x}_k)^T\right] \\ &= \mathbf{L}_k^i \mathbf{P}_{k|k-1}^i (\mathbf{L}_k^i)^T + \mathbf{K}_k^i \mathbf{R}^i (\mathbf{K}_k^i)^T, \end{aligned}$$

with the KF update $\mathbf{L}_k^i = \mathbf{I} - \mathbf{K}_k^i \mathbf{C}_k^i$. For this reason, each node i needs to keep track of an additional list of measurement noise terms

$$\mathbf{S}_{k,R^i}^i = [\boldsymbol{\Sigma}_{0,R^i}^i, \boldsymbol{\Sigma}_{1,R^i}^i, \dots, \boldsymbol{\Sigma}_{k,R^i}^i], \quad (18)$$

to account for double-counting of measurements. It is initialized at time step $k = 1$ with

$$\boldsymbol{\Sigma}_{1,R^i}^i = \boldsymbol{\Sigma}_{1,R^i}^i = \mathbf{K}_1^i \sqrt{\mathbf{R}_1^i},$$

where \mathbf{R}_1^i is the measurement covariance matrix of the first measurement (2) acquired by node i . The matrix \mathbf{K}_1^i is the Kalman gain used in this measurement update. The matrix \mathbf{S}_{k,R^i}^i is recursively updated according to¹

$$\mathbf{S}_{k,R^i}^i = [\mathbf{L}_k^i \mathbf{A}_k^i \mathbf{S}_{k-1,R^i}^i, \boldsymbol{\Sigma}_{k,R^i}^i] \quad (19)$$

with

$$\boldsymbol{\Sigma}_{k,R^i}^i = \mathbf{K}_k^i \sqrt{\mathbf{R}_k^i}.$$

When two sensor nodes exchange estimates for fusion, they also pass on all the square-root matrices. These matrices need to be kept separate from each other to trace back possible sources of double-counting. Node i that receives an estimate from node j then also keeps and manages the set \mathbf{S}_{k,R^i}^j , which is the corresponding set (19) from node j . The own and the received square-root matrices are updated similarly to (16) and (17) by

$$\begin{aligned} \mathbf{S}_{R^i}^f &= \mathbf{F}^i \mathbf{S}_{R^i}^i + \mathbf{F}^j \mathbf{S}_{R^i}^j, \\ \mathbf{S}_{R^j}^f &= \mathbf{F}^i \mathbf{S}_{R^j}^i + \mathbf{F}^j \mathbf{S}_{R^j}^j. \end{aligned}$$

¹Note that \mathbf{L}_k^i in [1] should be inside the brackets.

Bookkeeping of the received \mathbf{S}_{k,R^i}^j resembles (19). However, it differs in that it is filled with zeros during further processing according to

$$\mathbf{S}_{k,R^i}^j = \mathbf{L}_k^i [\mathbf{A}_k^i \mathbf{S}_{k-1,R^i}^j, \mathbf{0}] \quad (20)$$

as the measurement noise affecting node j is uncorrelated with the estimation errors at node i for the following time steps.

The square-root matrix \mathbf{S}_{k,R^i}^j can be used in a later fusion step to retrieve the cross-covariances stemming from the previous fusion step by

$$\mathbf{P}_{k,R}^{i,j} = \mathbf{S}_{k,R^i}^i (\mathbf{S}_{k,R^i}^j)^T + \mathbf{S}_{k,R^j}^j (\mathbf{S}_{k,R^i}^j)^T, \quad (21)$$

where \mathbf{S}_{k,R^i}^j is the common information with node i that has been tracked in node j . More precisely, \mathbf{S}_{k,R^i}^j is the corresponding set to (20) that was generated by node j when it received information from i . The reconstructed cross-covariance matrix (21) has to be combined with $\mathbf{P}_{k,Q}^{i,j}$ representing the common process noise, which finally results in the full cross-covariance matrix

$$\mathbf{P}_k^{i,j} = \mathbf{P}_{k,Q}^{i,j} + \mathbf{P}_{k,R}^{i,j}.$$

The amount of data that need to be stored and updated by each node grows linearly over time. Especially in networks with many sensor nodes, conservative bounding techniques can allow the nodes to surpass this burden.

1) **Limiting the Number of Square-Root Decomposition Terms for Measurement Noise:** Following the concept in Section III.A, we limit the number of processing steps encoded in the square-root decompositions to a fixed time horizon \mathcal{T}_R . The matrix (18) becomes

$$\mathbf{S}_R^i = [\boldsymbol{\Sigma}_{R,k-\mathcal{T}_R+1}^i, \boldsymbol{\Sigma}_{R,k-\mathcal{T}_R+2}^i, \dots, \boldsymbol{\Sigma}_{R,k}^i],$$

which has a constant number of entries. The remainder of the matrix is summarized in the residual term $\boldsymbol{\Omega}_R^i$. When two estimates are fused, a bound on the residual matrix as in (17) has to be computed by

$$\boldsymbol{\Omega}_R^f = \frac{1}{\omega} \mathbf{F}^i \boldsymbol{\Omega}_R^i (\mathbf{F}^i)^T + \frac{1}{1-\omega} \mathbf{F}^j \boldsymbol{\Omega}_R^j (\mathbf{F}^j)^T.$$

This bound also has to be combined with the residual bound (17) for the process noise.

2) **Keeping Track of Uncorrelated Measurements:** The treatment of correlated measurement information and double-counting can be simplified by computing a more general bound on the measurement covariance. This approach circumvents the explicit bookkeeping (18) of the information shared through the fusion of estimates.

The local covariance matrix of sensor node i is rewritten as

$$\mathbf{P}^i = \mathbf{P}_{Q,\mathcal{T}_Q} + \mathbf{P}_{Q,\Omega} + \mathbf{P}_R,$$

where $\mathbf{P}_{Q,\mathcal{T}_Q}$ accounts for the reconstructable cross-covariance matrix using (13), $\mathbf{P}_{Q,\Omega}$ accounts for the

residual (14), and \mathbf{P}_R represents possibly correlated measurement noise. We further separate this into

$$\mathbf{P}_R = \mathbf{P}_R^+ + \mathbf{P}_R^-,$$

where \mathbf{P}_R^+ denotes correlated measurement noise and \mathbf{P}_R^- denotes uncorrelated measurement noise. We can safely assume that measurements that have been obtained between fusion steps and thus have not been shared are uncorrelated. Therefore, only the part accounting for information that has been shared with other sensor nodes before is correlated and needs to be bounded. The uncorrelated measurement noise residual $\mathbf{P}_{k,R}^-$ can be calculated recursively:

$$\mathbf{P}_{k,R}^- = \mathbf{L}\mathbf{A}\mathbf{P}_{k-1,R}^-\mathbf{A}^T\mathbf{L}^T + \mathbf{K}_k^i\mathbf{R}_1^i(\mathbf{K}_k^i)^T.$$

To ensure the correctness of this assumption, $\mathbf{P}_{k,R}^-$ will be reset to the zero matrix when the fusion step has been executed or the information has been shared with other sensor nodes. The correlated measurement residual is calculated by

$$\mathbf{\Omega}_R^i = \mathbf{P}^i - \mathbf{S}_Q^i(\mathbf{S}_Q^i)^T - \mathbf{\Omega}_Q^i - \mathbf{P}_R^-.$$

The bounded part of the joint covariance matrix becomes

$$\mathbf{\Omega}_k^i = \mathbf{\Omega}_{k,Q}^i + \mathbf{\Omega}_{k,R}^i.$$

The rest of the fusion step is analogous to (15).

V. EVALUATION

The following section features three distinct examples to highlight the performance of the proposed algorithm under different conditions. First, we discuss an example using only two sensor nodes that constantly exchange information, which leads to highly correlated estimates. Second, we discuss the convergence rate of the proposed algorithm and compare it with standard consensus algorithms. Last, a tracking example using 25 heterogeneous sensor nodes in a sparse network but with synchronized fusion steps is analyzed.

A. Two Sensor Nodes

We consider two sensor nodes A and B , which observe the discrete-time time-invariant linear stochastic system in (1) with the parameters

$$\mathbf{A} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \Delta T = 0.1.$$

Both sensor nodes draw observations using the linear measurement model (2), where every measurement is corrupted by additive-white Gaussian noise \mathbf{v}_k^i with covariance matrix $\mathbf{R}^A = \mathbf{R}^B = 50$ and measurement matrices

$$\mathbf{C}^A = [1 \quad 0], \quad \mathbf{C}^B = [0 \quad 1].$$

Both sensor nodes are initialized with $\mathbf{P}_0 = 5\mathbf{Q}$ and $\hat{\mathbf{x}}_0 = [0 \quad 0]^T$. The data exchange between the two nodes is performed as follows:

- 1) both sensor nodes execute a local filter update,
- 2) node A sends its local information to node B ,
- 3) node B fuses information according to the selected fusion method and reinitializes its local state and covariance matrix with new fused information,
- 4) both sensor nodes execute a local filter update,
- 5) node B sends its local information to node A ,
- 6) node A fuses information according to the selected fusion method and reinitializes its local state and covariance matrix with new fused information,
- 7) repeat from beginning.

We calculate the MSE of both sensor nodes and then calculate the average. Fig. 3(a) shows the averaged MSE of both sensor nodes for 1000 Monte Carlo runs (MCRs). The results are compared with the optimal fusion result. The optimal fusion result is obtained by optimally keeping track of the cross-covariance matrices between the state estimates and performing the fusion step in one dedicated fusion center using a centralized network topology. After the fusion step is executed, the local state estimates and covariances matrices are reinitialized with the fusion result. This approach shows the lowest MSE as expected. The MSE of the naïve fusion, which neglects the correlations between state estimates, immediately diverges. The proposed square-root decomposition (SqDF) is shown in several configurations. The time horizon for the square-root matrix is $\mathcal{T}_Q = 5$. The square-root decomposition without bounding (SqDF_{no}) shows a relatively high MSE as it does not account for older process noise or any correlation due to measurement noise. Bounding of process noise (SqDF_{Qb}) performs a bit better in comparison as it does bound the process noise but also does not account for possibly correlated measurements. Covariance intersection performs better than SqDF_{no} and SqDF_{Qb}, but its performance is limited as it cannot account for uncorrelated parts. Using the proposed algorithm with partial bounding of measurement noise (SqDF_{Rbp}, see Section IV.B2) shows better performance than covariance intersection, as it can find a tighter bound. The proposed method from Section IV.B1 using the limited time horizon \mathcal{T}_R for the track-keeping of correlated measurement noise is also compared to the other methods. The square-root decomposition using a time horizon of $\mathcal{T}_R = 5$ (SqDF_{Rb1}) shows a lower MSE compared to all other methods. The square-root decomposition using a smaller time horizon of $\mathcal{T}_R = 2$ (SqDF_{Rb2}) is comparable to the performance of CI.

Fig. 3(b) shows the averaged normalized estimation error squared (ANEES) over both sensor nodes. The ANEES is a measure to determine whether the actual uncertainty matches the expected uncertainty [39].

TABLE I
Abbreviations and Parameterizations for Two Sensor Nodes Example

Method	Short		Parameterization
Covariance intersection [16]	CI	—+—	—
Naïve fusion [15]	Naïve	—+—	—
Optimal fusion [11] (central)	Optimal	—	—
Square-root decomposition (Section III)	SqDF _{no}	—	$\mathcal{T}_Q = 5$
Square-root decomposition (Section III.A)	SqDF _{Ob}	—	$\mathcal{T}_Q = 5$
Square-root decomposition (Section IV.B2)	SqDF _{Rbp}	—	$\mathcal{T}_Q = 5$
Square-root decomposition (Section IV.B1)	SqDF _{Rb1}	—	$\mathcal{T}_Q = 5, \mathcal{T}_R = 5$
Square-root decomposition (Section IV.B1)	SqDF _{Rb2}	—	$\mathcal{T}_Q = 5, \mathcal{T}_R = 2$

An ANEES below 1 indicates a conservative fusion estimate, while an ANEES above 1 indicates an underestimation of the actual uncertainty. Naïve fusion again diverges very fast and is therefore not included in the plot, and covariance intersection is overly conservative. Both methods without bounding (SqDF_{no} and SqDF_{Ob}) are inconsistent as it would be expected. The algorithm with partial bounding is close to 1, meaning that the actual MSE of the fused results matches the covariance matrix. The proposed methods using a limited time horizon to keep track of correlated measurement noise (SqDF_{Rb1} and SqDF_{Rb2}) are very close to the optimal fusion result but slightly more conservative, where SqDF_{Rb2} shows similar performance to the proposed method with the partial bounding of correlated measurement errors (SqDF_{Rbp}).

A summary of all used abbreviations and parameterizations of the used methods can be found in Table I.

B. Consensus between States

In the following example, we discuss how fast the proposed algorithm converges toward a global consensus. Consensus problems have been intensively studied in many different contexts [24]. Instead of accounting for dependencies within the network, consensus algorithms average the information of neighboring nodes iteratively until all sensor nodes have converged asymptotically to a global estimate [29]. While finding a consensus is usually not the goal of fusion algorithms, it is an interesting problem to investigate the effect of double-counting in sensor networks. This section demonstrates that a careful consideration of dependencies improves the convergence rate toward a global consensus. We define the averaged consensus estimate error (ACEE), which indicates the degree of consensus among estimates from all nodes in the network (see also [28]), as

$$\text{ACEE} = \frac{1}{N_s} \sum_{i=1}^{N_s} (\hat{x}^i - \bar{x}), \quad \bar{x} = \frac{1}{N_s} \sum_{i=1}^{N_s} \hat{x}^i.$$

We consider a network of ten sensor nodes with ring topology [see Fig. 2(a)]. The system description is similar to the one in Example 1, but the measurement covariances are reduced to $\mathbf{R}^A = \mathbf{R}^B = 0.2$ to decrease

oscillation. The sensor nodes alternate between the measurement model of node *A* and node *B*, which can also be seen in the figure. The sensor nodes first perform ten filtering steps independently and then communicate their local information toward their neighbors multiple times. The fusion algorithms are also compared with consensus algorithms, specifically consensus on measurements [24] (Cons_M), consensus on information [26] (Cons_I), and a hybrid consensus method called DHIWCF [28], which performs a consensus on measurement on the first iteration and a consensus on information afterward. All consensus methods are performed using Metropolis weights. We would like to point out that many consensus algorithms have been proposed in recent years and that the utilized algorithms may not be best tailored to the considered problem. A summary of all used abbreviations and parameterizations of the used methods can be found in Table II. Fig. 4(a) shows the convergence rate of the state estimates. Covariance Intersection (CI) and naïve fusion show very similar convergence rates. All consensus methods converge only slightly slower. The hybrid consensus algorithm DHIWCF lies between consensus on measurements and consensus on information. Furthermore, we see that the square-root decomposition of the measurement noise improves the convergence rate. Keeping track of all measurements (SqDF_{Opt}) leads to the fastest convergence, followed by the square-root decomposition with a time horizon $\mathcal{T}_R = 3$ (SqDF_{Rb1}) and using a time horizon $\mathcal{T}_R = 1$ (SqDF_{Rb2}), thus showing that even a short time horizon for the measurement

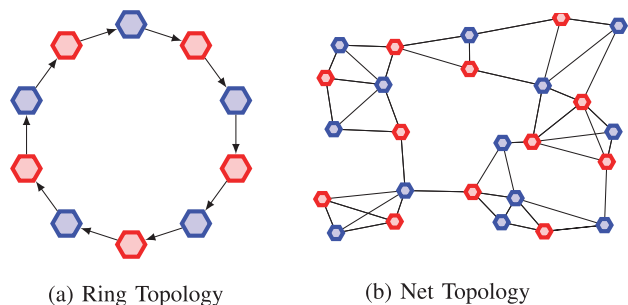












Fig. 2. Network topologies. Magenta nodes are using one measurement model and blue nodes are using the other measurement model.

TABLE II
Abbreviations and Parameterizations for Consensus Example

Method	Short		Parameterization
Covariance intersection [16]	CI		-
Naïve fusion [15]	Naïve		-
Optimal fusion [11] (central)	Optimal		-
Square-root decomposition (Section IV.B2)	SqDF _{Rbp}		$\mathcal{T}_Q = 11$
Square-root decomposition (Section IV.B)	SqDF _{Opt}		$\mathcal{T}_Q = 11, \mathcal{T}_R = 10$
Square-root decomposition (Section IV.B1)	SqDF _{Rb1}		$\mathcal{T}_Q = 11, \mathcal{T}_R = 5$
Square-root decomposition (Section IV.B1)	SqDF _{Rb2}		$\mathcal{T}_Q = 11, \mathcal{T}_R = 2$
Consensus on measurements [24]	Cons _M		Metropolis weights
Consensus on information [26]	Cons _M		Metropolis weights
Hybrid consensus filter [28]	DHIWCF		Metropolis weights

noise might make a huge difference. The time horizon of the square root matrix keeping track of the process noise is $\mathcal{T}_Q = 11$. Therefore, process noise and common prior information are fully tracked.

For further comparison, we computed the MSE for all sensor nodes and showed the averaged MSE in Fig. 4(b). Compared with all other fusion methods, the optimal track keeping of correlations achieves the lowest MSE fastest and almost approaches the centralized optimal fusion result. The square-root decomposition with a smaller time horizon, SqDF_{Rb1} and SqDF_{Rb2}, also performs well but converges more slowly. Consensus on information does not show any performance improvements in comparison to the other fusion methods. On the other hand, consensus on measurements converges slightly slower but outperforms all other methods after 18 time steps. The hybrid method DHIWCF shows slightly lower performance. Both consensus methods reach a lower average MSE because the utilization of measurement information is more effective than the exclusive fusion of state estimates.

Lastly, in Fig. 4(c), it can be seen that the average ANEES over all sensor nodes in the network is close to the optimal fusion result for SqDF_{Rb1}, SqDF_{Opt}, and SqDF_{Rb2}. All square-root decomposition-based algorithms that bound the measurement partially or fully are very close to the performance of covariance intersection and, therefore, overly conservative. Consensus on information shows similar performance to covariance intersection but performs slightly worse because Metropolis weights do not minimize the trace or determinant of the fused covariance matrix. The performance of consensus on measurements depends on the utilized correction weights to mitigate the averaging of measurements [40]. We chose the correction weight as 2 in the first consensus step when only two measurements are available to the sensor node. Then, we increment the correction weight by 1 in every consensus step until 10 to account for the ten measurements once a consensus is reached. Because of the averaging characteristics, the ANEES will start to rise as some measurements have higher weights than others during the averaging, leading to double-counting. Once the consensus is approached,

the ANEES will converge toward 1 again, meaning that the method will be consistent after a certain amount of time. DHIWCF shows slightly less conservative results than covariance intersection. This means that it can reach a relatively low MSE while still achieving consistent results, which is an interesting finding. Yet, the best trade-off between convergence rate, MSE, and consistency can be achieved using the proposed method.

C. Large-Scale Sparse Network

In our last example, we consider a simple tracking example featuring 25 sensor nodes in a sparse network, as depicted in Fig. 2(b). Nodes always receive information from the three closest sensor nodes. The movement of the tracked object is described by

$$\underline{\mathbf{x}}_{k+1} = \mathbf{A}\underline{\mathbf{x}}_k + \underline{\mathbf{w}}_k \text{ with } \underline{\mathbf{w}}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}),$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{Q} = 0.1 \begin{bmatrix} \frac{1}{3}\Delta T & 0 & \frac{1}{2}\Delta T & 0 \\ 0 & \frac{1}{3}\Delta T & 0 & \frac{1}{2}\Delta T \\ \frac{1}{2}\Delta T & 0 & \Delta T & 0 \\ 0 & \frac{1}{2}\Delta T & 0 & \Delta T \end{bmatrix}, \Delta T = 0.1.$$

Referring again to Fig. 2(b), the blue nodes observe the bearing toward a moving target and the red nodes the range. Their observation is described by a nonlinear measurement function

$$y_k^i = h^i(\underline{\mathbf{x}}_k) + \underline{\mathbf{v}}_k,$$

where nodes alternate between measuring the bearing or the range toward a moving target:

$$h^i(\underline{\mathbf{x}}_k) = \begin{cases} \text{atan2}(x_{y,k} - P_y^i, x_{x,k} - P_x^i) & \text{if } i \text{ is odd,} \\ \sqrt{(x_{x,k} - P_x^i)^2 + (x_{y,k} - P_y^i)^2} & \text{if } i \text{ is even,} \end{cases}$$

TABLE III
Abbreviations and Parameterizations for Large-Scale Network Example

Method	Short		Parameterization
Covariance intersection [16]	CI		-
Naïve fusion [15]	Naïve		-
Optimal fusion [11] (central)	Optimal		-
Square-root decomposition (Section III)	SqDF _{no}		$\mathcal{T}_Q = 5$
Square-root decomposition (Section III.A)	SqDF _{Qb}		$\mathcal{T}_Q = 5$
Square-root decomposition (Section IV.B2)	SqDF _{Rbp}		$\mathcal{T}_Q = 5$
Channel filter [21]	ChF		-

with measurement noise

$$\mathbf{R}^i = \left(\frac{2\pi}{180}\right)^2 \text{ if } i \text{ is odd, or } \mathbf{R}^i = 0.01 \text{ m}^2 \text{ if } i \text{ is even}$$

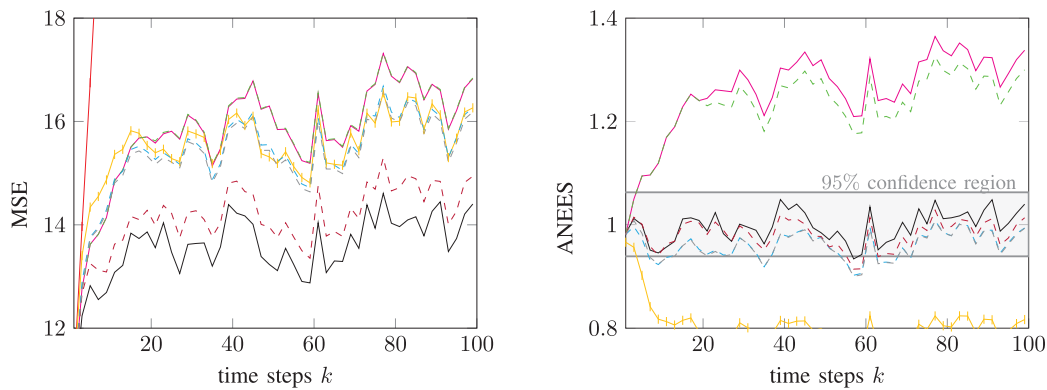
at the sensor node position $P^i = [P_x^i, P_y^i]^T$. The nodes are placed at random on a 10 m × 10 m field. They perform a synchronized fusion step at every fifth time step. Since the most recent five measurements are hence uncorrelated, a square-root decomposition of the measurement noise is not needed as only older measurements are correlated and their influence is increasingly becoming weaker. Therefore, we will utilize the additional information about uncorrelated measurements for the fusion. A summary of all used abbreviations and parameterizations of the used methods can be found in Table III. Fig. 5(a) shows the average MSE over all 25 sensor nodes. The time horizon for keeping track of process noise is $\mathcal{T}_Q = 5$. The results of the partial bounding SqDF_{Rbp} and the square-root decomposition without accounting for correlated measurements SqDF_{Qb} have the lowest MSE. As expected, the partial bounding SqDF_{Rbp} is more conservative than SqDF_{Qb} as indicated by the ANEES [see Fig. 5(b)]. We also observe that SqDF_{Qb} is even consistent, i.e., the ANEES is close to 1, which can be due to correlations that cancel each other out because of symmetries within the considered setup.

We also compared our proposed algorithm to the ChF [21], [41], which can be seen as a first-order approximation of the information graph technique. While the

ChF is suboptimal because it does not account for all common information, it might be only slightly suboptimal if the time between the occurrence of correlated estimation errors and the current fusion step is large enough. Furthermore, it requires very little additional computation and communication. Fig. 5 shows that the ChF's MSE is very close to the fusion result using CI. Yet, the ANEES indicates that the fusion result is consistent.

VI. RESULTS AND DISCUSSION

The second example shows that the convergence rate is improved when cross-covariances can be reconstructed accurately. However, the fusion can lead to numerical issues when sensor nodes are highly correlated since the joint covariance matrix cannot be inverted properly. While the additional square-root decomposition of the measurement noise is beneficial, it leads to additional communication, which grows with the number of sensor nodes. It might be possible to discard parts of these square roots when they traveled too far from their source. Therefore, correlations would only be tracked within a particular region of interest around a sensor node, which might improve the scalability of the algorithm. The choice of the time horizon determining the number of encoded dependent noise terms



(a) Averaged MSE of both sensor nodes.

(b) Averaged ANEES of both sensor nodes.

Fig. 3. Comparison of the fusion results of different algorithms for 1000 MCRs.

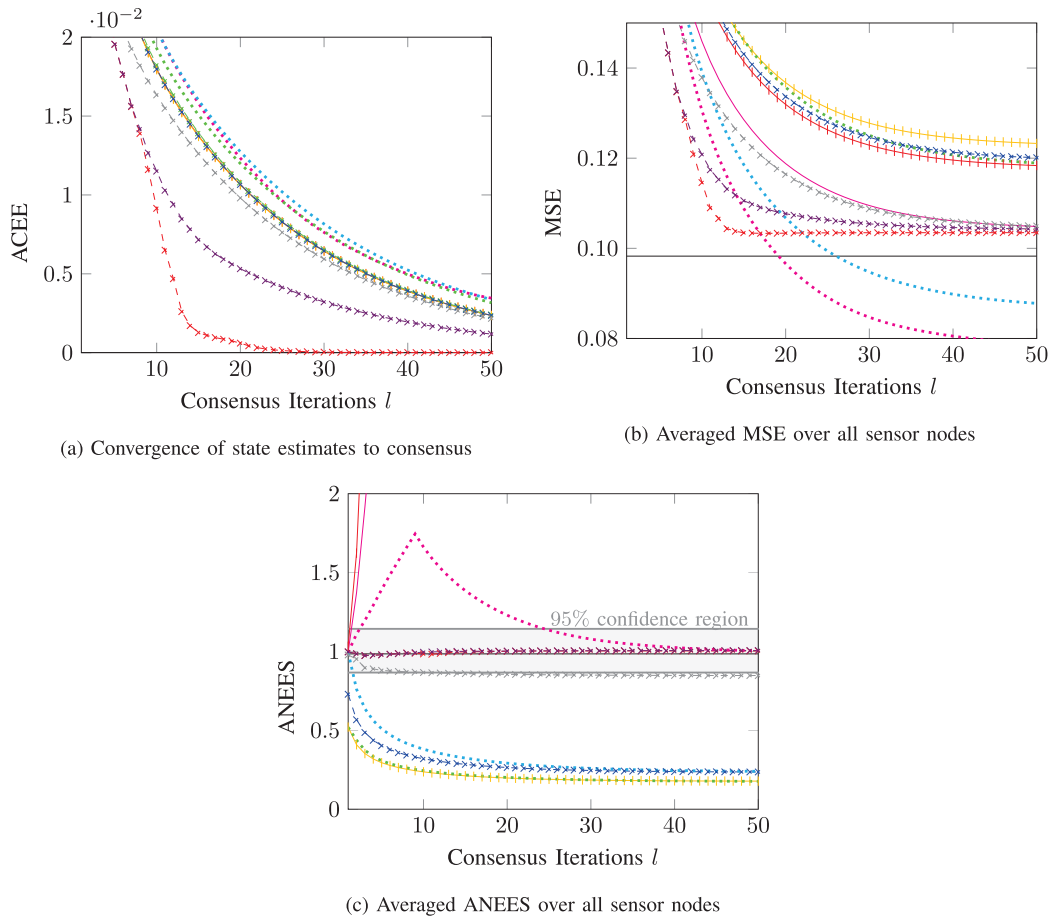


Fig. 4. Convergence of state estimates toward a common consensus and MSE for ring topology with 200 MCRs.

highly depends on the application and needs thorough consideration.

VII. CONCLUSION

This paper aims at solving the problem of fusing multiple state estimates in different network topologies with unknown correlations. The proposed method utilizes the square-root decomposition of correlated noise

covariances. The advantage of this approach is that every node can keep track of its local processing steps independently, and, thus, no dedicated fusion center is necessary to manage the sensor nodes or their communication with each other.

The results show that the fused estimate remains consistent in arbitrary network topologies and that the fusion results of several sensor nodes converge faster toward a consensus than other fusion methods

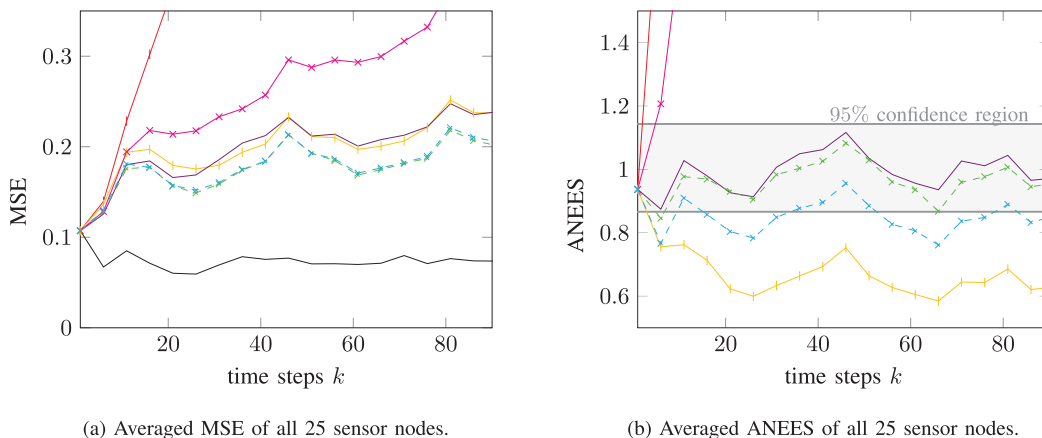


Fig. 5. Comparison of the fusion results of different algorithms for 100 MCRs.

while yielding more consistent results than consensus methods. The downside of this method is the increased amount of additional information that needs to be shared and constantly updated. Therefore, the choice of the right time horizon might be crucial for the performance in many applications.

The findings of this paper make several contributions to the current literature. First, the modification of cross-covariances between state estimates due to intermediate fusion steps is discussed. Second, the additional dependency due to the double-counting of measurement information is examined. The provided method can be tailored to the needs of the application by tuning the time horizon for the number of tracked correlated noise covariances to meet the bandwidth requirements. Furthermore, the time horizon for common prior information and common process noise can be chosen independently from the time horizon for common measurement information. This allows to only keep track of correlated estimation errors that contribute the most to the cross-covariance. By choosing a shorter time horizon, the fusion result becomes more conservative while still being a tighter bound than most other conservative fusion methods. In addition, the time horizon can be adjusted within the sensor network to provide more accuracy in certain areas where it is needed while allowing for rougher estimates in others.

Because of its flexibility, the method can even be utilized in low-cost sensor networks. An exciting application is the cooperative localization of robots, where many sources of correlated estimation errors occur, which are usually only addressed in a conservative fashion.

REFERENCES

- [1] S. Radtke, B. Noack, and U. D. Hanebeck
“Fully decentralized estimation using square-root decompositions,”
in *Proc. 23rd Int. Conf. Inf. Fusion*, New York, NY: IEEE, Jul. 2020, pp. 1–8.
- [2] D. L. Hall, C.-Y. Chong, J. Llinas, and M. E. Liggins II, Eds.,
Distributed Data Fusion for Network-Centric Operations, ser. The Electrical Engineering and Applied Signal Processing Series. CRC Press, 2013.
- [3] S. Grime and H. F. Durrant-Whyte
“Data fusion in decentralized sensor networks,”
Contr. Eng. Pract., vol. 2, no. 5, pp. 849–863, Oct. 1994.
- [4] C. Chong, K. Chang, and S. Mori
“A review of forty years of distributed estimation,”
in *Proc. 21st Int. Conf. Inf. Fusion*, Cambridge, United Kingdom, Jul. 2018, pp. 1–8.
- [5] A. G. Mutambara
Decentralized Estimation and Control for Multisensor Systems. CRC Press, 1998.
- [6] A. G. O. Mutambara and H. F. Durrant-Whyte
“Nonlinear information space: a practical basis for decentralization,”
in *Proc. SPIE’s Int. Symp. Photon. Ind. Appl.*, P. S. Schenker, Ed., vol. 2355. New York, NY: SPIE, 1994, pp. 97–105.
- [7] B. Noack, D. Lyons, M. Nagel, and U. D. Hanebeck
“Nonlinear information filtering for distributed multisensor data fusion,”
in *Proc. 2011 Amer. Contr. Conf.*, San Francisco, CA, USA, Jun. 2011, pp. 4846–4852.
- [8] J. L. Speyer
“Computation and transmission requirements for a decentralized linear-quadratic-gaussian control problem,”
IEEE Trans. Autom. Contr., vol. 24, no. 2, pp. 266–269, Apr. 1979.
- [9] F. Govaers and W. Koch
“An exact solution to track-to-track fusion at arbitrary communication rates,”
IEEE Trans. Aerosp. Electron. Syst., vol. 48, no. 3, Jul. 2012, pp. 2718–2729.
- [10] M. Reinhardt, B. Noack, and U. D. Hanebeck
“The hypothesizing distributed Kalman filter,”
in *Proc. 2012 IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst.*, Hamburg, Germany, Sep. 2012, pp. 305–312.
- [11] Y. Bar-Shalom and L. Campo
“On the track-to-track correlation problem,”
IEEE Trans. Autom. Contr., vol. 26, no. 2, pp. 571–572, Apr. 1981.
- [12] C.-Y. Chong, K.-C. Chang, and S. Mori
“Distributed tracking in distributed sensor networks,”
in *Proc. 1986 Amer. Contr. Conf.*, Seattle, WA, USA, 1986, pp. 1863–1868.
- [13] B. Noack, J. Sijs, M. Reinhardt, and U. D. Hanebeck
“Treatment of dependent information in multisensor Kalman filtering and data fusion,”
in *Multisensor Data Fusion: From Algorithms and Architectural Design to Applications*, H. Fourati, Ed. CRC Press, Aug. 2015, pp. 169–192.
- [14] Y. Bar-Shalom and L. Campo
“The effect of the common process noise on the two-sensor fused-track covariance,”
IEEE Trans. Aerosp. Electron. Syst., vol. AES-22, no. 6, pp. 803–805, 1986.
- [15] C.-Y. Chong and S. Mori
“Convex combination and covariance intersection algorithms in distributed fusion,”
in *Proc. 4th Int. Conf. Inf. Fusion*, Montréal, Québec, Canada, Aug. 2001.
- [16] S. J. Julier and J. K. Uhlmann
“A non-divergent estimation algorithm in the presence of unknown correlations,”
in *Proc. IEEE Amer. Contr. Conf.*, vol. 4, Albuquerque, NM, USA, Jun. 1997, pp. 2369–2373.
- [17] L. Chen, P. O. Arambel, and R. K. Mehra
“Fusion under unknown correlation: covariance intersection revisited,”
IEEE Trans. Autom. Contr., vol. 47, no. 11, pp. 1879–1882, Nov. 2002.
- [18] M. Reinhardt, B. Noack, P. O. Arambel, and U. D. Hanebeck
“Minimum covariance bounds for the fusion under unknown correlations,”
IEEE Signal Process. Lett., vol. 22, no. 9, pp. 1210–1214, Sep. 2015.
- [19] B. Noack, J. Sijs, M. Reinhardt, and U. D. Hanebeck
“Decentralized data fusion with inverse covariance intersection,”
Automatica, vol. 79, pp. 35–41, May 2017.
- [20] B. Noack, J. Sijs, and U. D. Hanebeck
“Inverse covariance intersection: new insights and properties,”
in *Proc. 20th Int. Conf. Inf. Fusion*, Xi’an, China, Jul. 2017, pp. 1–8.
- [21] C.-Y. Chong, K.-C. Chang, and S. Mori
“Fundamentals of distributed estimation,”

- in *Distributed Data Fusion for Network-Centric Operations*, ser. The Electrical Engineering and Applied Signal Processing Series, D. L. Hall, C.-Y. Chong, J. Llinas, and M. E. Liggins II, Eds. CRC Press, 2013, pp. 95–124.
- [22] C.-Y. Chong
“Hierarchical estimation,”
in *MIT/ONR Workshop C3 Syst.*, Monterey, CA, USA, 1979, pp. 205–220.
- [23] X. Tian, Y. Bar-Shalom, T. Yuan, E. Blasch, K. Pham, and G. Chen
“A generalized information matrix fusion based heterogeneous track-to-track fusion algorithm,”
Proc. SPIE, vol. 8050, 2011, pp. 469–476.
- [24] R. Olfati-Saber
“Distributed Kalman filter with embedded consensus filters,”
in *Proc. 44th IEEE Conf. Decis. Contr. Eur. Contr. Conf.*, Sevilla, Spain, Dec. 2005, pp. 8179–8184.
- [25] G. Battistelli and L. Chisci,
“Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability,”
Automatica, vol. 50, no. 3, p. 707–718, Mar. 2014.
- [26] R. Olfati-Saber
“Distributed Kalman filtering for sensor networks,”
in *Proc. 46th IEEE Conf. Decis. Contr.*, New Orleans, LA, USA, Dec. 2007, pp. 5492–5498.
- [27] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano
“Consensus-based linear and nonlinear filtering,”
IEEE Trans. Autom. Contr., vol. 60, no. 5, pp. 1410–1415, May 2015.
- [28] J. Liu, Y. Liu, K. Dong, Z. Ding, and Y. He
“A novel distributed state estimation algorithm with consensus strategy,”
Sensors, vol. 19, no. 9, pp. 1–29, 2019.
- [29] C.-Y. Chong, K.-C. Chang, and S. Mori
“Comparison of optimal distributed estimation and consensus filtering,”
in *Proc. 19th Int. Conf. Inf. Fusion*, 2016, pp. 1034–1041.
- [30] J. Čurn, D. Marinescu, N. O’Hara, and V. Cahill
“Data incest in cooperative localisation with the common past-invariant ensemble Kalman filter,”
in *Proc. 16th Int. Conf. Inf. Fusion*, Istanbul, Turkey, Jul. 2013, pp. 68–76.
- [31] M. Reinhardt, B. Noack, and U. D. Hanebeck
“Reconstruction of joint covariance matrices in networked linear systems,”
in *Proc. 48th Annu. Conf. Inf. Sci. Syst.*, Princeton, NJ, USA, Mar. 2014, pp. 1–6.
- [32] J. Steinbring, B. Noack, M. Reinhardt, and U. D. Hanebeck
“Optimal sample-based fusion for distributed state estimation,”
in *Proc. 19th Int. Conf. Inf. Fusion*, Heidelberg, Germany, Jul. 2016, pp. 1600–1607.
- [33] S. Radtke, B. Noack, U. D. Hanebeck, and O. Straka
“Reconstruction of cross-correlations with constant number of deterministic samples,”
in *Proc. 21st Int. Conf. Inf. Fusion*, Cambridge, U.K., Jul. 2018, pp. 1638–1645.
- [34] M. Reinhardt, S. Kulkarni, and U. D. Hanebeck
“Generalized covariance intersection based on noise decomposition,”
in *Proc. 2014 IEEE Int. Conf. Multisensor Fusion Inf. Integr.*, Beijing, China, Sep. 2014, pp. 1–8.
- [35] S. Radtke, B. Noack, and U. D. Hanebeck
“Distributed estimation using square root decompositions of dependent information,”
in *Proc. 22nd Int. Conf. Inf. Fusion*, Ottawa, Canada, Jul. 2019, pp. 1–8.
- [36] K.-C. Chang, R. K. Saha, and Y. Bar-Shalom
“On optimal track-to-track fusion,”
IEEE Trans. Aerosp. Electron. Syst., vol. 33, no. 4, pp. 1271–1276, Oct. 1997.
- [37] X. R. Li, Yunmin Zhu, Jie Wang, and Chongzhao Han
“Optimal linear estimation fusion – part I: unified fusion rules,”
IEEE Trans. Inf. Theory, vol. 49, no. 9, pp. 2192–2208, 2003.
- [38] W. Niehsen,
“Information fusion based on fast covariance intersection filtering,”
in *Proc. 5th Int. Conf. Inf. Fusion*, Annapolis, MD, USA, Jul. 2002, pp. 901–904.
- [39] X. R. Li, Z. Zhao, and V. P. Jilkov
“Practical measures and test for credibility of an estimator,”
in *Proc. Workshop Estimation, Tracking, Fusion*, 2001, pp. 481–495.
- [40] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano
“Consensus-based linear and nonlinear filtering,”
IEEE Trans. Autom. Contr., vol. 60, no. 5, pp. 1410–1415, 2015.
- [41] K. Chang, C. Chong, and S. Mori
“Analytical and computational evaluation of scalable distributed fusion algorithms,”
IEEE Trans. Aerosp. Electron. Syst., vol. 46, no. 4, pp. 2022–2034, 2010.



Susanne Radtke received the Master’s degree in mechatronics from the Technical University of Ilmenau, Germany, in 2016. Since 2017, she has been a Ph.D. student at the Intelligent Sensor Actuator-Systems Laboratory, Karlsruhe Institute of Technology (KIT), Germany. Her research interests include distributed estimation and data fusion in sensor networks.



Benjamin Noack received the Diploma in Computer Science from the Karlsruhe Institute of Technology (KIT), Germany, in 2009. Afterwards, he received the Ph.D. degree in 2013 from the Intelligent Sensor Actuator-Systems Laboratory, Karlsruhe Institute of Technology (KIT), Germany. Since 2013, he has been a Senior Researcher with the Karlsruhe Institute of Technology (KIT), Germany. His research interests are in the areas of multisensor data fusion, distributed and decentralized Kalman filtering, combined stochastic and set-membership approaches to state estimation, and event-based systems.



Uwe D. Hanebeck received the Ph.D. degree in 1997 and the habilitation degree in 2003, both in electrical engineering from the Technical University in Munich, Germany. He is a Chaired Professor of Computer Science with the Karlsruhe Institute of Technology (KIT), Germany, and the Director of the Intelligent Sensor Actuator-Systems Laboratory, Karlsruhe Institute of Technology (KIT), Germany. His research interests are in the areas of information fusion, nonlinear state estimation, stochastic modeling, system identification, and control with a strong emphasis on theory-driven approaches based on stochastic system theory and uncertainty models. He is an author and coauthor of more than 500 publications in various high-ranking journals and conferences and is an IEEE Fellow.

Analysis of Costs for the GNP Problem

MARK LEVEDAHL
JOHN D. GLASS

Track-to-track data association in a multisensor framework involves score functions to determine a solution. When sensor errors include both random noise and unknown bias terms, several options are available. Of these, two options are the global nearest pattern match (GNPM) and marginal track-to-track association (MTTA) scores. The former involves a joint likelihood of bias and association hypothesis and the latter is the result of integrating the total probability space over the unknown bias to remove the bias likelihood. Analytically, we show that the difference between these scores is the determinant of the a-posteriori bias covariance, and that the same bias estimation is inherent in both. Using a simple numerical example, we compare the weight each score formulation apportions to track assignment hypotheses based on the quality of the bias estimate, and show that GNPM tends to favor hypotheses with low a-posteriori bias covariance. Additionally, through evaluation of the incremental cost structure, we argue that the non-assignment cost used in both scores is nearly optimal, in the sense of correct associations, for GNPM. However, the same non-assignment cost is not optimal for the MTTA score, and the significance depends upon the uncertainty of bias and the number of associations made.

Manuscript received November 23, 2020; revised May 19, 2021; released for publication June 30, 2021.

Dr. Ramona Georgescu handled refereeing of this contribution.

Mark Levedahl and John D. Glass are with Raytheon Technologies, Waltham, MA 02451, USA (e-mail: {mark.d.levedahl, john.d.glass}@rtx.com).

1557-6418/21/\$17.00 © 2021 JAIF

I. INTRODUCTION

ASSOCIATING sets of observations from sensor systems is fundamental in multi-sensor tracking. With reliable multi-sensor track assignment, track fusion can achieve improved accuracy and allow handover of data from one sensor to another [4]. Furthermore, distributed sensor systems allow coverage of larger areas with different viewing angles and facilitate the formation of a complete track picture [6]. Basic complications that prohibit perfect track-to-track association are unknown residual bias errors, random errors contained in the observations of a sensor, unknown true target motion, and heterogeneous sensor coverage. Residual bias may arise from imperfections in sensor registration, transformation errors, and other sources, whereas random errors arise from stochastic effects of sensor systems such as thermal noise. Missed detections are often the result of sensor sensitivity/phenomenology and other aspects such as sensor resolution, thus causing heterogeneous sensor coverage. Unknown target motion may also induce error in the estimated track state, regardless of other errors, yielding cross-correlated error across sensors. Mathematical models of these sensor errors form the foundation of modern track-to-track association algorithms.

Track association in a multisensor framework involves score functions to assess alternate association hypotheses. Any hypothesized association of tracks implies a set of observed targets and locations, with the score function providing the probability the given tracks arise from common targets specified by the association hypothesis. These score functions in general have unknown, possibly random, parameters (e.g., location of targets) implicitly set such that the score is maximized at the observed values [12], [13], and thus evaluate how well each hypothesis fits the data. The classic formulation is termed the global nearest neighbor (GNN) problem and addresses random errors with heterogeneous sensor coverage, but assumes independent errors per track, ignoring bias errors [5], [6], [16]. Since the assignment score of a track tuple is independent of others, GNN is an N-D assignment problem with costs in the form of negative log-likelihoods based on assumed statistical models. Solving the two-sensor case is very efficient with solvers such as the auction or Jonker–Volgenant–Castanon algorithms. For a survey on solution methods to GNN see [22]. To handle heterogeneous sensor coverage, GNN algorithms include the cost of particular tracks not assigning, based on a-priori assumptions by which targets may appear. Uniform spatial distribution of targets in the surveillance volume with the total number of targets as Poisson distributed are standard assumptions in the literature.

When multiple sensors track the same target, errors of those tracks can become cross-correlated, assuming these errors arise due to common process noise. The basic ideas are in [2], including discussion of handling more than two sensors. As shown in [2], the scores of

track tuples remain independent with inclusion of cross-correlated errors of this form, maintaining the ability to use GNN solution methods. A more difficult problem arises when there are cross-correlated errors across a set of tracks from a sensor, herein called “bias” errors, though these errors need not be time-invariant nor 100% cross-correlated. This problem is very important when the magnitude of the residual bias errors is significant compared to the sum of target spacing and independent errors in tracks. A simple mitigation with a GNN algorithm is to inflate covariances to cover both the random errors and the residual bias. However, as shown in [16], the method of covariance inflation gives poor association performance as the magnitude of bias grows. Approaches that are more sophisticated attempt to recognize the bias and provide specific treatment. Early techniques involved sequential methods that first attempt to estimate and remove the bias, then use GNN as if the tracks are unbiased [24]. More recent techniques jointly solve for residual bias and assignments within the mathematical formulation of the problem.

A full treatment of bias errors requires different scoring functions than used in GNN. The global nearest pattern match (GNPM) function, presented in [16] for the case of two sensors, includes the most probable bias per hypothesis in the score. A variant of this approach based upon marginalizing the bias estimate, termed marginalized track-to-track association (MTTA), is presented in [20], again for the case of two sensors. These scoring functions facilitate solutions to what we call the global nearest pattern (GNP) problem. Compared to early solutions to the GNP problem that focused on independent bias estimation and assignment steps, the novelty of GNPM and MTTA is in the explicit treatment of sensor bias in the scoring functions, leading to joint assignment and bias estimation [16], [20]. As shown in [16], this joint approach can provide significantly improved data association performance compared to GNN even when bias errors are a small fraction of the independent random errors. The work of [7] and [14] extends the GNPM function for the N-sensor case, including cross-correlation due to process noise.

The GNP assignment problem is much more difficult to solve than the related GNN problem as the costs are not separable into independent costs per track pair. Instead, GNP gives coupled costs based upon the hypothesis dependent bias estimate, breaking the assumption underlying use of standard assignment solvers for this problem. For problems with only a handful of hypotheses to choose from, a feasible solution is to enumerate and score all. However, in many real-world cases, this approach is infeasible. The 6×8 association problem we investigate in Section III gives 93,289 possible hypotheses, illustrating how even a handful of tracks give a high number of total hypotheses. Addressing this problem, Levedahl in [17] provides a Dijkstra shortest path technique for providing the K-best solutions to the GNP problem, applicable to both the GNPM and MTTA cost

functions, and discusses performance (both runtime and accuracy) compared to GNN in [16]. Papageorgiou in [21] provides additional specialized mathematical programs for solving these problems, again including discussion of accuracy and runtime issues. The techniques discussed above have proved practical and useful in real time for problems much larger than the 6×8 problem included here.

It is worth noting that, in general, these techniques assume the targets within a single GNP problem have a common bias offset represented in the same dimension of the state space. Strictly speaking, the assumption of a common relative bias offset to the sets of data is seldom true in practice. For example, a registration bias is often modeled as additive constants in the measurement space of range and angle as in [19], which affects Cartesian tracks in a non-linear fashion. Thus, a small azimuth bias $\delta\theta$ affects position as range multiplied by $\delta\theta$, but also the velocity as the latter vector estimate has been rotated by $\delta\theta$. So long as the targets are not widely dispersed, an assumption of common bias is reasonable. We prefer to think of the common bias assumption as a linear approximation of a non-linear bias model about the centroid of the targets of interest. Conversely, a non-linear bias model of specific range and azimuth offsets for each sensor needs wide dispersion among targets to yield favorable observability [19], and requires non-linear estimation techniques. In addition, widely dispersed targets tend to unambiguous association problems where GNN covariance inflation approaches may suffice. GNP methods are appropriate where bias errors are significant compared to the noise error and inter-target spacing, and targets in the problem are not widely dispersed such that the common bias model is unreasonable. Regardless, the common bias representation is an approximation made by numerous authors, including [7], [8], [9], [14], [16], [17], [20], and [24], and is the focus of this work. We leave any extensions to non-linear bias models as future work, in part because such extensions preclude the closed-form solutions essential to the comparisons made in this paper.

The key objective of this paper is to understand the full mathematical foundation and relationship of GNPM and MTTA, along with sound mathematical rationale for selection between them. Therefore, we ignore any extension to greater than two sensors, and ignore any extension to general cross-correlated errors beyond sensor bias. Generally speaking, GNPM assumes that the most probable bias is a key variable in the association problem, while MTTA treats bias as a nuisance parameter and marginalizes bias in the score. Ferry in [8] and [9] also makes arguments based in Bayesian methodology in agreement with Papageorgiou’s treatment of bias, but Ferry incorporates fundamentally different a-priori target assumptions, most importantly that targets appear spatially according to a Gaussian distribution rather than uniform as GNPM/MTTA assumes. A benefit of the Gaussian assumption is closed-form integrals

rather than the approximations needed for the uniform case, but results in equations much more complex than GNPM/MTTA and are hard to decompose in a fashion that allow efficient solution. We note that the perceived value of bias marginalization by the authors of [20] is in ambiguity management, claiming that bias likelihood can be a corrupting presence in correctly determining the probabilities of various association hypotheses. Other authors have attributed the difficulty of reliable probability determination to the integral approximations inherent in the posterior [15], precisely the integral targeted in the work of Ferry in [8] and [9]. Although the work of [15] demonstrates that this integral approximation becomes less accurate in dense target scenarios where ambiguity management is critical, the role this integral plays was not discussed in [20]. We further note that the key metric used in this work, association accuracy, directly scores whether the highest probability hypothesis is most correct, and is a necessary but insufficient criterion to achieving correct hypothesis probabilities. For the problems investigated here, our findings show MTTA is sometimes worse, and never better, than GNPM in association accuracy.

We investigate GNPM/MTTA against the key criterion of maximizing the number of correct assignments, as that is the fundamental objective of data association. In Section II of this contribution, we start from the basic assumptions of the track association problem in the presence of bias to derive the exact difference between the GNPM and MTTA score functions. A part of this derivation includes expressing the GNPM and MTTA scores as a new, yet equivalent, stacked Gaussian density equation. We show that although MTTA has the bias term removed through integration, the same relative bias estimation of GNPM is implicit in the MTTA assignment score. Intuitively, we show that the difference between the two scores is the determinant of the covariance of the a-posteriori bias estimate, very similar to the conclusions made in [12] for marginalization of target locations. Leveraging this result, in Section III we elaborate on the practical differences between the GNPM and MTTA scores through analytic and numerical examples. Critically, we evaluate association performance for various non-assignment costs and show that the non-assignment cost often cited in the GNN/GNP literature is nearly optimal for GNPM in the sense of maximizing the probability of correct association. However, as the uncertainty of residual bias grows, this non-assignment cost can be far from optimal for MTTA. The covariance of the a-posteriori bias estimate in the MTTA cost is precisely the source of sub-optimal assignment performance, therefore any adjustments made to MTTA that maximize correct assignments give GNPM. Our findings show that GNPM can be much more accurate than MTTA for few assignments or large bias errors, and MTTA is never more accurate than GNPM. Therefore, we recommend the use of GNPM for the problems described here. We provide concluding remarks in Section IV.

II. GNP SCORES AND COSTS

In this section, we start from a basic mathematical description of the track association problem and derive the necessary total and conditional distributions required to reveal the relationship between the GNPM and MTTA assignment scores. For the nomenclature used in this paper, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a multivariate normal distribution of mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. All vectors are assumed column vectors denoted in lowercase bold, and matrices are uppercase bold. To reduce nomenclature complexity, we use $\mathbf{0}$ to denote either the zero matrix or the zero vector, which is obvious in the context of usage.

A. Observation Model

Assume n_t targets denoted as \mathbf{x}_t , $t = 1, \dots, n_t$, observed by sensors \mathcal{A} and \mathcal{B} , each observing a potentially different subset of targets. Assume sensor \mathcal{A} develops m distinct observations and \mathcal{B} develops n distinct observations with no false or redundant observations from either sensor. Without loss of generality, assume the observations satisfy $m \leq n \leq n_t$. Sensor \mathcal{A} observations are corrupted by zero-mean random noise with covariance $\mathbf{S}_{\mathcal{A},i}$, uncorrelated for each observation. Therefore the observations from \mathcal{A} take the form

$$\mathbf{x}_i^{\mathcal{A}} = \mathbf{x}_{\alpha_i} + \mathbf{n}_i^{\mathcal{A}}, \quad (1)$$

$$p(\mathbf{n}_i^{\mathcal{A}}) = \mathcal{N}(\mathbf{0}, \mathbf{S}_{\mathcal{A},i}), \quad (2)$$

where α_i is an unknown index to the target tracked. Observations from sensor \mathcal{B} follow a similar model but with errors specific to that sensor including an unknown relative bias term \mathbf{b} . Therefore

$$\mathbf{x}_j^{\mathcal{B}} = \mathbf{x}_{\eta_j} - \mathbf{b} + \mathbf{n}_j^{\mathcal{B}}, \quad (3)$$

$$p(\mathbf{n}_j^{\mathcal{B}}) = \mathcal{N}(\mathbf{0}, \mathbf{S}_{\mathcal{B},j}), \quad (4)$$

where η_j is an unknown index to the target tracked for sensor \mathcal{B} . The single relative bias term (relative to the coordinate frame of sensor \mathcal{A}) is assumed common to all observations from sensor \mathcal{B} and has the probability distribution

$$p(\mathbf{b}) = \mathcal{N}(\mathbf{0}, \mathbf{R}). \quad (5)$$

The covariances $\mathbf{S}_{\mathcal{A},i}$, $\mathbf{S}_{\mathcal{B},j}$, and \mathbf{R} are all assumed to be symmetric positive-definite matrices, and the dimension of all sensor observations is assumed to be of dimension d .

The goal of track-to-track assignment is to determine the underlying truth commonality in the observations. Truth commonality is represented as the i and j indexes such that $\alpha_i = \eta_j$. Since the actual ordering of the targets is arbitrary and unknown, we equivalently seek the assignment of tracks from sensor \mathcal{A} to sensor \mathcal{B} . Define the assignment vector as $\mathbf{h} = [h_1 \dots h_m]^T$ of length m

where the i^{th} element indicates the index in \mathcal{B} that is assigned to the i^{th} observation in \mathcal{A} . Unassigned observations in \mathcal{A} are indicated with an h_i of zero. Therefore, let $\mathcal{J} = \{i : h_i \neq 0\}$ be the set of assigned track indexes and $n_a = |\mathcal{J}|$ be the number of assignments in \mathbf{h} . Assuming a uniform prior on each \mathbf{x}_i location and that the number of targets in the surveillance volume is Poisson distributed, following the derivation in [14], the posterior probability of an assignment hypothesis and bias can be written as¹

$$\begin{aligned} & \Pr(\mathbf{h}, \mathbf{b} | \mathbf{x}_1^{\mathcal{A}}, \dots, \mathbf{x}_m^{\mathcal{A}}, \mathbf{x}_1^{\mathcal{B}}, \dots, \mathbf{x}_n^{\mathcal{B}}) \\ &= C \frac{1}{\sqrt{|2\pi\mathbf{R}|}} e^{-\frac{1}{2}\mathbf{b}^T\mathbf{R}^{-1}\mathbf{b}} \\ & \quad \times (\beta P_{A\bar{B}})^{n-n_a} (\beta P_{\bar{A}B})^{m-n_a} (\beta P_{AB})^{n_a} \\ & \quad \times \prod_{i \in \mathcal{J}} \frac{1}{\sqrt{|2\pi\mathbf{S}_i|}} e^{-\frac{1}{2}(\mathbf{x}_i^{\Delta} - \mathbf{b})^T \mathbf{S}_i^{-1} (\mathbf{x}_i^{\Delta} - \mathbf{b})}, \end{aligned} \quad (6)$$

with the difference terms and associated covariances expressed as

$$\begin{aligned} \mathbf{x}_i^{\Delta} &= \mathbf{x}_i^{\mathcal{A}} - \mathbf{x}_{h_i}^{\mathcal{B}}, \\ \mathbf{S}_i &= \mathbf{S}_{\mathcal{A},i} + \mathbf{S}_{\mathcal{B},h_i}, \end{aligned} \quad (7)$$

for all $i \in \mathcal{J}$. The β term is the spatial density of the targets, P_{AB} is the probability that both sensor \mathcal{A} and \mathcal{B} observe a target, $P_{A\bar{B}}$ is the probability that sensor \mathcal{A} but not \mathcal{B} observe a target, $P_{\bar{A}B}$ is the probability that sensor \mathcal{B} but not \mathcal{A} observe a target, and C is a normalizing constant.² Of significance in (6) is the sufficient statistic of an assignment hypothesis as the absolute difference between the track states, \mathbf{x}_i^{Δ} . As noted in Corollary 1 of [14], incorporation of cross-correlated errors between $\mathbf{x}_i^{\mathcal{A}}$ and $\mathbf{x}_{h_i}^{\mathcal{B}}$ due to common process noise involves a simple subtraction term to \mathbf{S}_i , which can be easily inserted into (7). We choose to leave that term omitted since we have not studied the effects of common process noise in our numerical simulations, but anticipate no impact upon the conclusions reached. As will be discussed in upcoming sections, GNPM is the joint posterior of (6), while MTTA requires the additional step of marginalizing \mathbf{b} .

B. Probability Distributions of Bias and Errors

Any joint probability density has an equivalence with marginal and conditional densities. Block forms of the

¹The authors in [14] generalized to more than two sensors, with a separate bias term per sensor instead of a single relative bias.

²A slight distinction with the derivation in [14] is the detection probabilities as hypothesized in \mathbf{h} , which are conditioned on the event that at least one sensor detected the target (i.e., undetected targets do not enter the assignment problem). Some authors have also made this distinction explicit as in [11] or [18]. We also note that C scales all hypothesis scores equally so is not needed for finding the best hypothesis, and in general is not determined as doing so may require enumerating all possible assignment hypotheses.

random vectors described in the observation model allow the use of the fundamental equations of linear estimation [3] to give marginal and conditional densities. Defining $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_{n_a}]$ to be a length n_a vector that contains an ordering of the indices in \mathcal{J} , the stacked vector of absolute differences of assigned tracks from (7) as

$$\mathbf{x}_{\Delta} = \begin{bmatrix} \mathbf{x}_{\gamma_1}^{\Delta} \\ \vdots \\ \mathbf{x}_{\gamma_{n_a}}^{\Delta} \end{bmatrix}, \quad (8)$$

and the block identity matrix as

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} \\ \vdots \\ \mathbf{I} \end{bmatrix}, \quad (9)$$

with n_a blocks of $d \times d$ identity matrices, the following marginal and conditional distributions are derived in Appendix A:

$$p(\mathbf{x}_{\Delta}) = \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\mathbf{x}_{\Delta}}), \quad (10)$$

$$p(\mathbf{x}_{\Delta} | \mathbf{b}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}_{\Delta} | \mathbf{b}}, \mathbf{Q}_{\mathbf{x}_{\Delta} | \mathbf{b}}), \quad (11)$$

$$p(\mathbf{b} | \mathbf{x}_{\Delta}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{b} | \mathbf{x}_{\Delta}}, \mathbf{Q}_{\mathbf{b} | \mathbf{x}_{\Delta}}), \quad (12)$$

with the corresponding elements as

$$\mathbf{Q}_{\mathbf{x}_{\Delta}} = \mathbf{Q}_{\mathbf{x}_{\Delta} | \mathbf{b}} + \mathbf{H}\mathbf{R}\mathbf{H}^T, \quad (13)$$

$$\boldsymbol{\mu}_{\mathbf{x}_{\Delta} | \mathbf{b}} = \mathbf{H}\mathbf{b}, \quad (14)$$

$$\mathbf{Q}_{\mathbf{x}_{\Delta} | \mathbf{b}} = \begin{bmatrix} \mathbf{S}_{\gamma_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_{\gamma_{n_a}} \end{bmatrix}, \quad (15)$$

$$\boldsymbol{\mu}_{\mathbf{b} | \mathbf{x}_{\Delta}} = \mathbf{P}_{\mathbf{b}\mathbf{x}_{\Delta}}^T \mathbf{Q}_{\mathbf{x}_{\Delta}}^{-1} \mathbf{x}_{\Delta} \quad (16)$$

$$\mathbf{Q}_{\mathbf{b} | \mathbf{x}_{\Delta}} = \mathbf{R} - \mathbf{P}_{\mathbf{b}\mathbf{x}_{\Delta}}^T \mathbf{Q}_{\mathbf{x}_{\Delta}}^{-1} \mathbf{P}_{\mathbf{b}\mathbf{x}_{\Delta}}, \quad (17)$$

$$\mathbf{P}_{\mathbf{b}\mathbf{x}_{\Delta}} = \mathbf{H}\mathbf{R}. \quad (18)$$

Each of these probability densities relate to the likelihood of a track assignment hypothesis and bias. Upon conversion into the block structure, (11) is the final term in (6), therefore

$$\begin{aligned} & \Pr(\mathbf{h}, \mathbf{b} | \mathbf{x}_1^{\mathcal{A}}, \dots, \mathbf{x}_m^{\mathcal{A}}, \mathbf{x}_1^{\mathcal{B}}, \dots, \mathbf{x}_n^{\mathcal{B}}) \\ &= C (\beta P_{A\bar{B}})^{n-n_a} (\beta P_{\bar{A}B})^{m-n_a} (\beta P_{AB})^{n_a} p(\mathbf{x}_{\Delta} | \mathbf{b}) p(\mathbf{b}) \end{aligned} \quad (19)$$

It is worth noting that $p(\mathbf{x}_{\Delta} | \mathbf{b})$ in (19) is the likelihood function of the bias and hypothesis given the data, although the conditioning term only mentions \mathbf{b} . By inspection of (7), the \mathbf{x}_{Δ} notation depends on the hypothesis, and therefore we do not add \mathbf{h} as a conditioning term. We use the $p(\mathbf{x}_{\Delta} | \mathbf{b})$ notation to identify

that the likelihood is a function of the differences of assigned tracks in a particular hypothesis, in addition to simplicity compared to the more formal, yet equivalent, $p(\mathbf{x}_1^A, \dots, \mathbf{x}_m^A, \mathbf{x}_1^B, \dots, \mathbf{x}_n^B | \mathbf{h}, \mathbf{b})$. Careful readers may also notice that (19) is not written explicitly as a function of $p(\mathbf{h})$. However, following the derivation in [14] $p(\mathbf{h})$ is part of the product of scalar terms in (19). More specifically, the derivation of (6) in [14] involves conditioning \mathbf{h} to the abstract α_i and η_j indices from (1) and (3), respectively, along with the unknown number of targets, which upon simplification gives $p(\mathbf{h})$ as being a contributor to the term $C(\beta P_{AB})^{n-n_a} (\beta P_{\bar{A}B})^{m-n_a} (\beta P_{AB})^{n_a}$. Rigorous technical details of the posterior density derivation exist in previous literature, including [9] and [14]. We also note that many authors use the word ‘‘likelihood’’ liberally when referring to posteriors and related terms, sometimes by admission as the authors of MTTA in [20]. In this work, we prefer to maintain more strict terminology usage, particularly with the use of the word likelihood as a specific contribution to the posterior. Furthermore, we define the product of likelihood and bias prior, $p(\mathbf{x}_\Delta | \mathbf{b})p(\mathbf{b})$, as the kinematic score.

The GNPM and MTTA scores differ only in kinematic terms, which are those depending upon \mathbf{x}_Δ or \mathbf{b} . These terms reveal the relationship of the GNPM/MTTA scores using $p(\mathbf{x}_\Delta)$ decomposed through Bayes law:

$$p(\mathbf{x}_\Delta) = \frac{p(\mathbf{x}_\Delta | \mathbf{b}) p(\mathbf{b})}{p(\mathbf{b} | \mathbf{x}_\Delta)}, \quad (20)$$

which is valid for any realization of \mathbf{b} .

C. GNPM and $p(\mathbf{x}_\Delta)$ Equivalence

In this section, we provide the relationship between the GNPM score of [16] and the distribution of the total errors. First, with algebraic manipulations (16) can be expressed as³

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{b} | \mathbf{x}_\Delta} &= \mathbf{P}_{\mathbf{b} \mathbf{x}_\Delta}^T \mathbf{Q}_{\mathbf{x}_\Delta}^{-1} \mathbf{x}_\Delta \\ &= \mathbf{R} \mathbf{H}^T (\mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}} + \mathbf{H} \mathbf{R} \mathbf{H}^T)^{-1} \mathbf{x}_\Delta \\ &= \mathbf{R} \mathbf{H}^T (\mathbf{I} + \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}^{-1} \mathbf{H} \mathbf{R} \mathbf{H}^T)^{-1} \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}^{-1} \mathbf{x}_\Delta \\ &= \mathbf{R} (\mathbf{I} + \mathbf{H}^T \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}} \mathbf{H} \mathbf{R})^{-1} \mathbf{H}^T \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}^{-1} \mathbf{x}_\Delta \\ &= (\mathbf{R}^{-1} + \mathbf{H}^T \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}^{-1} \mathbf{x}_\Delta. \end{aligned} \quad (21)$$

Recognizing that $\mathbf{H}^T \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}^{-1} \mathbf{H} = \sum_{i=1}^m \mathbf{S}_i^{-1}$, removal of the block form in (21) reveals an equivalence to the $\bar{\mathbf{x}}$ from

³An algebraic step here uses the relationship $(\mathbf{I} + \mathbf{P} \mathbf{Q})^{-1} \mathbf{P} = \mathbf{P} (\mathbf{I} + \mathbf{Q} \mathbf{P})^{-1}$ from traditional literature on the matrix inversion lemma [10].

[16]

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{b} | \mathbf{x}_\Delta} &= (\mathbf{R}^{-1} + \mathbf{H}^T \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}^{-1} \mathbf{x}_\Delta \\ &= \left(\mathbf{R}^{-1} + \sum_{i \in \mathcal{J}} \mathbf{S}_i^{-1} \right)^{-1} \sum_{i \in \mathcal{J}} (\mathbf{S}_i^{-1} \mathbf{x}_i^\Delta) = \bar{\mathbf{x}}, \end{aligned} \quad (22)$$

which is the bias estimate that maximizes the kinematic score for a given assignment hypothesis. We subsequently refer to $\boldsymbol{\mu}_{\mathbf{b} | \mathbf{x}_\Delta}$ as $\bar{\mathbf{x}}$, avoiding excessive use of subscripts and to clarify connections to previous literature. By inspection of (6) and (8) in [16], nomenclature translation allows the GNPM kinematic score to be written as

$$\begin{aligned} K_{GNPM} &= \frac{1}{\sqrt{|2\pi \mathbf{R}|}} e^{-\frac{1}{2} \bar{\mathbf{x}}^T \mathbf{R}^{-1} \bar{\mathbf{x}}} \\ &\quad \times \prod_{i \in \mathcal{J}} \frac{1}{\sqrt{|2\pi \mathbf{S}_i|}} e^{-\frac{1}{2} (\mathbf{x}_i^\Delta - \bar{\mathbf{x}})^T \mathbf{S}_i^{-1} (\mathbf{x}_i^\Delta - \bar{\mathbf{x}})} \\ &= \frac{1}{\sqrt{|2\pi \mathbf{R}|}} e^{-\frac{1}{2} \bar{\mathbf{x}}^T \mathbf{R}^{-1} \bar{\mathbf{x}}} \\ &\quad \times \frac{1}{\sqrt{|2\pi \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}|}} e^{-\frac{1}{2} (\mathbf{x}_\Delta - \mathbf{H} \bar{\mathbf{x}})^T \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}^{-1} (\mathbf{x}_\Delta - \mathbf{H} \bar{\mathbf{x}})}. \end{aligned} \quad (23)$$

Notice that the first term of (23) is (5) evaluated at $\mathbf{b} = \bar{\mathbf{x}}$ and the second term is (11), also evaluated at $\mathbf{b} = \bar{\mathbf{x}}$ by (14). Further observing from (22) and (12) that $p(\mathbf{b} | \mathbf{x}_\Delta)$ evaluated at $\mathbf{b} = \bar{\mathbf{x}}$ gives $1/\sqrt{|2\pi \mathbf{Q}_{\mathbf{b} | \mathbf{x}_\Delta}|}$, the relationship between K_{GNPM} and $p(\mathbf{x}_\Delta)$ is

$$\begin{aligned} p(\mathbf{x}_\Delta) &= \frac{p(\mathbf{x}_\Delta | \mathbf{b}) p(\mathbf{b})}{p(\mathbf{b} | \mathbf{x}_\Delta)} \Big|_{\mathbf{b} = \bar{\mathbf{x}}} \\ &= \frac{K_{GNPM}}{p(\mathbf{b} | \mathbf{x}_\Delta) \Big|_{\mathbf{b} = \bar{\mathbf{x}}}} = K_{GNPM} \sqrt{|2\pi \mathbf{Q}_{\mathbf{b} | \mathbf{x}_\Delta}|}. \end{aligned} \quad (24)$$

D. MTTA and $p(\mathbf{x}_\Delta)$ Equivalence

The derivation of MTTA in [20] began with GNPM, shown in the previous section to be $p(\mathbf{x}_\Delta | \mathbf{b})p(\mathbf{b})$, followed by integration of bias out of the score. Therefore, due to $\int p(\mathbf{x}_\Delta | \mathbf{b})p(\mathbf{b})d\mathbf{b} = p(\mathbf{x}_\Delta)$, we expect the MTTA likelihood to be equivalent to the distribution of the total errors. Here, we show the equivalence using the expansion of (20) about the point $\mathbf{b} = \mathbf{0}$. As a preliminary step, we rewrite (17) into an equivalent expression using the matrix inversion

lemma and removal of the block form,

$$\begin{aligned}
\mathbf{Q}_{\mathbf{b}|\mathbf{x}_\Delta} &= \mathbf{R} - \mathbf{P}_{\mathbf{b}|\mathbf{x}_\Delta}^T \mathbf{Q}_{\mathbf{x}_\Delta}^{-1} \mathbf{P}_{\mathbf{b}|\mathbf{x}_\Delta} \\
&= \mathbf{R} - \mathbf{R}\mathbf{H}^T (\mathbf{Q}_{\mathbf{x}_\Delta|\mathbf{b}} + \mathbf{H}\mathbf{R}\mathbf{H}^T)^{-1} \mathbf{H}\mathbf{R} \\
&= \left(\mathbf{R}^{-1} + \mathbf{H}^T \mathbf{Q}_{\mathbf{x}_\Delta|\mathbf{b}}^{-1} \mathbf{H} \right)^{-1} \\
&= \left(\mathbf{R}^{-1} + \sum_{i \in \mathcal{J}} \mathbf{S}_i^{-1} \right)^{-1}. \tag{25}
\end{aligned}$$

Upon nomenclature translation, the MTTA kinematic score as given for (8) in [20] is

$$K_{MTTA} = \frac{\sqrt{|2\pi\mathbf{V}|}}{\sqrt{\prod_{i \in \mathcal{J}^*} |2\pi\mathbf{S}_i|}} e^{-\frac{1}{2}\zeta}, \tag{26}$$

with

$$\mathbf{v} = \left(\sum_{i \in \mathcal{J}^*} \mathbf{S}_i^{-1} \right)^{-1}, \tag{27}$$

$$\zeta = \left(\sum_{i \in \mathcal{J}^*} (\mathbf{x}_i^\Delta)^T \mathbf{S}_i^{-1} \mathbf{x}_i^\Delta \right) - \mathbf{u}^T \mathbf{V} \mathbf{u}, \tag{28}$$

$$\mathbf{u} = \sum_{i \in \mathcal{J}^*} \mathbf{S}_i^{-1} \mathbf{x}_i^\Delta, \tag{29}$$

and the definitions $\mathbf{S}_0 = \mathbf{R}$, $\mathbf{x}_0^\Delta = \mathbf{0}$, and $\mathcal{J}^* = \{\mathcal{J}, 0\}$. From (22) and (25), we observe that $\mathbf{V} = \mathbf{Q}_{\mathbf{b}|\mathbf{x}_\Delta}$ and $\mathbf{u} = \mathbf{V}^{-1}\bar{\mathbf{x}}$, therefore

$$\begin{aligned}
\mathbf{u}^T \mathbf{V} \mathbf{u} &= \bar{\mathbf{x}}^T \mathbf{V}^{-1} \mathbf{V} \mathbf{V}^{-1} \bar{\mathbf{x}} \\
&= \bar{\mathbf{x}}^T \mathbf{V}^{-1} \bar{\mathbf{x}} \\
&= \bar{\mathbf{x}}^T \mathbf{Q}_{\mathbf{b}|\mathbf{x}_\Delta}^{-1} \bar{\mathbf{x}}, \tag{30}
\end{aligned}$$

and the full expansion of ζ can be rewritten as

$$\zeta = \left(\sum_{i \in \mathcal{J}} (\mathbf{x}_i^\Delta)^T \mathbf{S}_i^{-1} \mathbf{x}_i^\Delta \right) - \bar{\mathbf{x}}^T \mathbf{Q}_{\mathbf{b}|\mathbf{x}_\Delta}^{-1} \bar{\mathbf{x}}. \tag{31}$$

Substituting the expansions of \mathbf{V} , ζ , and rearranging terms in (26) to expose the specific Gaussian densities, we demonstrate the desired equivalency of MTTA and $p(\mathbf{x}_\Delta)$ following similar steps as in (23) and (24):

$$\begin{aligned}
K_{MTTA} &= \frac{\sqrt{|2\pi\mathbf{Q}_{\mathbf{b}|\mathbf{x}_\Delta}|}}{e^{-\frac{1}{2}(\bar{\mathbf{x}}^T \mathbf{Q}_{\mathbf{b}|\mathbf{x}_\Delta}^{-1} \bar{\mathbf{x}})}} \\
&\quad \times \prod_{i \in \mathcal{J}} \frac{1}{\sqrt{|2\pi\mathbf{S}_i|}} e^{-\frac{1}{2}(\mathbf{x}_i^\Delta)^T \mathbf{S}_i^{-1} \mathbf{x}_i^\Delta} \\
&\quad \times \frac{1}{\sqrt{|2\pi\mathbf{R}|}} \tag{32} \\
&= \left(\frac{1}{p(\mathbf{b}|\mathbf{x}_\Delta)} \times p(\mathbf{x}_\Delta|\mathbf{b}) \times p(\mathbf{b}) \right) \Bigg|_{\mathbf{b}=\mathbf{0}} \\
&= p(\mathbf{x}_\Delta).
\end{aligned}$$

E. Remarks on GNP Assignment Scores

Combining (24) and (32) gives the key result relating the kinematic scores and the distribution of the absolute errors

$$K_{MTTA} = K_{GNPM} \sqrt{|2\pi\mathbf{Q}_{\mathbf{b}|\mathbf{x}_\Delta}|} = p(\mathbf{x}_\Delta). \tag{33}$$

Consequently, although the MTTA formulation integrated the bias from the GNPM kinematic score, it inherently uses the same bias estimate that maximizes the association hypothesis as in GNPM. This result for bias mirrors the conclusions found for marginalizing the unknown target locations by Kaplan in [12] and [13]. In both cases, the difference between using the maximum likelihood value versus marginalizing reduces to a factor of the a-posteriori covariance. Furthermore, (33) implies the additional insight that bias estimation does not need to be a separate step in the calculation of a GNP score due to the equivalence with (10). In other words, combining the terms raised to the exponent in (23) gives an expression equivalent to $\mathbf{x}_\Delta^T \mathbf{Q}_{\mathbf{x}_\Delta}^{-1} \mathbf{x}_\Delta$. To solidify this result, we show the following equivalence algebraically in Appendix B:

$$\mathbf{x}_\Delta^T \mathbf{Q}_{\mathbf{x}_\Delta}^{-1} \mathbf{x}_\Delta = (\mathbf{x}_\Delta - \mathbf{H}\bar{\mathbf{x}})^T \mathbf{Q}_{\mathbf{x}_\Delta|\mathbf{b}}^{-1} (\mathbf{x}_\Delta - \mathbf{H}\bar{\mathbf{x}}) + \bar{\mathbf{x}}^T \mathbf{R}^{-1} \bar{\mathbf{x}} \tag{34}$$

which follows from the matrix inversion lemma along with several algebraic manipulations. A corollary of (34) is that $(\mathbf{x}_\Delta - \mathbf{H}\bar{\mathbf{x}})^T \mathbf{Q}_{\mathbf{x}_\Delta|\mathbf{b}}^{-1} (\mathbf{x}_\Delta - \mathbf{H}\bar{\mathbf{x}}) + \bar{\mathbf{x}}^T \mathbf{R}^{-1} \bar{\mathbf{x}}$ is a chi-square random variable with dn_a degrees of freedom, since $\mathbf{x}_\Delta^T \mathbf{Q}_{\mathbf{x}_\Delta}^{-1} \mathbf{x}_\Delta$ is a chi square random variable of dimension dn_a . This may not be immediately obvious at first glance, since with removal of the block form of the right hand side, (34) is the sum of $(n_a + 1)$ terms. In other words, degrees of freedom are lost through the estimation of $\bar{\mathbf{x}}$ with the data. An additional observation of (34) is that the left hand side is a function of \mathbf{R} , but the right hand side is a function of \mathbf{R}^{-1} . This allows various interpretations and simplifications if \mathbf{R} is assumed arbitrarily large or arbitrarily small.

As an additional remark, equating the normalization terms inherent in (33) gives

$$\frac{\sqrt{|2\pi\mathbf{Q}_{\mathbf{b}|\mathbf{x}_\Delta}|}}{\sqrt{|2\pi\mathbf{R}| \prod_{i \in \mathcal{J}} \sqrt{|2\pi\mathbf{S}_i|}}} = \frac{1}{\sqrt{|2\pi\mathbf{Q}_{\mathbf{x}_\Delta}|}}, \tag{35}$$

which after removal of the square roots and factoring out the constants, gives a simpler expression that relates the determinant terms of the structured matrices in the GNP problem:

$$|\mathbf{Q}_{\mathbf{x}_\Delta}| = \frac{|\mathbf{R}| \prod_{i \in \mathcal{J}} |\mathbf{S}_i|}{|\mathbf{Q}_{\mathbf{b}|\mathbf{x}_\Delta}|}. \tag{36}$$

F. GNP Costs

In this section, we provide the GNP costs in a form that includes the non-assignment gate with the same structure as found in [16] and [20]. Multiplying the joint posterior of (6) by the hypothesis-invariant term $\frac{1}{c} P_{AB}^{-m} \beta^{-n} P_{AB}^{(m-n)} (2\pi)^{d(m+1)/2} \sqrt{|\mathbf{R}|}$ gives⁴

$$\begin{aligned} & \Pr(\mathbf{h}, \mathbf{b} | \mathbf{x}_1^A, \dots, \mathbf{x}_m^A, \mathbf{x}_1^B, \dots, \mathbf{x}_n^B) \\ & \propto e^{-\frac{1}{2} \mathbf{b}^T \mathbf{R}^{-1} \mathbf{b}} \left(\frac{P_{AB}}{(2\pi)^{\frac{d}{2}} \beta P_{AB} P_{AB}} \right)^{-(m-n_a)} \\ & \times \prod_{i \in \mathcal{J}} \frac{1}{\sqrt{|\mathbf{S}_i|}} e^{-\frac{1}{2} (\mathbf{x}_i^A - \mathbf{b})^T \mathbf{S}_i^{-1} (\mathbf{x}_i^A - \mathbf{b})}. \end{aligned} \quad (37)$$

For a given hypothesis, the GNPM approach selects \mathbf{b} that maximizes the score, thus GNPM is (37) evaluated at $\mathbf{b} = \bar{\mathbf{x}}$. As a critical note, the results of [1] stress the use of unitless likelihood ratios. Since the first term in (37) is unitless and the units on the remaining terms are V^{n_a-m} and V^{-n_a} , respectively, where V is a unit hypersphere of the surveillance volume, the units of (37) are hypothesis invariant as V^{-m} . Therefore, hypotheses with varying numbers of assignments have the same units and one may safely use (37) within a specific GNP problem. However, if the GNPM cost is used in a higher context application, for example, in a sub-optimal solution for the association of more than two sensor data, care must be taken with the units of (37). We prefer to keep units in the score to be consistent with [16] and [20].

Taking the negative logarithm of (37) evaluated at $\mathbf{b} = \bar{\mathbf{x}}$ and multiplying by 2 gives the GNPM cost as

$$\begin{aligned} C_{GNPM}(\mathbf{h}) &= \bar{\mathbf{x}}^T \mathbf{R}^{-1} \bar{\mathbf{x}} \\ &+ 2(m - n_a) \log G_0 \\ &+ \sum_{i \in \mathcal{J}} \left[\log(|\mathbf{S}_i|) + (\mathbf{x}_i^A - \bar{\mathbf{x}})^T \mathbf{S}_i^{-1} (\mathbf{x}_i^A - \bar{\mathbf{x}}) \right], \end{aligned} \quad (38)$$

where

$$G_0 = \frac{P_{AB}}{(2\pi)^{d/2} \beta P_{AB} P_{AB}} \quad (39)$$

is the non-assignment gate value used in track-to-track assignment problems [5].⁵

Applying the equivalence from (33) and multiplying by the hypothesis-invariant term

⁴The determinant identity $\sqrt{|2\pi\boldsymbol{\Sigma}|} = (2\pi)^{d/2} \sqrt{|\boldsymbol{\Sigma}|}$ is used in (6) to allow the 2π term to be factored.

⁵A minor difference in the gate compared to previous literature is the density of false tracks, which we have taken as zero. For applications that need false target densities, we recommend using the gate value in [20], which is a trivial adjustment of (39). With false target densities as zero, the gate value in [20] is exactly (39).

$\frac{1}{c} P_{AB}^{-m} \beta^{-n} P_{AB}^{(m-n)} (2\pi)^{dm/2} \sqrt{|\mathbf{R}|}$ gives

$$\begin{aligned} & \Pr(\mathbf{h} | \mathbf{x}_1^A, \dots, \mathbf{x}_m^A, \mathbf{x}_1^B, \dots, \mathbf{x}_n^B) \\ & \propto \sqrt{|\mathbf{Q}_{\mathbf{b}|\mathbf{x}_\Delta}|} e^{-\frac{1}{2} \bar{\mathbf{x}}^T \mathbf{R}^{-1} \bar{\mathbf{x}}} \left(\frac{P_{AB}}{(2\pi)^{\frac{d}{2}} \beta P_{AB} P_{AB}} \right)^{-(m-n_a)} \\ & \times \prod_{i \in \mathcal{J}} \frac{1}{\sqrt{|\mathbf{S}_i|}} e^{-\frac{1}{2} (\mathbf{x}_i^A - \bar{\mathbf{x}})^T \mathbf{S}_i^{-1} (\mathbf{x}_i^A - \bar{\mathbf{x}})}, \end{aligned} \quad (40)$$

which has different units than GNPM through the determinant of the a-posteriori bias covariance. Converting to cost format the MTTA cost is

$$\begin{aligned} C_{MTTA}(\mathbf{h}) &= \bar{\mathbf{x}}^T \mathbf{R}^{-1} \bar{\mathbf{x}} - \log(|\mathbf{Q}_{\mathbf{b}|\mathbf{x}_\Delta}|) \\ &+ 2(m - n_a) \log G_0 \\ &+ \sum_{i \in \mathcal{J}} \left[\log(|\mathbf{S}_i|) + (\mathbf{x}_i^A - \bar{\mathbf{x}})^T \mathbf{S}_i^{-1} (\mathbf{x}_i^A - \bar{\mathbf{x}}) \right]. \end{aligned} \quad (41)$$

Note that (41) is not written exactly as was provided [20], but is equivalent through the result of (33) with hypothesis invariant terms removed.

G. Equivalence with GNN

Intuitively, the GNP problem in both the GNPM and MTTA form is expected to reduce to the classic GNN problem as $\mathbf{R} \rightarrow \mathbf{0}$. However, this does not directly follow from (38) due to the indeterminate 0/0 that arises in $\bar{\mathbf{x}}^T \mathbf{R}^{-1} \bar{\mathbf{x}}$. As shown in Appendix C, application of the key results of Section II.E avoids this issue and both the GNPM and MTTA costs reduce to GNN as $\mathbf{R} \rightarrow \mathbf{0}$, thus

$$C_{GNPM}(\mathbf{h})|_{\mathbf{R} \rightarrow \mathbf{0}} = C_{MTTA}(\mathbf{h})|_{\mathbf{R} \rightarrow \mathbf{0}} = C_{GNN}(\mathbf{h}), \quad (42)$$

where

$$\begin{aligned} C_{GNN}(\mathbf{h}) &= 2(m - n_a) \log G_0 \\ &+ \sum_{i \in \mathcal{J}} \left[\log(|\mathbf{S}_i|) + (\mathbf{x}_i^A)^T \mathbf{S}_i^{-1} (\mathbf{x}_i^A) \right]. \end{aligned} \quad (43)$$

Therefore, when \mathbf{R} is sufficiently small, a GNN algorithm is suitable since GNPM and MTTA effectively give the same answer as GNN, as demonstrated in [16].

III. PRACTICAL CONSIDERATIONS

In this section, we provide further insight into the cost differences and elaborate on the practical relevance. We begin with a discussion on behavior of non-assignment costs, and then conclude with a discussion on bias estimation within the costs.

A. Optimal Non-Assignment Costs

Motivated by solution algorithms, we prefer to think of the track assignment problem in an incremental cost

structure, which starts from no assignments and incrementally seeks additional assignments that lower the cost as in the algorithm of [17]. In light of this concept, by inspection of (43), the incremental cost of adding track tuple (i, h_i) to the assignment set in GNN is $(\mathbf{x}_i^\Delta)^T \mathbf{S}_i^{-1} (\mathbf{x}_i^\Delta) + \log(|\mathbf{S}_i|)$. Therefore, the optimal decision in GNN is to accept the assignment for consideration if the statistical distance (of d degrees of freedom) does not exceed the covariance-dependent threshold

$$(\mathbf{x}_i^\Delta)^T \mathbf{S}_i^{-1} (\mathbf{x}_i^\Delta) < 2 \log G_0 - \log(|\mathbf{S}_i|). \quad (44)$$

We interpret the physical meaning of the GNN assignment threshold as evaluating the probability that the tracks in the pair are on different targets that randomly appeared in the containment volume of the covariance, based on the a-priori spatial density that tracked targets may appear. Therefore, as the statistical distance of the pair increases, corresponding to a larger containment volume, it is more likely that one of the tracks is on a different target. Note that satisfying this inequality does not necessarily guarantee any particular assignment, as there may be other assignment pairs with lower cost. Once the best available assignment fails this inequality, no additional assignments may be added and all unassigned tracks remain as singletons.

Seeking an analogous threshold for the GNPM cost of (38) is challenging since the acceptance of a new track assignment adjusts the bias estimation within the hypothesis. Recalling the equivalence found in (22), algebraic manipulations give the expected value of the bias term $\bar{\mathbf{x}}^T \mathbf{R}^{-1} \bar{\mathbf{x}}$ in (38) as

$$\begin{aligned} & E[\bar{\mathbf{x}}^T \mathbf{R}^{-1} \bar{\mathbf{x}}] \\ &= E \left[\text{tr} \left(\bar{\mathbf{x}}^T \mathbf{R}^{-1} \bar{\mathbf{x}} \right) \right] \\ &= \text{tr} \left(\mathbf{R}^{-1} E[\bar{\mathbf{x}} \bar{\mathbf{x}}^T] \right) \\ &= \text{tr} \left(\mathbf{R}^{-1} E \left[\mathbf{P}_{\mathbf{b}_{\mathbf{x}_\Delta}}^T \mathbf{Q}_{\mathbf{x}_\Delta}^{-1} \mathbf{x}_\Delta \left(\mathbf{P}_{\mathbf{b}_{\mathbf{x}_\Delta}}^T \mathbf{Q}_{\mathbf{x}_\Delta}^{-1} \mathbf{x}_\Delta \right)^T \right] \right) \\ &= \text{tr} \left(\mathbf{R}^{-1} \mathbf{P}_{\mathbf{b}_{\mathbf{x}_\Delta}}^T \mathbf{Q}_{\mathbf{x}_\Delta}^{-1} \mathbf{Q}_{\mathbf{x}_\Delta} \mathbf{Q}_{\mathbf{x}_\Delta}^{-1} \mathbf{P}_{\mathbf{b}_{\mathbf{x}_\Delta}} \right) \\ &= \text{tr} \left(\mathbf{R}^{-1} \mathbf{R} \mathbf{H}^T \mathbf{Q}_{\mathbf{x}_\Delta}^{-1} \mathbf{H} \mathbf{R} \right) \\ &= \text{tr} \left(\mathbf{H}^T \left(\mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}} + \mathbf{H} \mathbf{R} \mathbf{H}^T \right)^{-1} \mathbf{H} \mathbf{R} \right) \\ &= \text{tr} \left(\mathbf{H}^T \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}^{-1} \left(\mathbf{I} + \mathbf{H} \mathbf{R} \mathbf{H}^T \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}^{-1} \right)^{-1} \mathbf{H} \mathbf{R} \right) \\ &= \text{tr} \left(\mathbf{H}^T \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}^{-1} \mathbf{H} \mathbf{R} \left(\mathbf{I} + \mathbf{H}^T \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}^{-1} \mathbf{H} \mathbf{R} \right)^{-1} \right) \quad (45) \\ &= \text{tr} \left(\mathbf{I} - \left(\mathbf{I} + \mathbf{H}^T \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}^{-1} \mathbf{H} \mathbf{R} \right)^{-1} \right) \end{aligned}$$

$$\begin{aligned} &= d - \text{tr} \left(\left(\mathbf{I} + \mathbf{H}^T \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}^{-1} \mathbf{H} \mathbf{R} \right)^{-1} \right) \\ &= d - \text{tr} \left(\left(\mathbf{I} + \sum_{i \in \mathcal{J}} \mathbf{S}_i^{-1} \mathbf{R} \right)^{-1} \right), \end{aligned}$$

which is limited to $[0, d]$ since each \mathbf{S}_i and \mathbf{R} are symmetric and positive definite matrices.⁶ Therefore, when the final term in (45) vanishes, the incremental cost of the i^{th} assignment, in an expected value sense, is completely contained in the $\log(|\mathbf{S}_i|) + (\mathbf{x}_i^\Delta - \bar{\mathbf{x}})^T \mathbf{S}_i^{-1} (\mathbf{x}_i^\Delta - \bar{\mathbf{x}})$ term. Thus, under this assumption and by inspection of (38), the analogous threshold from GNPM follows the same structure as GNN

$$(\mathbf{x}_i^\Delta - \bar{\mathbf{x}})^T \mathbf{S}_i^{-1} (\mathbf{x}_i^\Delta - \bar{\mathbf{x}}) < 2 \log G_0 - \log(|\mathbf{S}_i|), \quad (46)$$

which is a statistical distance of d degrees of freedom compared to a threshold that is dependent upon the covariance used in that statistical distance.

As discussed and demonstrated in [18], since GNPM follows the same threshold decision structure as GNN, G_0 is a nearly optimal gate for GNPM. Critically, the gate is optimal when the final term in (45) vanishes, which occurs after several assignments are made or after the first assignment when $\mathbf{R} \gg \mathbf{S}_i$. In [18], an optimal gate was provided for the case where only one assignment is made, but we do not recommend this in practice since intuitively the notion of a pattern match is only meaningful with multiple assignments.

The determination of incremental cost for MTTA is further complicated by the $\log(|\mathbf{Q}_{\mathbf{b}_{\mathbf{x}_\Delta}}|)$ term in (41), which introduces dependence upon the specific assignments made, including the incremental addition of tuple (i, h_i) . To allow an approximate analysis, we make the simplifying assumption that each $\mathbf{S}_i = \mathbf{S}$ (this condition is not required by GNPM or MTTA) and that enough assignments are made such that the final term in (45) vanishes. With these assumptions after n_a assignments are made, $\mathbf{Q}_{\mathbf{b}_{\mathbf{x}_\Delta}} = (\mathbf{R}^{-1} + \sum_{i \in \mathcal{J}} \mathbf{S}^{-1})^{-1} \approx (n_a \mathbf{S}^{-1})^{-1} = \mathbf{S}/n_a$. Given $n_a - 1$ assignments made before incrementally adding the tuple, an approximation of the incremental cost of the $-\log(|\mathbf{Q}_{\mathbf{b}_{\mathbf{x}_\Delta}}|)$ term is

$$-\log(|\mathbf{S}/n_a|) + \log(|\mathbf{S}/(n_a - 1)|) = \log \left(\frac{n_a}{n_a - 1} \right). \quad (47)$$

Therefore, for MTTA, the approximate incremental cost is $\log(|\mathbf{S}|) + (\mathbf{x}_i^\Delta - \bar{\mathbf{x}})^T \mathbf{S}^{-1} (\mathbf{x}_i^\Delta - \bar{\mathbf{x}}) + \log(n_a/(n_a - 1))$ and the analogous decision threshold is

$$\begin{aligned} & (\mathbf{x}_i^\Delta - \bar{\mathbf{x}})^T \mathbf{S}^{-1} (\mathbf{x}_i^\Delta - \bar{\mathbf{x}}) < 2 \log G_0 - \log(|\mathbf{S}|) \\ & \quad - \log \left(\frac{n_a}{n_a - 1} \right), \end{aligned} \quad (48)$$

⁶We use the relationships $(\mathbf{I} + \mathbf{P}\mathbf{Q})^{-1}\mathbf{P} = \mathbf{P}(\mathbf{I} + \mathbf{Q}\mathbf{P})^{-1}$ and $\mathbf{A}(\mathbf{I} + \mathbf{A})^{-1} = \mathbf{I} - (\mathbf{I} + \mathbf{A})^{-1}$ in (45).

which does not take the same form as the GNN threshold and G_0 is therefore not an optimal gate for MTTA. However, with many assignments, the $\log(n_a/(n_a - 1))$ term eventually vanishes and we expect G_0 to be nearly optimal for MTTA in problems with a high number of common targets.

The incremental cost of the i^{th} assignment is not the only mechanism of non-assignment behavior, the cost of the null hypothesis (i.e., the hypothesis of no assignments) also plays a significant role. Since the bias estimate in the null assignment is $\bar{\mathbf{x}}_0 = \mathbf{0}$ and recognizing from (25) that $\mathbf{Q}_{\mathbf{b}|\mathbf{x}_\Delta} = \mathbf{R}$ if there are no assignments, the GNPM and MTTA costs for the null hypothesis \mathbf{h}_0 are

$$C_{GNPM}(\mathbf{h}_0) = 2m \log G_0, \quad (49)$$

$$C_{MTTA}(\mathbf{h}_0) = 2m \log G_0 - \log(|\mathbf{R}|). \quad (50)$$

Therefore, with large \mathbf{R} , the null hypothesis in MTTA can dominate over other hypotheses. This trait is not present with GNPM, which can generally be expected to provide assignments using an arbitrarily large \mathbf{R} .

We demonstrate the analytical results for non-assignment behavior with numerical simulations. Consider a scenario where sensor \mathcal{A} observes six targets and sensor \mathcal{B} observes eight, with three targets in common. By the formula given in [16], a 6×8 track association problem has a total of 93,289 possible hypotheses. Assume that each of the 11 total targets are randomly generated in a hypersphere of dimension $d = 3$ with a uniform distribution, giving a target density of $\beta = 11$. These numbers are sufficient to evaluate the parameters in G_0 as $P_{AB} = 3/11$, $P_{\bar{A}\bar{B}} = 5/11$, and $P_{A\bar{B}} = 3/11$, and thus $G_0 = 0.0127$. Assume that the track covariances in each hypothesis satisfy $\mathbf{S}_i = \mathbf{S} = \sigma^2 \mathbf{I}$ and that the bias covariance satisfies $\mathbf{R} = \sigma_b^2 \mathbf{I}$. In Monte Carlo experiments, we evaluate the probability of correct association, which is the total number of correct entries in the most likely assignment vector \mathbf{h} as evaluated for the GNPM and MTTA costs. In the Monte Carlo experiments, a test gate, G_{test} , offset from the optimal gate of (39) is used in the cost functions and 10^4 Monte Carlo trials are performed for each G_{test} . The structure of these experiments is very similar to the numerical results of [23], which evaluated the fraction of correct assignments using various non-assignment thresholds.

In the first experiment, the Monte Carlo simulation varies σ while maintaining $\sigma_b = 5\sigma$, and these results are provided in Fig. 1. As shown, G_0 gives very close to optimal performance for GNPM, but a G_{test} slightly larger than G_0 gives maximal probability of correct association for MTTA. This illustrates the analytic result of (48) which, with several assumptions, predicts that G_0 is generally not an optimal gate for MTTA, particularly if there are few assignments made. The performance loss using G_0 for MTTA in this case is likely negligible as it causes less than a percentage point from maximal performance if that maximal performance is above 90%.

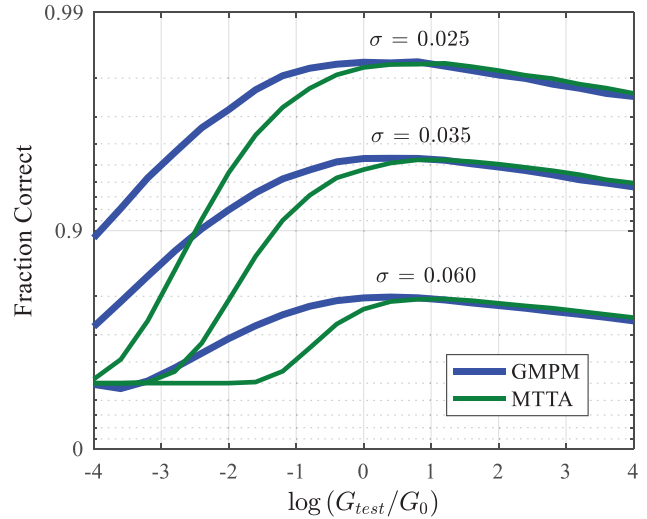


Fig. 1. Probability of correct track-to-track association for various covariance sizes. In this case, G_0 is a nearly optimal gate for GNPM.

In the next experiment, we maintain σ as the single value of 0.025, but set σ_b to values of 0.5σ , 5σ , and 60σ . The results are provided in Fig. 2, which illustrates that G_0 is not an optimal gate for MTTA when σ_b is large, while GNPM maintains G_0 as a nearly optimal gate. This illustrates the analytic result of (50), which states that the null hypothesis can dominate over other hypotheses if \mathbf{R} is large. As discussed in the derivation of (48) and (50), the effects of the $\log(|\mathbf{Q}_{\mathbf{b}|\mathbf{x}_\Delta}|)$ term in the MTTA cost cause performance loss with G_0 . However, by the key result of (33), any removal of the effects $\log(|\mathbf{Q}_{\mathbf{b}|\mathbf{x}_\Delta}|)$ cause in the incremental cost structure for MTTA effectively gives the GNPM cost.

Additionally, the result in Fig. 2 corresponding to the lowest σ_b illustrates (42), which states that GNPM and MTTA are equivalent as $\mathbf{R} \rightarrow \mathbf{0}$. As a final observation, maximal performance of both GNPM and MTTA reduces as σ_b grows. This is the intuitive result that

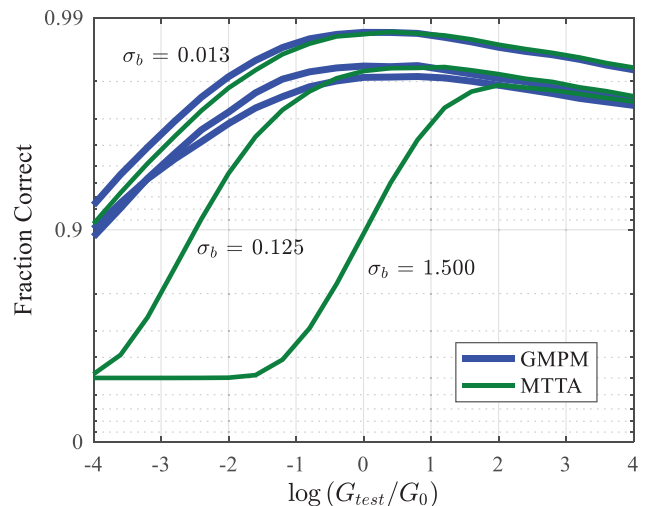


Fig. 2. Probability of correct track-to-track association for various σ_b values. With large σ_b , G_0 is not an optimal non-assignment gate for MTTA.

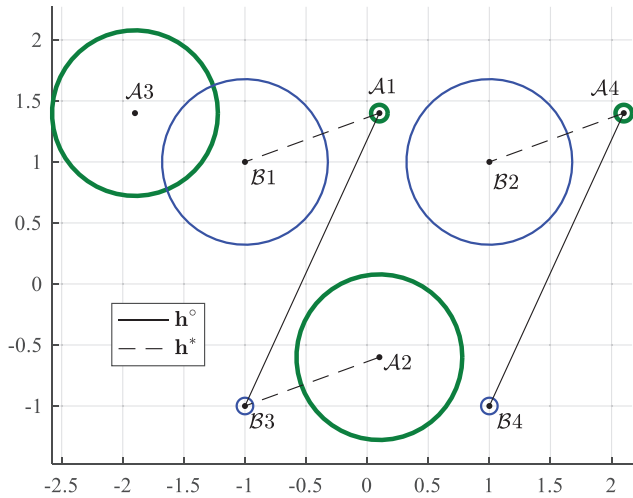


Fig. 3. 4×4 track assignment example. Targets 1 and 4 from Sensor \mathcal{A} and 3 and 4 from Sensor \mathcal{B} have a variable covariance $\sigma^2 \mathbf{I}$. All other tracks have covariance $0.1\mathbf{I}$. Circles represent 90% containment areas with $\sigma^2 = 10^{-3}$.

some track association performance is lost when bias is added to the GNN problem, and this was also reported in [16].

B. A-Posteriori Bias Covariance

As evidenced by (33), the difference between the GNPM and MTTA costs is in the a-posteriori bias covariance. Inspired by the example in [8], we use the 4×4 , $d = 2$ numerical example in Fig. 3 to illustrate the practical difference between the cost functions. A 4×4 track assignment problem gives 209 possible hypotheses. In this scenario, we let the covariance values of tracks 1 and 4 from Sensor \mathcal{A} and tracks 3 and 4 from Sensor \mathcal{B} vary from very high to very low values, but let the others maintain the value of $0.1\mathbf{I}$. If the variable covariances are large, the hypothesis of three assignments, $\mathbf{h}^* = [1 \ 3 \ 0 \ 2]$ (i.e., $\mathcal{A}1 \rightarrow \mathcal{B}1$, $\mathcal{A}2 \rightarrow \mathcal{B}3$, and $\mathcal{A}4 \rightarrow \mathcal{B}2$ as illustrated in Fig. 3), is preferable since the track states align and only a small shift is needed for the alignment. However, as the variable covariances shrink to very small size, the hypothesis of $\mathbf{h}^o = [3 \ 0 \ 0 \ 4]$ becomes more probable. In other words, given the a-priori assumptions that targets appear at random locations in the surveillance volume, the probability that the pattern difference $[(\mathcal{A}1 - \mathcal{A}4) - (\mathcal{B}3 - \mathcal{B}4)]^2 < \sigma^2$ occurs by random chance is essentially zero as $\sigma^2 \rightarrow 0$.

To illustrate the practical difference in the cost formulations, we find the track covariance size for GNPM and MTTA that gives \mathbf{h}^o as the definitive hypothesis. Provided in Fig. 4 is the a-posteriori bias covariance of the top hypothesis from the GNPM and MTTA costs. For this numerical experiment, we let $\mathbf{R} = \mathbf{I}$ and $G_0 = 19.2$. As shown, GNPM determines the definitive hypothesis with a larger σ than MTTA. This example illustrates that GNPM generally tends to prefer (and

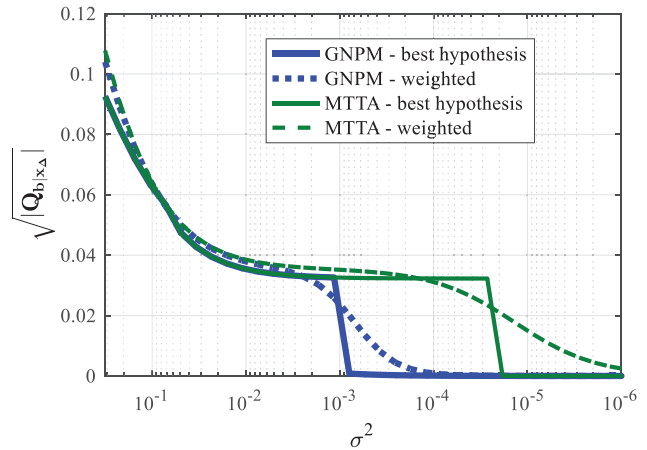


Fig. 4. A-posteriori bias covariance sizes of GNPM/MTTA hypotheses from the track sets in Fig. 3. GNPM determines \mathbf{h}^o as the best hypothesis near $\sigma^2 = 10^{-3}$ and MTTA near $\sigma^2 = 10^{-5}$.

score favorably) hypotheses that give smaller $|\mathbf{Q}_{\mathbf{b}|x_A}|$. Further illustrating this concept, we also provide the posterior-weighted $\sqrt{|\mathbf{Q}_{\mathbf{b}|x_A}|}$ in Fig. 4, using all 209 posteriors (from (37) and (40)) normalized to sum to unity. As shown, hypotheses with large a-posteriori bias covariance scored by GNPM have nearly zero weight as $\sigma^2 < 10^{-4}$, while MTTA maintains significant on those hypotheses.

IV. CONCLUDING REMARKS

GNP costs have their typical use in track-to-track association problems. Compared to traditional literature for track-to-track association, the GNP problem includes unknown sensor bias into the observation model. The two types of GNP costs discussed in this work are the GNPM and MTTA costs. GNPM involves the joint likelihood of both a hypothesis and a-posteriori bias estimate, while MTTA marginalizes bias from the problem. Here, we showed the intuitive result that the analytic difference between GNPM and MTTA kinematic scores is the determinant of the a-posteriori bias covariance. Several key insights arise through that result, including equivalences with the distribution of total errors and the role of bias estimation as a separate step in cost calculations. Leveraging this result, through an inspection of the GNPM incremental assignment cost, we argue that the non-assignment cost G_0 is nearly optimal for GNPM and demonstrate with numerical examples. However, through similar inspection of the MTTA incremental cost, G_0 is not optimal for MTTA and the significance diminishes for problems with many assignments but grows with large \mathbf{R} . Removal of the covariance of the a-posteriori bias from the MTTA non-assignment cost to give maximal probability of correct association effectively yields the GNPM cost. Therefore, if the goal of a GNP algorithm is to maximize the probability of correct association, we recommend GNPM. As a final experiment, through a simple two-dimensional exam-

ple, we show that GNPM tends to favor hypotheses with smaller a-posteriori bias covariance compared to MTTA. In conclusion, the results contained here expand upon previous literature to reveal important design considerations for specific track-to-track association problems.

APPENDIX A

Provided in this appendix is a derivation of the marginal and conditional densities for the random vectors within the track assignment problem. Explicitly writing \mathbf{x}_i^Δ from (7) to expose the noise terms gives

$$\begin{aligned}\mathbf{x}_i^\Delta &= \mathbf{x}_i^A - \mathbf{x}_{h_i}^B \\ &= \mathbf{x}_i + \mathbf{n}_i^A - (\mathbf{x}_i - \mathbf{b} + \mathbf{n}_{h_i}^B) \\ &= (\mathbf{n}_i^A - \mathbf{n}_{h_i}^B) + \mathbf{b},\end{aligned}\quad (51)$$

since, given \mathbf{h} is the correct hypothesis, each track is an observation of \mathbf{x}_i . Defining the combined noise term as $\mathbf{n}_i = (\mathbf{n}_i^A - \mathbf{n}_{h_i}^B)$ which is zero mean with covariance $\mathbf{S}_i = \mathbf{S}_{A,i} + \mathbf{S}_{B,h_i}$, we have

$$\mathbf{x}_i^\Delta = \mathbf{n}_i + \mathbf{b}.\quad (52)$$

Let $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_{n_a}]$ to be a length n_a vector that contains an ordering of the indices in \mathcal{J} , which, in other words, is simply a list of the track indices from sensor \mathcal{A} that are assigned to a track from sensor \mathcal{B} . Assuming all error terms are uncorrelated, the stacked vector of error terms is a normally distributed random vector with a block diagonal covariance, expressed as

$$\begin{bmatrix} \mathbf{n}_{\gamma_1} \\ \vdots \\ \mathbf{n}_{\gamma_{n_a}} \\ \mathbf{b} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{S}_{\gamma_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{S}_{\gamma_{n_a}} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{R} \end{bmatrix}\right).\quad (53)$$

Defining the stacked vector of error terms as

$$\mathbf{x}_\Delta = \begin{bmatrix} \mathbf{x}_{\gamma_1}^\Delta \\ \vdots \\ \mathbf{x}_{\gamma_{n_a}}^\Delta \end{bmatrix},\quad (54)$$

left multiplication of (53) by the transform matrix

$$\mathbf{V}_\Delta = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} & \mathbf{I} \end{bmatrix},\quad (55)$$

gives the distribution of absolute error between the observations as a normally distributed random vector

$$p(\mathbf{x}_\Delta) = \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\mathbf{x}_\Delta}),\quad (56)$$

with covariance

$$\begin{aligned}\mathbf{Q}_{\mathbf{x}_\Delta} &= \mathbf{V}_\Delta \begin{bmatrix} \mathbf{S}_{\gamma_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{S}_{\gamma_{n_a}} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{R} \end{bmatrix} \mathbf{V}_\Delta^T \\ &= \begin{bmatrix} \mathbf{S}_{\gamma_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_{\gamma_{n_a}} \end{bmatrix} + \begin{bmatrix} \mathbf{R} & \cdots & \mathbf{R} \\ \vdots & \ddots & \vdots \\ \mathbf{R} & \cdots & \mathbf{R} \end{bmatrix}.\end{aligned}\quad (57)$$

Next, we separate the distribution of absolute errors into conditional distributions. Defining the joint vector of absolute errors and bias as

$$\mathbf{x}_b = \begin{bmatrix} \mathbf{x}_\Delta \\ \mathbf{b} \end{bmatrix},\quad (58)$$

left multiplication of (53) by a similar transformation matrix

$$\mathbf{V} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{bmatrix},\quad (59)$$

gives the joint distribution of absolute error and bias as a zero mean normally distributed random vector

$$p(\mathbf{x}_b) = \mathcal{N}(\mathbf{0}, \mathbf{Q}),\quad (60)$$

with covariance written in block partition form

$$\begin{aligned}\mathbf{Q} &= \mathbf{V} \begin{bmatrix} \mathbf{S}_{\gamma_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{S}_{\gamma_{n_a}} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{R} \end{bmatrix} \mathbf{V}^T \\ &= \begin{bmatrix} \mathbf{Q}_{\mathbf{x}_\Delta} & \mathbf{P}_{\mathbf{b}\mathbf{x}_\Delta} \\ \mathbf{P}_{\mathbf{b}\mathbf{x}_\Delta}^T & \mathbf{R} \end{bmatrix}.\end{aligned}\quad (61)$$

The cross-correlation matrix is the block matrix

$$\mathbf{P}_{\mathbf{b}\mathbf{x}_\Delta} = \begin{bmatrix} \mathbf{R} \\ \vdots \\ \mathbf{R} \end{bmatrix}.\quad (62)$$

Applying the fundamental equations of linear estimation from [3] gives the conditional distributions according to both \mathbf{x}_Δ and \mathbf{b} . Defining a stacked matrix of identity matrices as $\mathbf{H} = [\mathbf{I} \dots \mathbf{I}]^T$, the conditional distribution of the absolute errors given the relative bias is a normally distributed random vector

$$p(\mathbf{x}_\Delta|\mathbf{b}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}_\Delta|\mathbf{b}}, \mathbf{Q}_{\mathbf{x}_\Delta|\mathbf{b}}), \quad (63)$$

with mean

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{x}_\Delta|\mathbf{b}} &= \mathbf{P}_{\mathbf{b}\mathbf{x}_\Delta} \mathbf{R}^{-1} \mathbf{b} \\ &= \mathbf{H}\mathbf{b}, \end{aligned} \quad (64)$$

and corresponding covariance

$$\begin{aligned} \mathbf{Q}_{\mathbf{x}_\Delta|\mathbf{b}} &= \mathbf{Q}_{\mathbf{x}_\Delta} - \mathbf{P}_{\mathbf{b}\mathbf{x}_\Delta} \mathbf{R}^{-1} \mathbf{P}_{\mathbf{b}\mathbf{x}_\Delta}^T \\ &= \mathbf{Q}_{\mathbf{x}_\Delta} - \mathbf{H} \mathbf{P}_{\mathbf{b}\mathbf{x}_\Delta}^T \\ &= \mathbf{Q}_{\mathbf{x}_\Delta} - \begin{bmatrix} \mathbf{R} & \dots & \mathbf{R} \\ \vdots & \ddots & \vdots \\ \mathbf{R} & \dots & \mathbf{R} \end{bmatrix} \\ &= \begin{bmatrix} S_{\gamma_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & S_{\gamma_{na}} \end{bmatrix}. \end{aligned} \quad (65)$$

Next we apply conditional density relationships to (60) to write the converse distribution. The conditional distribution of the bias given the absolute errors is a normally distributed random vector

$$p(\mathbf{b}|\mathbf{x}_\Delta) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{b}|\mathbf{x}_\Delta}, \mathbf{Q}_{\mathbf{b}|\mathbf{x}_\Delta}), \quad (66)$$

with mean and covariance

$$\boldsymbol{\mu}_{\mathbf{b}|\mathbf{x}_\Delta} = \mathbf{P}_{\mathbf{b}\mathbf{x}_\Delta}^T \mathbf{Q}_{\mathbf{x}_\Delta}^{-1} \mathbf{x}_\Delta, \quad (67)$$

$$\mathbf{Q}_{\mathbf{b}|\mathbf{x}_\Delta} = \mathbf{R} - \mathbf{P}_{\mathbf{b}\mathbf{x}_\Delta}^T \mathbf{Q}_{\mathbf{x}_\Delta}^{-1} \mathbf{P}_{\mathbf{b}\mathbf{x}_\Delta}, \quad (68)$$

thus completing the derivation of the desired probability distributions.

APPENDIX B

In this appendix, we algebraically show the equivalence of (34). To reduce cumbersome nomenclature, we drop the subscripts used in (34). Specifically, we establish the following equivalence:

$$\mathbf{x}^T \mathbf{Q}^{-1} \mathbf{x} = (\mathbf{x} - \mathbf{H}\mathbf{b})^T \mathbf{S}^{-1} (\mathbf{x} - \mathbf{H}\mathbf{b}) + \mathbf{b}^T \mathbf{R}^{-1} \mathbf{b}, \quad (69)$$

given $\mathbf{b} = (\mathbf{R}^{-1} + \mathbf{H}^T \mathbf{S}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{S}^{-1} \mathbf{x}$, $\mathbf{Q} = \mathbf{S} + \mathbf{H}\mathbf{R}\mathbf{H}^T$, and $\mathbf{H} = [\mathbf{I} \dots \mathbf{I}]^T$. Assume that all necessary matrix inverses exist.

Beginning with expression for \mathbf{b} , multiplication of both sides by $(\mathbf{R}^{-1} + \mathbf{H}^T \mathbf{S}^{-1} \mathbf{H})$ gives the useful preliminary relationship,

relationship,

$$\begin{aligned} (\mathbf{R}^{-1} + \mathbf{H}^T \mathbf{S}^{-1} \mathbf{H}) \mathbf{b} &= \mathbf{H}^T \mathbf{S}^{-1} \mathbf{x} \\ \mathbf{R}^{-1} \mathbf{b} + \mathbf{H}^T \mathbf{S}^{-1} \mathbf{H} \mathbf{b} &= \mathbf{H}^T \mathbf{S}^{-1} \mathbf{x} \\ \mathbf{R}^{-1} \mathbf{b} &= \mathbf{H}^T \mathbf{S}^{-1} \mathbf{x} - \mathbf{H}^T \mathbf{S}^{-1} \mathbf{H} \mathbf{b}. \end{aligned} \quad (70)$$

Application of the matrix inversion lemma to \mathbf{Q}^{-1} gives

$$\begin{aligned} \mathbf{Q}^{-1} &= (\mathbf{S} + \mathbf{H}\mathbf{R}\mathbf{H}^T)^{-1} \\ &= \mathbf{S}^{-1} - \mathbf{S}^{-1} \mathbf{H} (\mathbf{R}^{-1} + \mathbf{H}^T \mathbf{S}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{S}^{-1}, \end{aligned} \quad (71)$$

therefore, the full chi-square term can be written as

$$\begin{aligned} \mathbf{x}^T \mathbf{Q}^{-1} \mathbf{x} &= \mathbf{x}^T \left[\mathbf{S}^{-1} - \mathbf{S}^{-1} \mathbf{H} (\mathbf{R}^{-1} + \mathbf{H}^T \mathbf{S}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{S}^{-1} \right] \mathbf{x} \\ &= \mathbf{x}^T \mathbf{S}^{-1} \mathbf{x} - \mathbf{x}^T \mathbf{S}^{-1} \mathbf{H} (\mathbf{R}^{-1} + \mathbf{H}^T \mathbf{S}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{S}^{-1} \mathbf{x}. \end{aligned} \quad (72)$$

Since the expression for \mathbf{b} appears in (72), we have

$$\mathbf{x}^T \mathbf{Q}^{-1} \mathbf{x} = \mathbf{x}^T \mathbf{S}^{-1} \mathbf{x} - \mathbf{x}^T \mathbf{S}^{-1} \mathbf{H} \mathbf{b}. \quad (73)$$

Recognizing that (73) is a portion of the quadratic expansion of $(\mathbf{x} - \mathbf{H}\mathbf{b})^T \mathbf{S}^{-1} (\mathbf{x} - \mathbf{H}\mathbf{b})$, rewriting to include the addition of terms that complete the quadratic expansion gives

$$\begin{aligned} \mathbf{x}^T \mathbf{Q}^{-1} \mathbf{x} &= (\mathbf{x} - \mathbf{H}\mathbf{b})^T \mathbf{S}^{-1} (\mathbf{x} - \mathbf{H}\mathbf{b}) \\ &\quad + \mathbf{b}^T (\mathbf{H}^T \mathbf{S}^{-1} \mathbf{x} - \mathbf{H}^T \mathbf{S}^{-1} \mathbf{H} \mathbf{b}). \end{aligned} \quad (74)$$

Substituting (70) into (74) gives the desired equivalency of (69).

APPENDIX C

In this appendix, we establish the equivalence between the GNPM, MTTA, and GNN costs as $\mathbf{R} \rightarrow \mathbf{0}$. Unfortunately, direct substitution of $\mathbf{R} = \mathbf{0}$ into the GNPM and MTTA costs of (38) and (41) gives indeterminate terms. Applying the key results of Section II.E avoids this issue, and allows simplification to the GNN cost. Converting the chi-square terms of GNPM into a block structure followed by application of (36) gives the following equivalence:

$$\begin{aligned} \bar{\mathbf{x}}^T \mathbf{R}^{-1} \bar{\mathbf{x}} + \sum_{i \in \mathcal{J}} (\mathbf{x}_i^\Delta - \bar{\mathbf{x}})^T \mathbf{S}_i^{-1} (\mathbf{x}_i^\Delta - \bar{\mathbf{x}}) \\ &= \bar{\mathbf{x}}^T \mathbf{R}^{-1} \bar{\mathbf{x}} + (\mathbf{x}_\Delta - \mathbf{H}\bar{\mathbf{x}})^T \mathbf{Q}_{\mathbf{x}_\Delta|\mathbf{b}}^{-1} (\mathbf{x}_\Delta - \mathbf{H}\bar{\mathbf{x}}) \\ &= \mathbf{x}_\Delta^T \mathbf{Q}_{\mathbf{x}_\Delta}^{-1} \mathbf{x}_\Delta \end{aligned} \quad (75)$$

By inspection of (13), $\mathbf{Q}_{\mathbf{x}_\Delta} = \mathbf{Q}_{\mathbf{x}_\Delta|\mathbf{b}}$ if $\mathbf{R} = \mathbf{0}$, therefore, the limiting form of GNPM can be

written as

$$\begin{aligned}
C_{GNPM}(\mathbf{h})|_{\mathbf{R} \rightarrow \mathbf{0}} &= \mathbf{x}_\Delta^T \mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}^{-1} \mathbf{x}_\Delta + 2(m - n_a) \log G_0 \\
&\quad + \sum_{i \in \mathcal{J}} \log(|\mathbf{S}_i|) \\
&= 2(m - n_a) \log G_0 \\
&\quad + \sum_{i \in \mathcal{J}} \left[\log(|\mathbf{S}_i|) + (\mathbf{x}_i^\Delta)^T \mathbf{S}_i^{-1} (\mathbf{x}_i^\Delta) \right] \\
&\equiv C_{GNN}(\mathbf{h}), \tag{76}
\end{aligned}$$

which establishes the desired equivalence of GNPM with GNN. Evaluating MTTA as $\mathbf{R} \rightarrow \mathbf{0}$ involves the additional complication of $\mathbf{Q}_{\mathbf{b} | \mathbf{x}_\Delta}$, which includes \mathbf{R}^{-1} by inspection of (25). Rearranging (36) gives

$$\frac{|\mathbf{Q}_{\mathbf{b} | \mathbf{x}_\Delta}|}{|\mathbf{R}|} = \frac{\prod_{i \in \mathcal{J}} |\mathbf{S}_i|}{|\mathbf{Q}_{\mathbf{x}_\Delta}|}, \tag{77}$$

which for $\mathbf{R} \rightarrow \mathbf{0}$ can be reduced to

$$\frac{\prod_{i \in \mathcal{J}} |\mathbf{S}_i|}{|\mathbf{Q}_{\mathbf{x}_\Delta | \mathbf{b}}|} = \frac{\prod_{i \in \mathcal{J}} |\mathbf{S}_i|}{\prod_{i \in \mathcal{J}} |\mathbf{S}_i|} = 1. \tag{78}$$

Therefore, by reintroducing the hypothesis-invariant term $|\mathbf{R}|$ into (40) and applying (75) and (78) gives the MTTA cost as the same functional form of (76) since $\log(1) = 0$, which establishes $C_{MTTA}(\mathbf{h})|_{\mathbf{R} \rightarrow \mathbf{0}} = C_{GNN}(\mathbf{h})$.

REFERENCES

- [1] Y. Bar-Shalom, S. Blackman, and R. Fitzgerald
“Dimensionless score function for multiple hypothesis tracking,”
IEEE Trans. Aerosp. Electron. Syst., vol. 43, no. 1, pp. 392–400, 2007.
- [2] Y. Bar-Shalom and H. Chen
“Multisensor track-to-track association for tracks with dependent errors,”
J. Adv. Inf. Fusion, vol. 1, no. 1, pp. 3–14, Jul. 2006.
- [3] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation.
New York, NY, USA: John Wiley and Sons, 2001.
- [4] Y. Bar-Shalom, P. K. Willett, and X. Tian
Tracking and Data Fusion: A Handbook of Algorithms.
Storrs, CT, USA: YBS Publishing, 2011.
- [5] S. S. Blackman and R. Popoli
Design and Analysis of Modern Tracking Systems. Norwood,
MA, USA: Artech House, 1999.
- [6] C. Y. Chong, S. Mori, and K. C. Chang
“Distributed multitarget multisensor tracking,”
in *Multitarget Multisensor Tracking: Advanced Applications*.
Boston, MA, USA: Artech House, 1990, pp. 247–295.
- [7] S. Danford, B. Kragel, and A. Poore
“Joint MAP bias estimation and data association:
algorithms,”
in *Proc. SPIE, Signal and Data Process. Small Targets*, vol.
6699, 2007.
- [8] J. Ferry
“Exact bias removal for the track-to-track association
problem,”
in *Proc. 12th Int. Conf. Inf. Fusion*, Seattle, WA, USA, 2009.

- [9] J. Ferry
“Exact association probability for data with bias and
features,”
J. Adv. Inf. Fusion, vol. 5, no. 1, pp. 41–67, 2010.
- [10] H. V. Henderson and S. R. Searle
“On deriving the inverse of a sum of matrices,”
SIAM Review, vol. 23, no. 1, pp. 53–60, 1981.
- [11] M. B. Hurley
Track association with Bayesian probability theory, MIT
Lincoln Laboratory,
Tech. Rep., Lexington, MA, USA, 2003.
- [12] L. M. Kaplan, Y. Bar-Shalom, and W. D. Blair
“Assignment costs for multiple sensor track-to-track
association,”
IEEE Trans. Aerosp. Electron. Syst., vol. 44, no. 2, pp.
655–677, 2008.
- [13] L. M. Kaplan and W. D. Blair
“Assignment costs for multiple sensor track-to-track
association,”
in *Proc. 7th Int. Conf. Inf. Fusion*, Stockholm, Sweden, 2004.
- [14] B. Kragel, S. Danford, and A. Poore
“Concurrent MAP data association and absolute bias
estimation with an arbitrary number of sensors,”
in *Proc. SPIE, Signal and Data Process. Small Targets*,
vol. 6969, 2008.
- [15] B. D. Kragel, S. M. Herman, and N. J. Roseveare
“Efficiency and sensitivity of methods for assessing
ambiguity in data association decisions,”
in *Proc. SPIE, Signal and Data Process. Small Targets*,
Orlando, FL, USA, 2008.
- [16] M. Levedahl
“An explicit pattern matching assignment algorithm,”
in *Proc. SPIE, Signal and Data Process. Small Targets*,
vol. 4728, pp. 461–469, 2002.
- [17] M. Levedahl
Method and system for assigning observations.
U.S. Patent 7092924, 28 February 2002.
- [18] M. Levedahl and J. D. Glass
“Optimal non-assignment costs for the GNP problem,”
in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, 2020.
- [19] X. Lin, T. Kirubarajan, and Y. Bar-Shalom
“Multisensor bias estimation with local tracks without a
priori association,”
in *Proc. SPIE, Signal and Data Process. Small Targets*,
vol. 5204, Bellingham, 2003.
- [20] D. J. Papageorgiou and M. Holender
“Track-to-track association and ambiguity management in
the presence of sensor bias,”
J. Adv. Inf. Fusion, vol. 6, no. 2, pp. 77–100, 2011.
- [21] D. J. Papageorgiou and J. D. Sergi
“Simultaneous track-to-track association and bias removal
using multistart local search,”
in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, 2008.
- [22] K. R. Pattipati, T. Kirubarajan, and R. L. Popp
“Survey of assignment techniques for multitarget tracking,”
in *Multitarget-Multisensor Tracking: Applications and
Advances*, vol. 3, Norwood, MA, USA: Artech House, 2000,
pp. 77–159.
- [23] L. Stone, T. M. Tran, and M. Williams
“Improvement in track-to-track association from using an
adaptive threshold,”
in *Proc. 12th Int. Conf. Inf. Fusion*, Seattle, WA, USA, 2009.
- [24] L. D. Stone, M. L. Williams, and T. M. Tran
“Track to track association and bias removal,”
in *Proc. SPIE, Signal and Data Process. Small Targets*,
vol. 4728, pp. 315–329, 2002.



Mark Levedahl is a Raytheon Engineering Fellow. He has a bachelor's degree in mechanical engineering from McGill University and has been employed by Raytheon since 1991. Mr. Levedahl has more than 40 years of experience working in systems engineering, guidance, navigation, and control-system development, battle management, multi-target multi-sensor tracking, and simulation primarily related to missile defense.



John D. Glass is a Systems Engineer with Raytheon Technologies. In 2009, he graduated from the University of Tennessee with a bachelor of science in electrical engineering, and in 2010 at Georgia Tech with a Master of science in electrical and computer engineering. In May 2015, he completed the Ph.D. degree in electrical and computer engineering at Georgia Tech, focusing on monopulse processing and target tracking. From 2013 to 2017, Dr. Glass was a Research Engineer at Georgia Tech Research Institute working on radar simulation and multi-sensor fusion analysis. From 2011 to 2017, Dr. Glass was a Member of the Editorial Board for the Aerospace and Electronic Systems Magazine as Associate Editor for Student Research, recruiting and handling student highlight articles. Currently, Dr. Glass is a Session Organizer for the IEEE AEROSPACE CONFERENCE. His research interests include target tracking, multisensor fusion, and detection/estimation applied to radar.

A Constrained POMDP Formulation and Algorithmic Solution for Radar Resource Management in Multi-Target Tracking

MAX IAN SCHÖPE
HANS DRIESSEN
ALEXANDER G. YAROVY

The radar resource management problem in a multitarget tracking scenario is considered. The problem is solved using a dynamic budget balancing algorithm. It models the different sensor tasks as partially observable Markov decision processes and solves them by applying a combination of Lagrangian relaxation and policy rollout. The algorithm has a generic architecture and can be applied to different radar or sensor systems and cost functions. This is shown through simulations of two-dimensional tracking scenarios. Moreover, it is demonstrated how the algorithm allocates the sensor time budgets dynamically to a changing environment in a nonmyopic fashion. Its performance is compared with different resource allocation techniques and its computational load is investigated with respect to several input parameters.

Manuscript received October 16, 2020; revised February 4, 2021; released for publication June 30, 2021.

Associate Editor: Jason Williams.

The authors are with the Microwave Sensing, Signals and Systems (MS3), Delft University of Technology, 2628 CD Delft, The Netherlands (E-mail: {m.i.schope; j.n.driessen; a.yarovoy}@tudelft.nl).

This research was partially funded by the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) through the Integrated Cooperative Automated Vehicles (i-CAVE) project.

1557-6418/21/\$17.00 © 2021 JAIF

I. INTRODUCTION

Recent advances in multifunction radar (MFR) systems led to an increase of their degrees of freedom. As a result, modern MFR systems are capable of adjusting many parameters during runtime. An automatic adaptation of the radar system to changing situations, like weather conditions, interference, or target maneuvers, is often mentioned in the context of MFR and is usually called radar resource management (RRM). It is frequently considered within the broader context of so-called cognitive radar (see, e.g. [7], [10], [15], [19], [26]). Possible applications of these management approaches include automotive scenarios such as autonomous driving or traffic monitoring, (maritime) surveillance, and air traffic control. This paper aims at developing a generic framework and approximately optimal algorithmic solutions for solving RRM problems. Although the focus is on MFR, the approach is not limited to such sensor systems and has a wider applicability.

A. Radar Resource Management

Much of the research on RRM (see e.g. the overview by Hero and Cochran in [21] or by Moo and Ding in [34]) focuses on a single task, e.g. keeping a constant track quality even under target maneuvers. This usually means managing the time budget spent on a certain task. However, MFR systems are usually operating at their sensor time and/or energy budget limit. In such cases, increasing the budget for one task means simultaneously decreasing the budget of the others, inevitably deteriorating their performance. In this paper, part of the RRM problem is therefore seen as a budget or resource balancing act over the individual tasks.

Heuristic solutions have been presented in the past (see, for instance, the overview in [24]), some relying on assigning task priorities and priority-based scheduling. Applying heuristics too early in the design leads to complicated solutions, e.g. nested if-then-else rules. It is not easy to understand what problem is solved within those approaches and whether or not and in what sense the solution is optimal. This usually does not lead to a reusable generic algorithm. In addition, a priority-based scheduler usually does not balance the budget over all tasks but simply schedules the jobs in order of priority (as, e.g. applied in [33] and [39]). When the timeline is fully occupied, it often leaves a set of tasks with the lowest priorities that together do not fit anymore. These approaches do not consider decreasing the time budgets of individual tasks. Furthermore, the determination of the levels of priorities and the rules for assigning them is often not easy and prone to heuristics.

In this paper, the problem is treated as an optimal stochastic control problem. This relies on an explicit formulation of:

- the inference problem that the radar has to solve in terms of dynamic and measurement models,

- the control actions that are available to the sensor, which reflect the degrees of freedom of the MFR mentioned earlier,
- a cost function that reflects the system performance that the user would like to optimize.

To the best of our knowledge, an overall solution to the RRM problem using this approach has not been presented so far. It has been suggested that a truly optimal solution could possibly lead to a significant improvement of the performance of adaptive sensors [20], but that still needs to be illustrated. However, even if the performance would not improve much over heuristic solutions that are carefully tuned to each application, a reusable generic framework will reduce the design effort of RRM solutions. As a consequence, such a framework would reduce the development cost and time and aid in understanding the system behavior.

B. Markov Decision Processes in Resource Management

Markov decision processes (MDPs) and partially observable MDPs (POMDPs) are attractive frameworks for modeling and solving RRM problems. They use a number of states to formulate a dynamic control problem in which the optimal actions can be found through optimizing a cost or reward function. A very good overview of how these schemes can be used for RRM has been published by Charlish et al. in [12].

Those frameworks have been applied to single tasks, for instance, by Charlish and Hoffmann in [13] or by Krishnamurthy in [29]. Both methods optimize the time between consecutive measurement operations. Charlish and Hoffmann are considering a radar tracking example, where the track quality needs to be optimized while Krishnamurthy presents a more general sensor scenario where the measurement performance is optimized regarding false-alarm rate and the quality of the estimate. The former approach applied policy rollout, while the latter used a stochastic dynamic programming algorithm. Two other approaches show how radar actions can be determined by applying reinforcement learning (RL) [38] and deep RL [42] to solve an underlying MDP. In their papers, both Selvi et al. and Thornton et al. are optimizing the sensing strategies for a single target while a communication signal is using the same frequency band. Both publications show that the optimal policy can improve the performance despite the presence of the interferer. We believe that RL is an interesting approach to RRM but that it is often not feasible because of the huge state space that comes with many problem formulations. In such a case, the training of the algorithm would need an enormous amount of data and a lot of computation time.

Constrained (PO)MDPs have been proposed to solve multitask control problems, where the constraint(s) among others can represent the limit on the available resources or budgets for all the tasks. Possi-

ble applications are radar networks or single radars with multiple tasks. The computational complexity of these problems is potentially large. It has been suggested to decouple the main optimization problem into smaller and easier-to-solve subproblems by the use of Lagrangian relaxation (LR). One LR approach for sensor networks with an energy constraint on the inter-sensor communication has been published by Williams et al. in [45]. Some notable LR approaches for multitask radar scenarios are, e.g. [46] by Wintenby and Krishnamurthy and [44] by White and Williams. Wintenby and Krishnamurthy apply a Markov chain consisting of performance states for each tracking task and solve it with a combination of LR and approximate dynamic programming. White and Williams assume a discretized state space and a fully observable MDP, which they solve by the use of dynamic programming. In addition to that, Castañón applies LR in combination with a constrained POMDP for multiobject classification in [11]. The chosen POMDP solution method in that approach is the so-called Witness algorithm. Similar to LR, one could also consider the quality of service resource allocation method (Q-RAM) in combination with POMDPs. Although Q-RAM requires an action-space discretization while LR allows the subproblems to be solved analytically, these methods are conceptually very similar. Some interesting approaches using Q-RAM have been shown by Irci et al. in [22] and Charlish et al. in [16] and [14].

Another interesting approach for applying POMDPs for RRM has been introduced by Krishnamurthy and Djonin in [30], where they divide the RRM algorithm into “sensor micromanagement” and “sensor macromanagement.” The former is formulated as a POMDP and determines after which time the resource allocation has to be updated. There is always one task that gets a high resource allocation, while the others receive a lower one. The macromanagement, on the other hand, decides which target will get the highest priority and therefore the highest resource allocation. This process is based on the realized cost of the micromanagement and some heuristic rules. Our research, on the other hand, aims at combining micro- and macromanagement. The resource distribution is defined directly through the cost function and without any heuristic functions. In addition to that, the budgets of the tasks can gradually change over time contrarily to the approach of Krishnamurthy and Djonin, where only two different actions exist.

C. The Cost Function

When applying such an RRM approach, the final performance of the sensor system will be determined by the cost function. This is preferred over a heuristic approach; however, it introduces the explicit formulation of such a cost function in the application of the framework. Sometimes it has been suggested that generic measures of performance, such as the information gain, or the Renyi divergence applied to the posterior density of the full state

could be applied (see, e.g. [28], [43]). It is our strong conviction that one single cost function will not meet the desires of different users in different applications with different sensors, targets, and environments.

The development of specific cost functions is important and will be a development task in itself that will require close cooperation with potential users. However, since the primary focus in this paper is on the development of a generic framework for RRM, the development of such user-specific cost functions is out of the scope.

D. Our Approach

In this paper, the RRM problem is considered as a multitask time budget constrained control problem, where the individual tasks are different tracking tasks. Our chosen problem formulation directly leads to the assumption of a constrained POMDP.

Previously, we have already shown the optimal balancing of sensor budgets in a linear time-invariant (LTI) setting by using the optimal steady-state budget balancing (OSB) algorithm [36]. It applies LR to distribute the resources over the different tasks. We have subsequently considered generic dynamical problems by utilizing the POMDP framework and introduced the approximately optimal dynamic budget balancing (AODB) algorithm [37] with a cost function based on the predicted error-covariance of the Kalman filter (KF). We have shown that the results of the AODB algorithm are approximately optimal with respect to the steady-state error-covariance of a KF. The RRM problem was solved non-myopically by using an online Monte Carlo technique called policy rollout, which stochastically predicts the future.

E. Novelty

In this paper, we compare the performance of the AODB algorithm to several other resource allocation techniques. Furthermore, we investigate its computational load. Compared to our previous papers, we apply the AODB algorithm to a more complete dynamic radar tracking scenario to emphasize its practical value in variable problem settings. In order to do so, we show how the AODB algorithm can be applied to dynamic radar scenarios assuming different measurement types and system parameters.

F. Structure of the Paper

The remainder of this paper is structured as follows. Section II defines the problem as a constrained optimization problem in a POMDP framework, while Section III explains the general application of LR and policy rollout to that problem. Section IV introduces the assumed radar scenario. In Section V, we compare the results of the OSB and the AODB algorithm in a simplified LTI scenario, similar to our work in [37]. In Section VI, we

assume a dynamic radar-related scenario, with more realistic parameters than in our previous work. It is solved by applying the AODB algorithm and optimizing both dwell time and revisit interval. Subsequently, an analysis of the algorithm's performance and its computational load is conducted in Sections VII and VIII, respectively. Finally, Section IX contains the conclusions.

II. RRM PROBLEM DEFINITION

A. Motion Model

At every moment in time t , each target considered within this model can be characterized by a state based on its position and velocity in the x and y directions within a two-dimensional Cartesian coordinate system. For target n , this state is defined as

$$\mathbf{s}_t^n = [x_t^n \quad y_t^n \quad \dot{x}_t^n \quad \dot{y}_t^n]^T, \quad (1)$$

where x_t^n, y_t^n and \dot{x}_t^n, \dot{y}_t^n are the position and velocity of target n in x and y , respectively. The future target state at time $t + \Delta t$ can be calculated following a function

$$\mathbf{s}_{t+\Delta t}^n = f_{\Delta t}(\mathbf{s}_t^n, \mathbf{w}_t^n), \quad (2)$$

where $\mathbf{s}_{t+\Delta t}^n$ is the next following state at time $t + \Delta t$ and $\mathbf{w}_t^n \in \mathbb{R}^4$ is the maneuverability noise for target n at time t . The state evolution equation (2) directly defines the evolution probability density function, which is given as

$$p(\mathbf{s}_{t+\Delta t}^n | \mathbf{s}_t^n). \quad (3)$$

B. Measurement Model

We assume a sensor that is taking noisy observations of the state \mathbf{s}_t^n with sensor action $\mathbf{a}_t^n \in \mathbb{R}^m$, where m is the amount of adjustable action parameters. A measurement \mathbf{z}_t^n of target n at time t can be characterized by using the measurement function \mathfrak{h} as

$$\mathbf{z}_t^n = \mathfrak{h}(\mathbf{s}_t^n, \mathbf{v}_t^n, \mathbf{a}_t^n), \quad (4)$$

where $\mathbf{v}_t^n \in \mathbb{R}^q$ is the measurement noise for target n and q is the amount of measurement parameters. The measurement equation (4) directly defines the measurement probability density function, which can be written as

$$p(\mathbf{z}_t^n | \mathbf{s}_t^n, \mathbf{a}_t^n). \quad (5)$$

C. Tracking Algorithm

For the tracking scenarios considered in this paper, a tracking algorithm should be chosen that aims at computing the posterior density. For linear systems, a KF can be adopted as an exact solution. For nonlinear systems, possible algorithms are an extended KF (EKF) or a particle filter, for example.

D. Budget Optimization Problem

As mentioned in Section I, the radar sensor is assumed to have a limited maximum budget Θ_{\max} of any kind. For action \mathbf{a}_t^n that is executed for each task n , a certain amount of budget (e.g. time or energy allocations) is required. In an overload situation, the current tasks require more total budget than is available. Thus, the available budget has to be distributed over the tasks in a way that minimizes a cost (e.g. related to the uncertainty of the current situation).

At time t , the optimization problem for N different tasks can be written as

$$\begin{aligned} & \underset{\mathbf{a}_t}{\text{minimize}} && \sum_{n=1}^N c(\mathbf{a}_t^n, \mathbf{s}_t^n) \\ & \text{subject to} && \sum_{n=1}^N \Theta_t^n(\mathbf{a}_t^n) \leq \Theta_{\max}, \end{aligned} \quad (6)$$

where $\Theta_t^n \in [0, 1]$ is the budget for task n at time t , $c(\cdot)$ is the used cost function, and $\Theta_{\max} \in [0, 1]$ is the maximum available budget (0: no budget assigned, 1: all budget assigned). The definition of an operationally relevant cost function is important to efficiently benefit from these techniques, but is not the focus of this paper. An example of an operationally relevant cost function has been discussed by Katsilieris et al. [25].

III. PROPOSED SOLUTION FOR RRM PROBLEM

A. Distribution of Sensor Budgets Using LR

This paper is partly based on our previous research [36], where we used LR to include the constraints into the cost function. By doing so, the original optimization problem is decoupled into smaller ones, one for each task. This leads to the Lagrangian dual (LD), which needs to be optimized. The LD problem (LDP) is formulated as

$$Z_D = \max_{\lambda_t} \left(\min_{\mathbf{a}_t} \left(\sum_{n=1}^N (c(\mathbf{a}_t^n, \mathbf{s}_t^n) + \lambda_t \cdot \Theta_t^n) \right) - \lambda_t \cdot \Theta_{\max} \right), \quad (7)$$

where $\lambda_t \in \mathbb{R}$ is the Lagrange multiplier for the budget constraint. Due to the sum in the LDP, the minimization problem can be solved for each target n in parallel before updating the Lagrangian multiplier in an iterative process. The exact procedure is shown in [36] and is summarized in the following, where an internal index l is used for the iterations within the LR process.:

- 1) $l = 0$: set an initial Lagrange multiplier ($\lambda = \lambda_0$).
- 2) For each task n , minimize the LD with respect to the actions and save resulting \mathbf{a}_l^n and Θ_l^n .
- 3) Choose a subgradient for the Lagrange multiplier as $\mu_l^\lambda = \sum_{n=1}^N \Theta_l^n - \Theta_{\max}$.

- 4) Check if $\mu_l^\lambda \approx 0$ with the desired precision. If it is, stop the process. The current λ_l , \mathbf{a}_l^n , and Θ_l^n are the final LR solution for λ_t at time t .
- 5) Set $\lambda_{l+1} = \max\{0, \lambda_l + \gamma_l \mu_l^\lambda\}$, where γ_l is the LR step size at time l . This stage is responsible for iteratively maximizing the LD with respect to λ .
- 6) Go to step 2 and set $l = l + 1$.

Lagrange multipliers and LR have been extensively covered in literature and more information can be found, e.g. in [2], [5], [6], [9], and [31].

B. Definition of a POMDP

A POMDP describes an MDP for which the state cannot be observed directly. Instead, an observation is taken, which generates a probability distribution over the possible states. This is called the belief state. Based on the belief state and the knowledge of the underlying MDP, a POMDP allows us to solve optimization problems nonmyopically, meaning that it takes the expected future into account. In the following, the time is assumed to be discretized in intervals k with length T , which is the time between two consecutive observations.

Generally, a POMDP is defined by the following parameters (see, e.g. [35], [17]):

State space \mathcal{S} : It consists of all possible states that can be reached within the process, see (1). At time step k , the state is defined as \mathbf{s}_k . Based on the underlying states and the received observations, the belief state defines a probability distribution over all possible states. It is defined as \mathbf{b}_k .

Action space \mathcal{A} : It consists of all possible actions within the process. Each action has a certain cost defined by the cost function. The action at time step k is denoted by \mathbf{a}_k .

Observation space \mathcal{Z} : It consists of all possible observations that can be received within the process. An observation at time step k is defined as \mathbf{z}_k .

Transition probability $\Psi(\mathbf{s}_k, \mathbf{s}_{k+1}, \mathbf{a}_k)$: It is the probability function $p(\mathbf{s}_{k+1}|\mathbf{s}_k, \mathbf{a}_k)$ that defines the probability of transitioning from state \mathbf{s}_k to state \mathbf{s}_{k+1} given action \mathbf{a}_k . Note: In this paper, the transition probability does not depend on the action.

Probability of observation $\mathcal{O}(\mathbf{z}_k, \mathbf{s}_{k+1}, \mathbf{a}_k)$: It is the probability function $p(\mathbf{z}_k|\mathbf{s}_{k+1}, \mathbf{a}_k)$ that defines the probability of receiving a certain observation \mathbf{z}_k when executing action \mathbf{a}_k with the resulting state being \mathbf{s}_{k+1} .

Cost function $c(\mathbf{s}_k, \mathbf{a}_k)$: It is the immediate cost of executing action \mathbf{a}_k in state \mathbf{s}_k . Note: In this paper, the cost function does not directly depend on the state.

Discount factor γ : It is a discount factor that discounts future time steps with respect to the present. Note: In this paper, the discount factor is always set to $\gamma = 1$.

POMDPs can be solved for finite or infinite horizons. In order to reduce the necessary computational power, a

limited horizon \mathcal{H} is assumed in this paper. The value of \mathcal{H} represents the number of considered measurement time steps into the future. Every time a new budget allocation is calculated, the horizon \mathcal{H} will be reapplied from the current moment in time. This approach is therefore also called a receding horizon.

In [13], Charlish and Hoffmann have written a very clear summary of the general solution of a POMDP, which is used as a base for the following equations. We would like to find the actions that minimize the total cost (value $V_{\mathcal{H}}$ over horizon \mathcal{H}). Starting at time step k_0 , this can be expressed as

$$V_{\mathcal{H}} = E \left[\sum_{k=k_0}^{k_0+\mathcal{H}} c(s_k, \mathbf{a}_k) \right]. \quad (8)$$

Using $C_B(\mathbf{b}_k, \mathbf{a}_k) = \sum_{s \in \mathcal{S}} \mathbf{b}_k(s) c(s, \mathbf{a}_k)$ being the expected cost given belief state \mathbf{b}_k , $V_{\mathcal{H}}$ can be written as a so-called value function of the belief state \mathbf{b}_{k_0} at time step k_0 :

$$V_{\mathcal{H}}(\mathbf{b}_{k_0}) = E \left[\sum_{k=k_0}^{k_0+\mathcal{H}} C_B(\mathbf{b}_k, \mathbf{a}_k) | \mathbf{b}_{k_0} \right]. \quad (9)$$

For belief state \mathbf{b}_0 and taking action \mathbf{a}_0 , the optimal value function is defined according to Bellman's equation [1] as

$$V_{\mathcal{H}}^*(\mathbf{b}_0) = \min_{\mathbf{a}_0 \in \mathbf{A}} (C_B(\mathbf{b}_0, \mathbf{a}_0) + \gamma \cdot E [V_{\mathcal{H}-1}^*(\mathbf{b}_1) | \mathbf{b}_0, \mathbf{a}_0]). \quad (10)$$

For very long or infinite horizons, the discount factor can be set to $\gamma < 1$. Using this equation, the optimal policy can be expressed as

$$\pi_0^*(\mathbf{b}_0) = \arg \min_{\mathbf{a}_0 \in \mathbf{A}} (C_B(\mathbf{b}_0, \mathbf{a}_0) + \gamma \cdot E [V_{\mathcal{H}-1}^*(\mathbf{b}_1) | \mathbf{b}_0, \mathbf{a}_0]). \quad (11)$$

For each \mathbf{b}_k and \mathbf{a}_k , the optimal so-called Q-value is then defined as

$$Q_{\mathcal{H}-k}(\mathbf{b}_k, \mathbf{a}_k) = C_B(\mathbf{b}_k, \mathbf{a}_k) + \gamma \cdot E [V_{\mathcal{H}-k-1}^*(\mathbf{b}_{k+1}) | \mathbf{b}_k, \mathbf{a}_k]. \quad (12)$$

Another way to find the optimal policy is to find the action \mathbf{a}_k that minimizes the optimal Q-value:

$$\pi_k^*(\mathbf{b}_k) = \arg \min_{\mathbf{a}_k \in \mathbf{A}} (Q_{\mathcal{H}-k}(\mathbf{b}_k, \mathbf{a}_k)). \quad (13)$$

It is therefore necessary to calculate the Q-value for all possible actions, which is generally infeasible.

C. Solution Methods for POMDPs

For solving a POMDP, there are both online and offline approaches. The choice of the type of these methods usually depends on the size of the state space. The so-called state-space explosion limits the usefulness of exact offline techniques.

Most offline methods are based on the so-called value iteration (VI), which iteratively calculates the

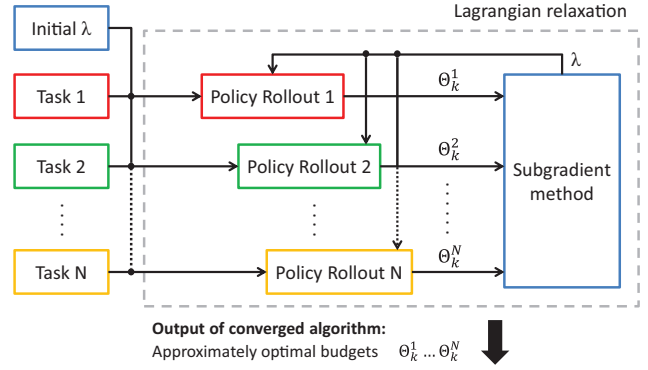


Fig. 1. High-level block scheme of the proposed algorithm.

cost/reward values of all possible states. There are exact approaches to VI (e.g. One-Pass algorithm [40]), as well as approximate point-based algorithms (e.g. PBVI or Perseus [41]). The former techniques often lead to very complicated optimization problems, while the latter ones require many grid points within the state space (and therefore a lot of memory and computational effort) in order to converge toward the exact solution. The advantage of offline solutions is that the POMDP is solved only once, and the solution is always valid afterward. Unfortunately, those methods are already infeasible for a small-dimensional state space.

In contrast to that, online algorithms only solve a small part of the POMDP that is relevant at the current moment. This makes them less accurate, but much easier and faster to compute. Some of the online approaches involve approximate tree methods (see, e.g. the overview in [35]) or Monte Carlo sampling (e.g. policy rollout).

Since an exact and complete solution of the POMDP is usually infeasible in real scenarios, this paper focuses on the implementation of policy rollout as an approximate solution. The general structure of our proposed algorithm is illustrated in Fig. 1. The outputs of the algorithm are the converged budgets for each task.

D. Policy Rollout for POMDPs

The policy rollout technique takes Monte Carlo samples of the expected future, which means that it stochastically explores the possible future actions and the according costs. Within a rollout, observations and belief states are generated from a given initial belief state and a given candidate action. There is a rollout evaluation per action \mathbf{a} in the action space \mathbf{A} . The candidate action is taken in the first step of the rollout, while a so-called base policy π_{base} is used for every following step, until the horizon \mathcal{H} is reached. In each rollout, the total cost is summed up. This procedure is repeated M times and then the summed cost of all M rollouts is averaged. This is the expected cost of the evaluated action. The action that produced the lowest expected cost is chosen for the next time step. It has been shown that policy rollout leads to a policy that is at least as good as the base policy with a very high probability, if enough samples are provided

TABLE I
System Parameters of the Assumed Radar Systems with Respect to the Reference Measurement

System	Measurement	$\sigma_{r,0}^2$ (m ²)	$\sigma_{\theta,0}^2$ (rad ²)	$\sigma_{d,0}^2$ ((m/s) ²)
A	r/θ	625	4e-4	-
B	$r/\theta/d$	2500	2e-4	25

[3]. The choice of the base policy and the amount of samples to be taken are therefore crucial to the performance of the algorithm. The number of samples is equivalent to the number of rollouts M per action that are used to average the cost, or, in other words, one sample is the evaluation of one possible future. Finding a good base policy for a radar scenario is no trivial task. As an example, one could think about using information from previously experienced situations that were similar to the current one. If the executed actions from the last run have been saved, they can be reused again to improve the policy further. This could be considered in the context of RL, for instance. Unfortunately, it is not very likely to experience the exact same situation multiple times if a very big state space is assumed, so the usefulness in such a case is questionable (see also our remark about RL in Section I-B). Another very simple choice of the base policy could be an equal resource allocation to all the tasks. Policy rollout has been covered extensively, e.g. by Bertsekas in [2]–[4].

The policy rollout can be expressed mathematically as shown in (14) and (15). The Q -value is defined as

$$Q^{\pi_{\text{base}}}(\mathbf{b}_k, \mathbf{a}_k) = C_B(\mathbf{b}_k, \mathbf{a}_k) + E[V^{\pi_{\text{base}}}(\mathbf{b}_{k+1}) | \mathbf{b}_k, \mathbf{a}_k], \quad (14)$$

where $E[\cdot]$ is the expectation. The best policy can then be found by applying

$$\pi_k(\mathbf{b}_k) = \arg \min_{\mathbf{a}_k \in \mathcal{A}} (Q^{\pi_{\text{base}}}(\mathbf{b}_k, \mathbf{a}_k)). \quad (15)$$

Policy rollout does not necessarily lead to the optimal policy. It rather aims at improving the chosen base policy π_{base} .

IV. ASSUMED RADAR SCENARIO

For the rest of this paper, we assume a two-dimensional radar tracking example that will be solved using the AODB algorithm. Measurements are taken in range, angle, and possibly radial velocity. The algorithm is jointly optimizing the revisit interval T (the time between two consecutive measurements) and the dwell time τ (the time that the sensor spends focusing on a target). The algorithm calculates the budgets of all tasks and makes sure that they fit into the time frame, but does not create an explicit schedule. Therefore, the assumed measurements are taken independently of each other and can be overlapping in time. In order to put all tasks into a single timeline, an explicit scheduler needs to

TABLE II
Parameters of Reference Measurement

SNR (SNR ₀)	RCS (ζ_0)	Dwell time (τ_0)	Range (r_0)
1	10 m ²	1 s	50 km

be implemented at a lower level. At which moments this budget calculation is performed depends on the preferences of the user. In the following, the assumptions of the assumed radar scenario are explained in more details.

A. Assumed Radar Systems

In the simulations, two different sets of system parameters are assumed as given in Table I. The table shows the measurement noise variances for range ($\sigma_{r,0}^2$), azimuth ($\sigma_{\theta,0}^2$) angle, and radial velocity ($\sigma_{d,0}^2$) with respect to the measurement of a reference target. The parameters of the reference measurement are shown in Table II and are valid for all simulations that are presented in this paper. Radar system A measures range and azimuth only, while system B is able to measure radial velocity as well. The values of the variances in Table I are chosen rather arbitrarily. We do not intend to compare the different radar systems, but rather use them to show how the AODB algorithm can universally be applied to different systems.

B. Velocity Model

The velocity model is assumed to be constant. Between two resource allocation updates, the actions are assumed to stay unchanged. The action vector $\mathbf{a}_n \in \mathbb{R}^2$ consists of the dwell time and the revisit interval. The latter defines the time between the measurements of target n . In contrast to our previous publications, in this paper, the revisit interval T_n and the dwell time τ_n are optimized jointly. The revisit intervals with length T_n depend on the targets and are therefore denoted by k_n . Considering this, (2) can explicitly be written as

$$\mathbf{s}_{k_n+1}^n = \mathbf{F}_n \mathbf{s}_{k_n}^n + \mathbf{w}_{k_n}^n, \quad (16)$$

with $\mathbf{F}_n \in \mathbb{R}^{4 \times 4}$ defined as

$$\mathbf{F}_n = \begin{bmatrix} 1 & 0 & T_n & 0 \\ 0 & 1 & 0 & T_n \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (17)$$

and the maneuverability noise \mathbf{w}^n with covariance

$$\mathbf{Q}_n = \begin{bmatrix} T_n^4/4 & 0 & T_n^3/2 & 0 \\ 0 & T_n^4/4 & 0 & T_n^3/2 \\ T_n^3/2 & 0 & T_n^2 & 0 \\ 0 & T_n^3/2 & 0 & T_n^2 \end{bmatrix} \sigma_{w,n}^2, \quad (18)$$

where $\sigma_{w,n}^2$ is the maneuverability noise variance of target n .

Because of the nonlinear relationship between measurements and states, an EKF is applied. The corresponding observation matrix $\mathbf{H}_{k_n}^n$ is defined as the Jacobian of the measurement transformation function \mathbf{h} :

$$\mathbf{H}_{k_n}^n = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{s}} \right|_{\mathbf{s}_{k_n}^n}. \quad (19)$$

It has dimensions $\mathbf{H}_{k_n}^n \in \mathbb{R}^{2 \times 4}$ for system A and $\mathbf{H}_{k_n}^n \in \mathbb{R}^{3 \times 4}$ for system B.

C. Signal-To-Noise Ratio (SNR) Model

In the following examples, we assume sensor measurements in range (r), azimuth (θ), and radial velocity (d). Since the transformation between polar and Cartesian coordinates is nonlinear, the measurement equation in (4) for target n at time step k_n can be defined as

$$\mathbf{z}_{k_n}^n = \mathbf{h}(\mathbf{s}_{k_n}^n) + \mathbf{v}_{k_n}^n, \quad (20)$$

where $\mathbf{h}(\mathbf{s}_{k_n}^n) \in \mathbb{R}^3$ is the measurement transformation function at $\mathbf{s}_{k_n}^n$, which for system B is defined as

$$\mathbf{h}(\mathbf{s}_{k_n}^n) = \left[\sqrt{(x_{k_n}^n)^2 + (y_{k_n}^n)^2}, \text{atan2}(y_{k_n}^n, x_{k_n}^n), \frac{x_{k_n}^n \dot{x}_{k_n}^n + y_{k_n}^n \dot{y}_{k_n}^n}{\sqrt{(x_{k_n}^n)^2 + (y_{k_n}^n)^2}} \right]^T \quad (21)$$

and $\mathbf{v}_{k_n}^n \in \mathbb{R}^3$ is the measurement noise for target n . The range, azimuth, and radial velocity components of $\mathbf{v}_{k_n}^n$ are independent:

$$\mathbf{v}_{k_n}^n = [v_{k_n}^{r,n} \quad v_{k_n}^{\theta,n} \quad v_{k_n}^{d,n}]^T, \quad (22)$$

with variances $\sigma_{r,n}^2$, $\sigma_{\theta,n}^2$, and $\sigma_{d,n}^2$. In this paper, the SNR is calculated by using (23), which is based on equations by Koch in [27]:

$$\text{SNR}_{k_n}(\zeta_n, \tau_n, r_{k_n}^n) = \text{SNR}_0 \left(\frac{\zeta_n}{\zeta_0} \right) \left(\frac{\tau_n}{\tau_0} \right) \left(\frac{r_{k_n}^n}{r_0} \right)^{-4} e^{-2\Delta\alpha}, \quad (23)$$

where $\Delta\alpha$ is the relative beam positioning error, ζ_n is the constant radar cross section (RCS) of the target n , $r_{k_n}^n$ is the distance of target n at time step k_n , and ζ_0 , τ_0 , and r_0 are the corresponding values for a reference target. In (23), the dwell time is used equivalently to the transmitted energy mentioned by Koch. Similar to the approach in [27], the relative beam positioning error is calculated using

$$\Delta\alpha = \frac{(\theta_{k_n} - \hat{\theta}_{k_n})^2}{\Gamma^2}, \quad (24)$$

where θ_{k_n} is the real target angle, $\hat{\theta}_{k_n}$ is the predicted target angle in azimuth at time k_n , and Γ is the one-sided beam width in azimuth.

Using (23), the variance of the range, azimuth, and radial velocity measurement noise for target n can be de-

defined as (see, e.g. [32])

$$\sigma_{\bullet,n}^2 = \frac{\sigma_{\bullet,0}^2}{\text{SNR}_{k_n}(\zeta_n, \tau_n, r_{k_n}^n)}, \quad (25)$$

where $\bullet \in (r, \theta, d)$ and $\sigma_{\bullet,0}^2$ is the measurement noise variance for a reference target 0, as defined in Table I.

Due to the independent measurements, the measurement covariance when using system B can be defined as

$$\mathbf{R}_{k_n}^n = \begin{bmatrix} \sigma_{r,n}^2 & 0 & 0 \\ 0 & \sigma_{\theta,n}^2 & 0 \\ 0 & 0 & \sigma_{d,n}^2 \end{bmatrix}. \quad (26)$$

D. Optimization Problem

It is assumed that there are N tracked targets in the environment. The RRM problem can thus be expressed as

$$\begin{aligned} & \underset{\mathbf{T}, \boldsymbol{\tau}}{\text{minimize}} && \sum_{n=1}^N c(T_n, \tau_n, \mathbf{s}_{k_n}^n) \\ & \text{subject to} && \sum_{n=1}^N \frac{\tau_n}{T_n} \leq \Theta_{\max}, \end{aligned} \quad (27)$$

where $\Theta_{\max} \in [0, 1]$ is the total available budget. The term budget refers to a ratio of dwell time τ to revisit interval T .

Furthermore, every detection is always correctly assigned to the corresponding target.

E. Cost Function

The assumed cost function is constructed from the predicted error-covariance matrix at time step $k_n + 1$. The current predicted error-covariance matrix $\mathbf{P}_{k_n|k_{n-1}}^n \in \mathbb{R}^{4 \times 4}$ at time step k_n can be defined for target n as

$$\mathbf{P}_{k_n|k_{n-1}}^n(T_n, \tau_n) = \mathbf{F}_n \mathbf{P}_{k_{n-1}|k_{n-1}}^n(T_n, \tau_n) \mathbf{F}_n^T + \mathbf{Q}_n, \quad (28)$$

where \mathbf{F}_n is the transition matrix with interval length T_n as defined in (17), $\mathbf{P}_{k_{n-1}|k_{n-1}}^n \in \mathbb{R}^{4 \times 4}$ is the last filtered error-covariance matrix, and \mathbf{Q}_n is the maneuverability covariance with interval length T_n as defined in (18). Based on this, another estimation and prediction cycle is applied to the error-covariance. The result is the error-covariance $\mathbf{P}_{k_n+1|k_n}^n \in \mathbb{R}^{4 \times 4}$ for time $k_n + 1$:

$$\mathbf{P}_{k_n+1|k_n}^n(T_n, \tau_n) = \mathbf{F}_n \mathbf{P}_{k_n|k_n}^n(T_n, \tau_n) \mathbf{F}_n^T + \mathbf{Q}_n. \quad (29)$$

The cost function that is used in the following sections is based on this expression.

V. LTI EXAMPLE

In this section, a simplified linear time-invariant scenario is assumed in order to investigate if the AODB algorithm converges to the same results as given by the OSB algorithm, which is the optimal solution in this case.

TABLE III
General Simulation Parameters of LTI Scenario

Parameter	Value	Parameter	Value
Precision of LR (δ_{LR}):	0.001	Maximum budget (Θ_{\max}):	1
Action discretization ($\Delta T, \Delta \tau$):	0.0025 s	Budget update (t_B):	5 s
Number of rollouts (M):	10	Beam positioning error ($\Delta \alpha$):	0
Rollout horizon (\mathcal{H}):	10 steps	Probability of detection (P_D):	1

A. General Simulation Parameters

We consider radar system A, as mentioned in Table I. For this simple example, no beam positioning error is taken into account, e.g. due to a very wide beam by using an MFR with a phased array antenna applying digital beamforming (DBF) on receive. The probability of detection is assumed to be 1. The implemented base policy is simply to apply the evaluated action in every step of the policy rollout. Therefore, $\pi_{\text{base}} = \mathbf{a}$. A constant LR step size is applied in all simulations. Within the policy rollout, the expected future cost is simulated over the defined horizon for each possible action. The action that produces the lowest expected cost will be chosen for the measurements during the next time steps. No additional random movement (process noise) is considered within the policy rollout. For the simulations in this section, the sum of the predicted error-covariance for the position in the x and y directions is applied as a cost function. Because we want to avoid choosing parameters that are impractical in a real application, an extra term is added that penalizes small values of T . Using (29), this can be expressed as

$$c(T_n, \tau_n) = \text{trace}(\mathbf{E}\mathbf{P}_{k_n+1|k_n}(T_n, \tau_n)\mathbf{E}^T) + \frac{1000}{(T_n)^2}, \quad (30)$$

where

$$\mathbf{E} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (31)$$

is the selection matrix that selects the upper left two-by-two submatrix from the error-covariance matrix.

Table III shows general simulation parameters. The initial Lagrange multiplier value is set to 1. The budgets are recalculated every $t_B = 5$ s. The base policy is executing the evaluated action in every step of the policy rollout horizon ($\pi_{\text{base}} = \mathbf{a}$). Within the policy rollout, the expected future is simulated and evaluated for each possible action. The radar is always positioned at the origin of the Cartesian coordinate system.

B. Comparison of OSB and AODB

In order to prove the validity of the proposed AODB algorithm, a comparison is conducted with the OSB al-

TABLE IV
Initial Target Parameters for LTI Scenario

Target n	x_0^n (km)	y_0^n (km)	x_0^n (m/s)	y_0^n (m/s)	σ_w^2 (m/s ²) ²	S_n (m ²)
1	50	0	0	0	25	10
2	50	0	0	0	25	20
3	50	0	0	0	25	30
4	50	0	0	0	25	40
5	50	0	0	0	25	50

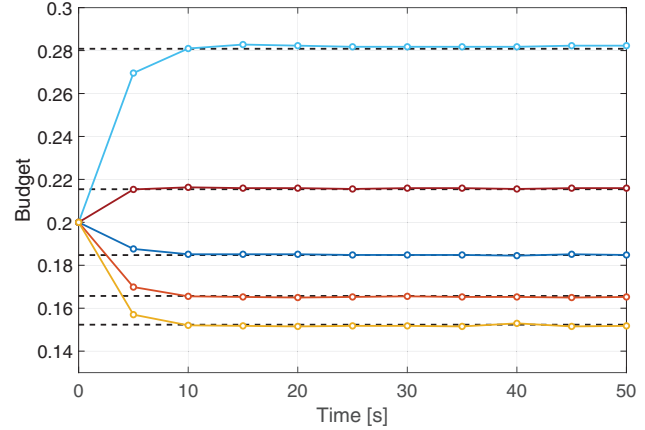


Fig. 2. Budget per task over time after initialization of the AODB algorithm. Solid lines: results from AODB. Dashed lines: optimal steady-state results from OSB. Lines from top to bottom: targets 1–5.

gorithm, as proposed in [36]. The OSB algorithm calculates the optimal steady-state error-covariance given a revisit interval T and a dwell time τ by using equations by Kalata in [23] and by Gray and Murray in [18]. It is used as explained in [36] with the general simulation parameters from Table III.

For the comparison, system A and five target tracking tasks are considered with the parameters shown in Table IV. The revisit interval T and the dwell time τ are discretized in steps of 0.0025 s. It is assumed that the budget values are recalculated every 5 s. In between, measurements of the targets are taken with the previously calculated revisit intervals T_n and dwell times τ_n . The tracks are assumed to be initialized at the beginning of the simulation.

Since the steady-state solution of the OSB algorithm is only valid for a single dimension, we assume that the targets are all positioned at the same position and the system knows the exact azimuth angle. All targets are static and only the RCS is considered to be different.

The simulation results are shown Fig. 2. It can be seen that the budget allocations $\Theta^n = \tau_n/T_n$ converge to results that are very close to the values that have been determined with the OSB algorithm.

Theoretically, the AODB algorithm should work with any number of tasks. In order to demonstrate that, the above simulation has been repeated with 10 tasks. Equivalent to targets 1–5, the RCS values of targets 6–10

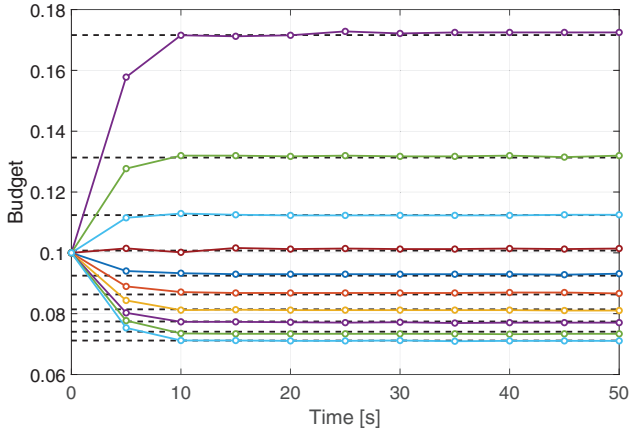


Fig. 3. Budget per task over time after initialization of the AODB algorithm. Same simulation as for Fig. 2, but with ten tracking tasks. Lines from top to bottom: targets 1–10.

TABLE V
General Simulation Parameters of Dynamic Scenario

Parameter	Value	Parameter	Value
Precision of LR (δ_{LR}):	0.001	Maximum budget (Θ_{max}):	1
Action discretization ($\Delta T, \Delta \tau$):	0.0025 s	Budget update (t_B):	5 s
Number of rollouts (M):	5	Beam positioning error ($\Delta \alpha$):	0
Rollout horizon (\mathcal{H}):	15 steps	Probability of detection (P_D):	1

are increasing in steps of 10 m^2 . Fig. 3 shows the approximately optimal budget distribution.

VI. DYNAMIC RADAR EXAMPLE

In this section, the performance of the AODB algorithm is investigated in a more realistic radar-related example with different system parameters.

A. General Simulation Parameters

The cost function as introduced in (30) is applied. Table V shows general simulation parameters for these simulations. The initial Lagrange multiplier value is set to 1. The budgets are recalculated every $t_B = 5 \text{ s}$ and measurements are taken in between with the current calculated resource allocations. The base policy is executing the evaluated action in every step of the policy rollout horizon ($\pi_{base} = \mathbf{a}$). Within the policy rollout, the expected future is simulated and evaluated for each possible action. The radar is always positioned at the origin of the Cartesian coordinate system.

B. Dynamic Radar Scenario for $P_D = 1$

A dynamic scenario with five moving targets is considered in this simulation. The initial target parameters

TABLE VI
Initial Target Parameters for Dynamic Scenario

Target n	x_0^n (km)	y_0^n (km)	x_0^n (m/s)	y_0^n (m/s)	σ_w^2 (m/s^2) ²	ζ_n (m^2)
1	12	10	9	-15	25	20
2	12	15	-30	15	25	20
3	7	11	45	30	64	10
4	19	2	-35	0	64	10
5	10	11	-20	-25	64	10

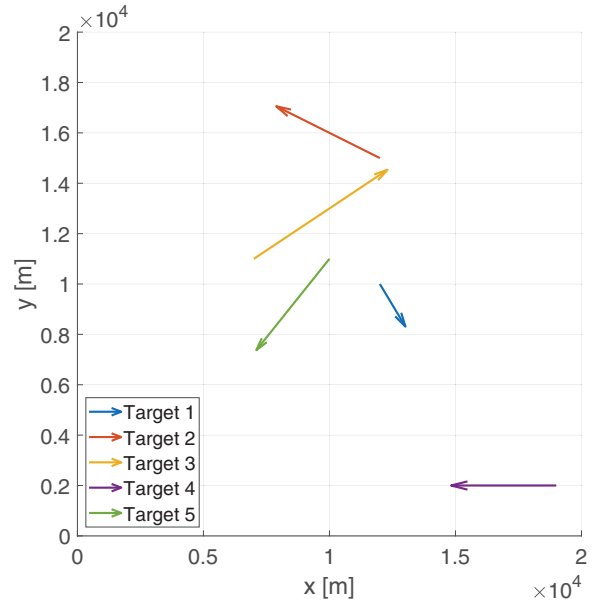


Fig. 4. Trajectories of targets in a dynamic scenario.

are given in Table VI and are valid at the moment when the corresponding track is started. Their trajectories are shown in Fig. 4. The simulation is conducted with systems A and B separately. As in the LTI simulations of Section V, no beam positioning error is taken into account, e.g. due to a very wide beam by using an MFR with a phased array antenna applying DBF on receive. The probability of detection is assumed to be 1. A horizon of $\mathcal{H} = 15$ is assumed. Targets 1–4 are tracked from the beginning, while target 5 joins as a new track after 25 s. After 60 s, the total budget is reduced to $\Theta_{max} = 0.9$. The reason for this could be that an operator manually assigned 10% of the budget to another task, for instance. At 90 s, the maneuverability variance of target 1 increases by a factor of 36 to a value of $900 (\text{m/s}^2)^2$, which is known to the system in advance, for instance, through some knowledge of the environment. The simulation results for system A can be found in Figs. 5 and 6, where the former shows the resource distribution over the tasks over time and the latter shows the amount of LR iterations that was needed for convergence. The corresponding simulation results for system B are shown in Figs. 7 and 8.

The algorithm manages to calculate the budget for both systems, while adjusting to unknown and known changes. Before the known variance change at 90 s, the

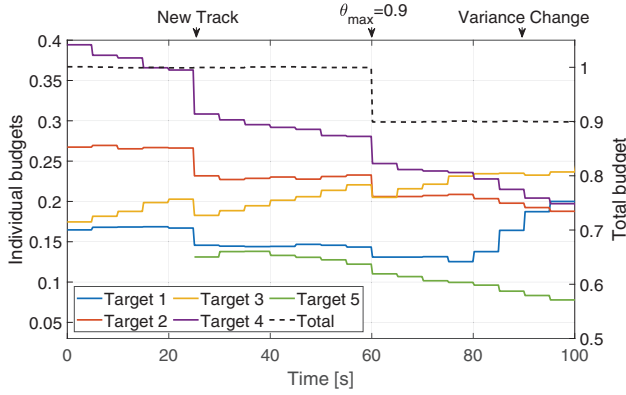


Fig. 5. Dynamic scenario simulation using radar system A.

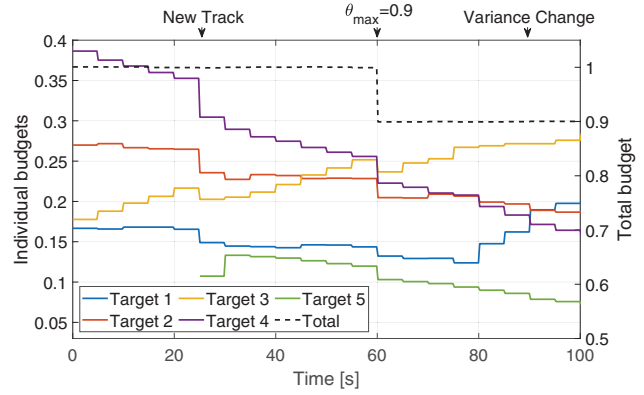


Fig. 7. Dynamic scenario simulation using radar system B.

algorithm already gradually increases the budget for target 1. The algorithm delivers very similar but still different solutions for the two systems. It can be seen that the amount of LR iterations needed until convergence stays low, unless bigger changes in the situation take place. For the chosen parameters, the maximum is 66 iterations for a single resource allocation calculation assuming system B. Using system A leads to similar peak values.

Apart from the impact of the three mentioned sudden changes that are applied to the system, it is also obvious that there seems to be a certain dependence of the budgets on the range. While the budget assigned to targets 1 and 2 stays roughly constant in between different events, target 3 gets an increasing amount of resources assigned, while the resources of targets 4 and 5 are decreasing. The reason for this is that targets 1 and 2 are moving roughly perpendicular to the radar, while target 3 is moving away from it and targets 4 and 5 are moving toward it. In Section VI-D, this effect is investigated with an extra simulation.

C. Dynamic Radar Scenario for $P_D < 1$

In a real situation, a low SNR can lead to missed detections. In addition to that, the used radar system might not have the capability to transmit with a wide beam and apply DBF on receive. Therefore, another simulation is presented in this subsection that takes into account a probability of detection based on the calculated SNR

and the beam positioning error. The scenario is identical with the one shown in Section VI-B, and apart from the probability of detection and the beam positioning error, all values in Table V are applied. The SNR is calculated using (23) and taking into account a beam width of 2° . In addition to that, a measurement in the simulation as well as in the policy rollout is only generated with the probability of detection [27]

$$P_{D,k_n} = P_{FA}^{\frac{1}{1+SNR_{k_n}}}, \quad (32)$$

where $P_{FA} = 10^{-4}$ is the constant probability of false alarm. It is assumed that the false alarms have no influence on the tracks. The result of this simulation assuming system A can be found in Figs. 9 and 10.

It can be seen that the resulting budget allocations are less smooth than in the simulations assuming $P_D = 1$. Still, the AODB algorithm leads to comparable results despite the fact that some of the probabilities of detection are quite low.

D. Analysis of the Impact of the Chosen Cost Function

To show the impact of the range on the resource distribution by the AODB algorithm, another simulation has been conducted with three targets. Target 1 has the initial parameters $x_0 = 6$ km, $y_0 = 6$ km, $\dot{x}_0 = 50$ m/s, and $\dot{y}_0 = 50$ m/s. Targets 2 and 3 are static at positions $x = 12.4$ km, $y = 9$ km and $x = 8.4$ km, $y = 9.2$ km, respectively. The simulation result is presented in Fig.

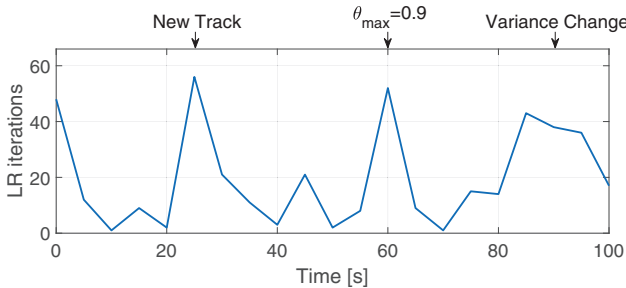


Fig. 6. Number of LR iterations for a dynamic scenario simulation using radar system A.

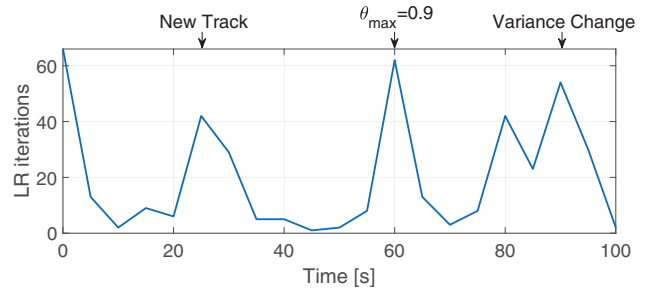


Fig. 8. Number of LR iterations for a dynamic scenario simulation using radar system B.

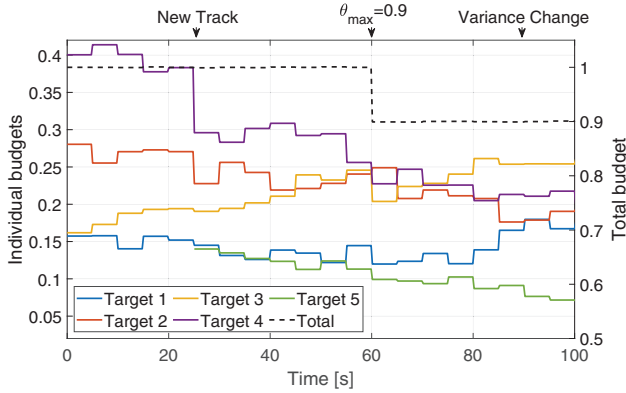


Fig. 9. Dynamic scenario simulation using radar system A with $P_D < 1$.

11 and shows that the budget assigned to target 1 is increasing with growing target distance from the radar, while the budget assigned to the other targets is decreasing. This behavior is expected but does not represent what is typically desired or expected for a radar application.

VII. ANALYSIS OF PERFORMANCE

In the following subsections, we will take a closer look at the general performance of the AODB algorithm with respect to other resource allocation methods.

The assumed scenario is the same as in Section VI, so the radar and target parameters are identical to Tables I, V, and VI. For the following simulations, we consider one implementation of the AODB algorithm and three other strategies using radar system A. The cost evaluation is done for two cases, firstly for $P_D = 1$ and secondly for $P_D < 1$, based on the SNR including a beam positioning error as presented in Section VI-C.

It is generally difficult to judge the performance of RRM algorithms in theory, because it depends on the specific situation and the specific mission where they are applied in. Depending on the user of the radar system, there might be different views on the different parameters. It is possible to show that an approach optimizes the resource distribution according to the chosen cost function, but if the cost function is not well designed, the tracking, detection, or classification performance can

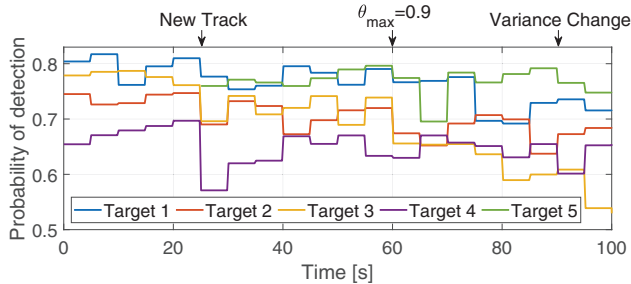


Fig. 10. Average probability of detection per budget update interval for the dynamic scenario with $P_D < 1$.

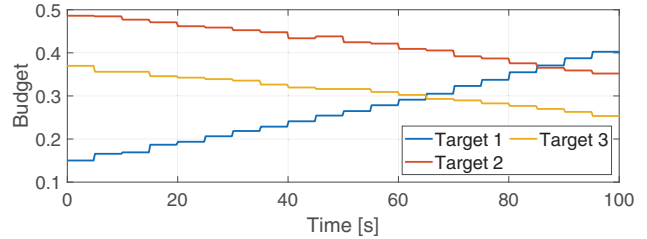


Fig. 11. Budget allocation for two static and one moving target.

still be unsatisfying. Therefore, we only focus on the expected cost in this section.

The techniques that are compared to each other are as follows:

- **Random policy:** For a given revisit interval $T = 1.2$ s, randomly divide the available resources among all tasks.
- **Equal policy:** For a given revisit interval $T = 1.2$ s, the available budget is always distributed equally to all tasks.
- **Unequal policy:** For a given revisit interval $T = 1.2$ s, target 1 gets more resources assigned than the other targets. The remaining resources are distributed equally over targets 2–5.
- **AODB15:** Nonmyopic AODB algorithm, assigning resources using policy rollout ($\mathcal{H} = 15, M = 5$).

Figs. 12 and 13 show how the expected cost develops over time for the different techniques that are mentioned above. For the heuristic methods, the future expected cost during a horizon of $\mathcal{H} = 15$ has been evaluated stochastically assuming the chosen action, equivalently to the policy rollout. One can see how the AODB clearly minimizes the cost compared to the other techniques for both $P_D = 1$ and $P_D < 1$.

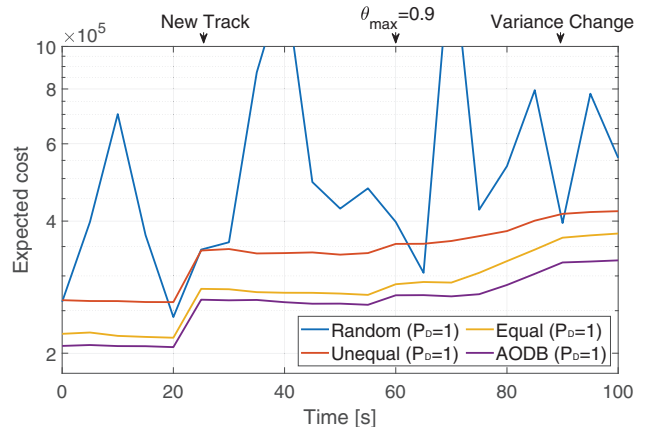


Fig. 12. Comparison of the expected cost for different resource distribution methods assuming radar system A and $P_D = 1$. Note that the cost is plotted in a logarithmic scale.

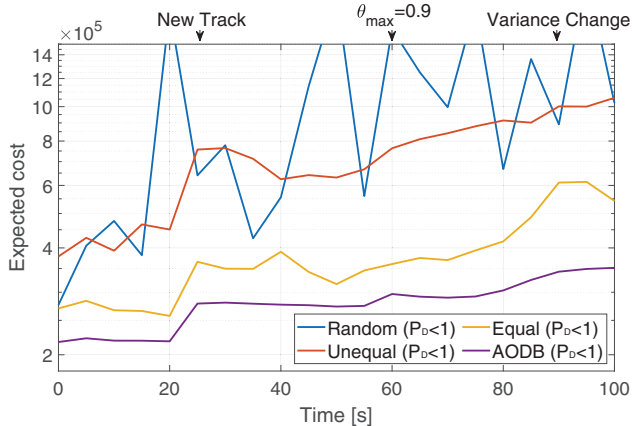


Fig. 13. Comparison of the expected cost for different resource distribution methods assuming radar system A and $P_D < 1$. Note that the cost is plotted in a logarithmic scale.

TABLE VII
General Simulation Parameters for Computational Load Analysis

Parameter	Value	Parameter	Value
Precision of LR (δ_{LR}):	0.01	Maximum budget (Θ_{max}):	1
Action discretization ($\Delta T, \Delta \tau$):	0.0035 s	Number of simulations:	10
Number of rollouts (M):	2	Beam positioning error ($\Delta \alpha$):	0
Rollout horizon (\mathcal{H}):	2 steps	Probability of detection (P_D):	1

VIII. ANALYSIS OF COMPUTATIONAL LOAD

In this section, the computational load of the AODB algorithm is investigated. It should be noted that the current version of the algorithm has not been derived with high efficiency in mind. The following results should be seen as indications, since the process can still be optimized.

The computational load of the algorithm has been investigated with respect to the following parameters:

- amount of tracking tasks,
- step size of LR,
- desired precision of results,
- initial value of the Lagrange multiplier,
- rollout length.

In the following, simulation results are shown based on a single budget calculation. This means that we look at the way the LR converges to its final result based on the above parameters. To generate the figures, the results of ten simulations have been averaged. The general simulation parameters are shown in Table VII. Those parameters are valid for all following simulations, except for the currently evaluated parameter. For that one, a sweep over different values is applied, which is specified in the corresponding subsection. We assume a fixed action space that is the same for each calculation in the

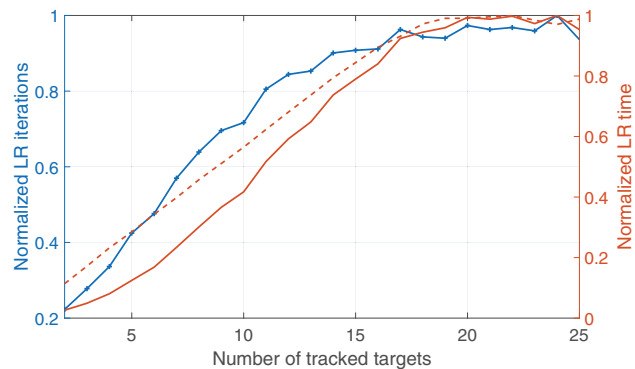


Fig. 14. Convergence for different numbers of tracking tasks. Number of LR iterations (blue with crosses) and total time (solid red) needed for convergence, as well as time per LR iteration (dashed red).

parameter sweep. The chosen system for these simulations produces measurements in range and angle (system A) and the target parameters are the same as in Section VI (see Table VI). The initial Lagrange multiplier value is set to 1. In addition to that, the cost function as introduced in (30) is used for all following simulations. In the following figures, we show normalized times and normalized LR iterations numbers. This means that each data graph is normalized w.r.t. its maximum value. This is done in order to emphasize that the capability of the hardware and the choice of the general input parameters are not relevant for the discussion of the results.

A. Influence of Number of Tasks on AODB

The following simulation shows the influence of an increasing number of tasks on the computational load and execution time of the AODB algorithm. Using the above-mentioned parameters, 24 different simulations have been conducted for 2–25 tracking tasks. The initial Lagrange multiplier value is 1 and the chosen constant LR step size is 8000. Therefore, it is assumed that there is no prior knowledge about the optimal Lagrange multiplier. The results of this simulation can be seen in Fig. 14. It can be seen that the amount of iterations, the total time until the LR converges, and the time needed for each LR iteration are increasing approximately linearly for a rising number of tracked targets, until the increase slows down for larger amounts of targets of 15 and more.

B. Influence of LR Step Size on AODB

In this subsection, a simulation shows the influence of an increasing LR step size on the computational load and execution time of the AODB algorithm. We consider 5 tracking tasks and 50 step sizes between 250 and 20 000, while the initial Lagrange multiplier value is 1. The results of this simulation can be found in Fig. 15. It can be seen that the amount of LR iterations needed and the time until convergence are decreasing exponentially.

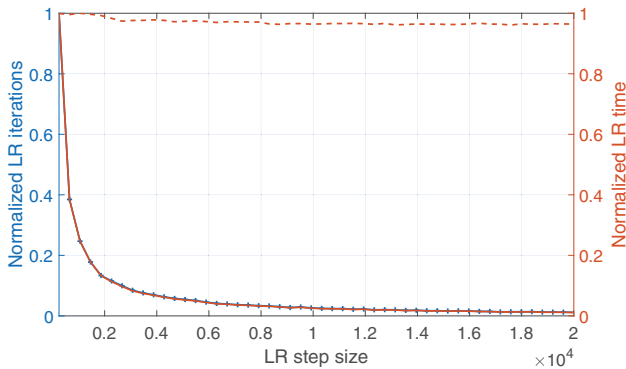


Fig. 15. Convergence for different LR step sizes. Number of LR iterations (blue with crosses) and total time (red) needed for convergence, as well as average time per LR iteration (dashed red).

The average time per LR iteration stays approximately constant.

C. Influence of LR Precision on AODB

The following simulation shows the influence of different LR result precisions on the computational load and execution time of the AODB algorithm. We consider 5 tracking tasks and 50 precision values between 0.001 and 0.2. The results of this simulation can be found in Fig. 16. The initial Lagrange multiplier value is 1 and the chosen constant step size is 8000. It can be seen that the amount of LR iterations and the total LR convergence time are decreasing roughly exponentially. The average time per LR iteration stays approximately constant.

D. Influence of Initial Lagrange Multiplier Value on AODB

This simulation shows the influence of different initial Lagrange multiplier values on the computational load and execution time of the AODB algorithm. We consider 5 tracking tasks and 50 initial Lagrangian multiplier values between 1 and 100 000. The chosen constant step size is 8000. The results of this simulation can be found in Fig. 17. It can be seen that the amount of

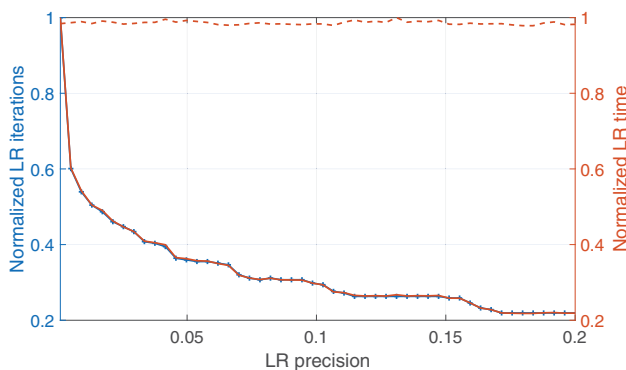


Fig. 16. Convergence for different LR result precisions. Number of LR iterations (blue with crosses) and total time (red) needed for convergence, as well as average time per LR iteration (dashed red).

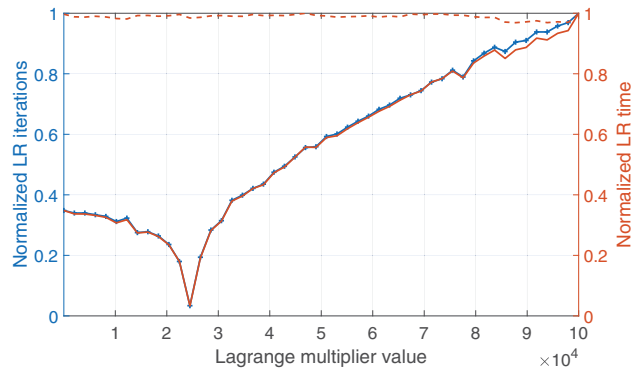


Fig. 17. Convergence for different initial Lagrange multiplier values. Number of LR iterations (blue with crosses) and total time (red) needed for convergence, as well as average time per LR iteration (dashed red).

LR iterations and the LR convergence time have a clear minimum at about 24 000. This is the best starting value, because it allows for the fastest convergence. The average time per LR iteration stays approximately constant.

E. Influence of Rollout Horizon Lengths on AODB

The following simulation shows the influence of different policy rollout horizon lengths on the computational load and execution time of the AODB algorithm. We consider five tracking tasks and the rollout length to vary from 1 to 25. The initial Lagrange multiplier value is 1 and the chosen constant step size is 8000. The results of this simulation can be found in Fig. 18. It can be seen that the amount of LR iterations increases fast in the beginning, before slowly decreasing again for horizon lengths of 6 and longer. The total time needed increases approximately linearly, as well as the time needed per LR iteration.

F. Conclusions on Computational Load

Based on the simulation result of the previous subsections, some conclusions can be made regarding the choice of the input parameters. They will be summarized in the following paragraphs.

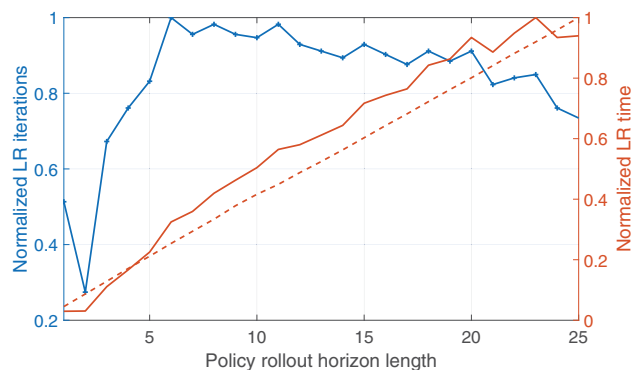


Fig. 18. Convergence for different rollout horizon lengths. Number of LR iterations (blue with crosses) and total time (red) needed for convergence, as well as average time per LR iteration (dashed red).

Number of targets and initial Lagrange multiplier value: Both the necessary number of LR iterations and the total LR convergence time increase with an increasing number of tracking tasks. Unfortunately, it is generally not possible to influence the amount of tasks at will. However, the effect of increasing convergence time can be reduced by choosing the appropriate initial Lagrange multiplier value. We found that there is a distinct minimum in the number of LR iterations before convergence (see Fig. 17). The minimum convergence time, which is equivalent to a single LR iteration, is attained when that Lagrange multiplier value is chosen as the initial value. It is interesting to see that initial Lagrange multiplier values that are bigger than the optimal value lead to longer computations compared to values smaller than the optimum. If some prior knowledge about the Lagrange multiplier value is available (e.g. from the previous budget calculation), this can tremendously decrease the convergence time, if the situation has not changed too much since. Boyd et al. have labeled this approach a “Warm Start” [8].

LR step size and precision of LR result: Increasing LR step size and decreasing precision both lead to a decreasing number of LR iterations and time until convergence, while the time needed for one LR iteration stays more or less constant. Generally, it is useful to choose a rather big LR step size, but if it is chosen too big with respect to the precision and the action-space discretization, the algorithm might not converge but oscillate around the minimum. If the desired results lie in a local minimum instead of the global one, the algorithm might miss that minimum entirely, in case the step size is chosen too big. Therefore, choosing a constant step size is probably not the best solution and adaptive step sizes could increase the performance. There is more freedom to choose the precision, but one should keep in mind that a lower precision will lead to a less accurate result, which can lead to not precisely meeting the maximum budget constraint.

Policy rollout horizon length: Although this paper does not investigate the advantages of choosing different horizon lengths for the policy rollout, it was chosen to examine its impact on the computational load for the sake of completeness. In the future, the impact of the horizon needs to be studied in more detail. Increasing the horizon length leads to an almost linear increase of the time per LR iteration. Very short horizons seem to lead to very low numbers of LR iterations until convergence. For horizon lengths longer than 2, the number of LR iterations increases very quickly, although for horizons longer than 6, it slightly decreases again. The total LR convergence time increases with growing rollout length (see Fig. 18). It is therefore reasonable to choose the shortest horizon necessary. It needs to be kept in mind that this is a trade-off with an impact on the track performance, so a longer horizon can potentially improve the mission performance further.

IX. CONCLUSIONS

In this paper, we have developed a framework and proposed approximately optimal algorithmic solutions for solving RRM problems and shown applicability of the algorithm to a dynamic multitarget tracking scenario. The proposed framework models the different sensor tasks as constrained POMDPs and solves them by applying a combination of Lagrangian relaxation and policy rollout. In contrast to previous work where LTI scenarios were considered, this paper focuses on dynamic situations with different parameters. We believe that the proposed solution is a step toward a truly generic framework.

In a simple radar tracking scenario, the dwell time and the revisit interval were optimized using a cost function based on the predicted position error-covariance that was computed using the EKF.

It was shown that the AODB algorithm budget allocations are close to the optimal steady-state solution in an LTI setting. Furthermore, the simulation results show that the AODB algorithm can be applied to different systems, and it was pointed out how it adjusted itself to known as well as unknown situational changes in a dynamic scenario.

The presented cost function leads to a larger budget being given to tracking tasks with higher uncertainty. At first glance, this may seem to be fully appropriate; however, in radar this means that more budget will be assigned to targets at longer range. Thus, a simple error-covariance-based cost function will not always suit practical radar applications.

An analysis of the performance of the algorithm has also been conducted by comparing the optimized cost to other resource distribution methods. It was found that the AODB always led to the lowest cost values compared to the other considered techniques. Finally, the computational load of the algorithm was investigated. Based on those results, suggestions about a good choice of input parameters have been presented.

In future work, we will investigate the usage of the AODB algorithm in a combined tracking and classification scenario. Furthermore, we will investigate the impact of choosing different horizon lengths and its impact on the cost and the track accuracies. Finally, we will have a closer look at the convergence of the algorithm and how its efficiency can be improved.

REFERENCES

- [1] R. E. Bellman
Dynamic Programming. New Jersey, NJ, USA: Princeton Univ. Press, 1957.
- [2] D. P. Bertsekas
Constrained Optimization and Lagrange Multiplier Methods. Nashua, NH, USA: Athena Scientific, 1996.
- [3] D. P. Bertsekas and D. A. Castanon
“Rollout algorithms for stochastic scheduling problems,” *J. Heuristics*, vol. 5, no. 1, pp. 89–108, Apr. 1999.

- [4] D. P. Bertsekas, J. N. Tsitsiklis, and C. Wu Rollout “Algorithms for combinatorial optimization,” *J. Heuristics*, vol. 3, no. 3, pp. 245–262, Dec. 1997.
- [5] D. Bertsimas and J. N. Tsitsiklis Introduction to Linear Optimization. Nashua, NH, USA: Athena Scientific, 1997.
- [6] S. S. Blackman and R. Popoli Design and Analysis of Modern Tracking Systems. London, U.K.: Artech House, 1999.
- [7] M. Bockmair, C. Fischer, M. Letsche-Nuesseler, C. Neumann, M. Schikorr, and M. Steck “Cognitive radar principles for defence and security applications,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 34, no. 12, pp. 20–29, Dec. 2019.
- [8] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
- [9] S. Boyd and L. Vandenberghe *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [10] S. Brüggewirth, M. Warnke, S. Wagner, and K. Barth “Cognitive radar for classification,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 34, no. 12, pp. 30–38, Dec. 2019.
- [11] D. A. Castañón “Approximate dynamic programming for sensor management,” in *Proc. 36th IEEE Conf. Decis. Control*, Dec. 1997, pp. 1202–1207.
- [12] A. Charlish, K. Bell, and C. Kreucher “Implementing perception–action cycles using stochastic optimization,” in *Proc. IEEE Radar Conf.*, Sep. 2020.
- [13] A. Charlish and F. Hoffmann “Anticipation in Cognitive Radar using Stochastic Control,” in *Proc. 2015 IEEE Radar Conf.*, May 2015, pp. 1692–1697.
- [14] A. Charlish and F. Hoffmann “Cognitive radar management,” in *Radar Techniques and Applications, Volume 2 – Waveform Diversity and Cognitive Radar, and Target Tracking and Data Fusion*. London, U.K.: Scitech Publishing, 2017, pp. 157–193.
- [15] A. Charlish, F. Hoffmann, and I. Schlangen “The development from adaptive to cognitive radar resource management,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 35, no. 6, pp. 8–19, Jun. 2020.
- [16] A. Charlish, K. Woodbridge, and H. Griffiths “Phased array radar resource management using continuous double auction,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 3, pp. 2212–2224, Jul. 2015.
- [17] E. K. P. Chong, C. M. Kreucher, and A. O. Hero “Partially observable markov decision process approximations for adaptive sensing,” *Discrete Event Dyn. Syst.*, vol. 19, no. 3, pp. 377–422, Sep. 2009.
- [18] J. E. Gray and W. Murray “A derivation of an analytic expression for the tracking index for the alpha-beta-gamma filter,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 29, no. 3, pp. 1064–1065, Jul. 1993.
- [19] S. Haykin “Cognitive radar: a way of the future,” *IEEE Signal Process. Mag.*, vol. 23, no. 1, pp. 30–40, Jan. 2006.
- [20] A. O. Hero, D. A. Castañón, D. Cochran, and K. Kastella *Foundations and Applications of Sensor Management*. New York, NY, USA: Springer Publishing, 2008.
- [21] A. O. Hero and D. Cochran “Sensor management: past, present, and future,” *IEEE Sensors J.*, vol. 11, no. 12, pp. 3064–3075, Dec. 2011.
- [22] A. Irci, A. Saranlı, and B. Baykal “Study on Q-RAM and feasible directions based methods for resource management in phased array radar systems,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, no. 4, pp. 1848–1864, Oct. 2010.
- [23] P. R. Kalata “The tracking index: a generalized parameter for alpha–beta and alpha–beta–gamma target trackers,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 20, no. 2, pp. 174–182, Mar. 1984.
- [24] F. Katsilieris “Sensor Management for Surveillance and Tracking: An Operational Perspective,” Ph.D. thesis, Delft, The Netherlands, 2015.
- [25] F. Katsilieris, H. Driessen, and A. Yarovoy “Threat-based sensor management for target tracking,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 4, pp. 2772–2785, Oct. 2015.
- [26] R. Klemm, H. Griffiths, and W. Koch *Novel Radar Techniques and Applications, Volume 2 – Waveform Diversity and Cognitive Radar, and Target Tracking and Data Fusion*. London, U.K.: Scitech Publishing, 2017.
- [27] W. Koch “On adaptive parameter control for phased-array tracking,” in *Proc. SPIE’s Int. Symp. Opt. Sci., Eng., Instrum.*, vol. 3809, 1999, pp. 444–455.
- [28] C. Kreucher, K. Kastella, and A. O. Hero “Sensor management using an active sensing approach,” *Signal Process.*, vol. 85, no. 3, pp. 607–624, Mar. 2005.
- [29] V. Krishnamurthy “POMDP Sensor Scheduling with Adaptive Sampling,” in *Proc. 17th Int. Conf. Inf. Fusion*, Jul. 2014, pp. 1–7.
- [30] V. Krishnamurthy and D. V. Djonin “Optimal threshold policies for multivariate POMDPs in radar resource management,” *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3954–3969, Oct. 2009.
- [31] F. L. Lewis, D. L. Vrabie, and V. L. Syrmos *Optimal Control*. Hoboken, NJ, USA: Wiley, 2012.
- [32] H. Meikle *Modern Radar Systems*. Norwood, MA, USA: Artech House, 2008.
- [33] H. S. Mir and A. Guitouni “Variable dwell time task scheduling for multifunction radar,” *IEEE Trans. Automat. Sci. Eng.*, vol. 11, no. 2, pp. 463–472, Apr. 2014.
- [34] P. W. Moo and Z. Ding *Adaptive Radar Resource Management*. London, U.K.: Academic Press, 2015.
- [35] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa “Online planning algorithms for POMDPs,” *J. Artif. Intell. Res.*, vol. 32, no. 1, pp. 663–704, Jul. 2008.
- [36] M. I. Schöpe, H. Driessen, and A. Yarovoy “Optimal balancing of multi-function radar budget for multi-target tracking using lagrangian relaxation,” in *Proc. 22nd Int. Conf. Inf. Fusion*, Jul. 2019.
- [37] M. I. Schöpe, H. Driessen, and A. Yarovoy “Multi-task sensor resource balancing using Lagrangian relaxation and policy rollout,” in *Proc. 23rd Int. Conf. Inf. Fusion*, Jul. 2020.
- [38] E. Selvi, R. M. Buehrer, A. Martone, and K. Sherbondy

- “Reinforcement learning for adaptable bandwidth tracking radars,”
IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 5, pp. 3904–3921, Oct. 2020.
- [39] M. Shaghghi and R. S. Adve
 “Task selection and scheduling in multifunction multichannel radars,”
 in *Proc. 2017 IEEE Radar Conf.*, May 2017, pp. 969–974.
- [40] R. D. Smallwood and E. J. Sondik
 “The optimal control of partially observable Markov processes over a finite horizon,”
Oper. Res., vol. 21, no. 5, pp. 1071–1088, 1973.
- [41] M. T. J. Spaan and N. Vlassis
 “Perseus: randomized point-based value iteration for POMDPs,”
J. Artif. Intell. Res., vol. 24, pp. 195–220, 2005.
- [42] C. E. Thornton, M. A. Kozy, R. M. Buehrer, A. F. Martone, and K. D. Sherbondy
 “Deep reinforcement learning control for radar detection and tracking in congested spectral environments,”
IEEE Trans. Cogn. Commun. Netw., vol. 6, no. 4, pp. 1335–1349, Dec. 2020.
- [43] K. A. B. White, J. L. Williams, and P. Hoffensetz
 “Radar sensor management for detection and tracking,”
 in *Proc. 2008 11th Int. Conf. Inf. Fusion*, Jun. 2008, pp. 1–8.
- [44] K. A. B. White and J. L. Williams
 “Lagrangian relaxation approaches to closed loop scheduling of track updates,”
 in *Proc. SPIE’s Int. Symp. Opt. Sci., Eng., Instrum.*, 2012, pp. 8393–8393.
- [45] J. L. Williams, J. W. Fisher, and A. S. Willsky
 “Approximate Dynamic Programming for Communication-Constrained Sensor Network Management,”
IEEE Trans. Signal Process., vol. 55, no. 8, pp. 4300–4311, Aug 2007.
- [46] J. Wintenby and V. Krishnamurthy
 “Hierarchical resource management in adaptive airborne surveillance radars,”
IEEE Trans. Aerosp. Electron. Syst., vol. 42, no. 2, pp. 401–420, Apr. 2006.



Max Ian Schöpe received the B.Eng. degree in electrical engineering from the University of Applied Sciences in Hamburg, Germany, in February 2015. Subsequently, he started the M.Sc. program of telecommunication and sensing systems at TU Delft, the Netherlands, which he finished in October 2017. After his graduation, he started as a Ph.D. student in the Microwave Sensing, Signals and Systems group of TU Delft. His main research interests are in the areas of radar resource management using a partially observable Markov decision process framework.



Hans Driessen obtained the M.Sc. and Ph.D. degree in 1987 and 1992, respectively, both from the Department of Electrical Engineering, TU Delft. Since then he has been employed with Thales Nederland BV. He has been and is still involved with various (international) research projects and radar development programs in the area of signal/data processing and radar management. Since January 2015, he has been additionally holding a part-time position as Associate Professor at the EEMCS faculty of TU Delft in the Microwave Signals Sensor and Systems group in the field of radar systems, waveforms, and processing. His interest is in the practical application of detection, estimation, information, and control theory to various problems in sensor systems.



Alexander G. Yarovoy (FIEEE 2015) graduated from Kharkov State University, Ukraine, in 1984 with the Diploma with honors in radiophysics and electronics. He received the Candidate Phys. & Math. Sci. and Doctor Phys. & Math. Sci. degrees in radiophysics in 1987 and 1994, respectively. In 1987, he joined the Department of Radiophysics, Kharkov State University, as a Researcher and became a Full Professor there in 1997. From September 1994 through 1996, he was with the Technical University of Ilmenau, Germany, as a Visiting Researcher. Since 1999, he has been with the Delft University of Technology, the Netherlands. Since 2009, he has been leading there as Chair of Microwave Sensing, Systems and Signals. His main research interests are in high-resolution radar, microwave imaging, and applied electromagnetics (in particular, UWB antennas). He has authored and co-authored more than 450 scientific or technical papers, 6 patents, and 14 book chapters. He is the recipient of the European Microwave Week Radar Award for the paper that best advances the state-of-the-art in radar technology in 2001 (together with L. P. Ligthart and P. van Genderen) and in 2012 (together with T. Savelyev). In 2010, together with D. Caratelli, Prof. Yarovoy got the best paper award of the Applied Computational Electromagnetic Society (ACES). Prof. Yarovoy served as the General TPC chair of the 2020 European Microwave Week (EuMW'20), as the Chair and TPC Chair of the 5th European Radar Conference (EuRAD'08), as well as the Secretary of the 1st European Radar Conference (EuRAD'04). He also served as the Co-Chair and TPC Chair of the Xth International Conference on GPR (GPR 2004). He served as an Associated Editor for the *International Journal of Microwave and Wireless Technologies* from 2011 to 2018 and as a Guest Editor for five special issues of the IEEE Transactions and other journals. During the period 2008–2017, Prof. Yarovoy served as Director of the European Microwave Association (EuMA).

Improvement of Proportional Conflict Redistribution Rules of Combination of Basic Belief Assignments

THÉO DEZERT
JEAN DEZERT
FLORENTIN SMARANDACHE

This paper discusses and analyzes the behaviors of the Proportional Conflict Redistribution rules no. 5 (PCR5) and no. 6 (PCR6) to combine several distinct sources of evidence characterized by their basic belief assignments defined over the same frame of discernment. After a brief review of these rules, the paper shows through simple examples why their behaviors can sometimes increase the uncertainty more than necessary, which is detrimental to decision-making support drawn from the result of the combination. We present a theoretical improvement of these rules, and establish new PCR5⁺ and PCR6⁺ rules of combination. These new rules overcome the weakness of PCR5 and PCR6 rules by computing binary-keeping indexes that allow to keep only focal elements that play an effective role in the partial conflict redistribution. PCR5⁺ and PCR6⁺ rules are not associative but they preserve the neutrality of the vacuous belief assignment contrary to the PCR5 and PCR6 rules, and they make a more precise redistribution which does not increase improperly the mass of partial uncertainties.

Manuscript received July 16, 2021; released for publication June 30, 2021.

Théo Dezert was with Gustave Eiffel University, GERS-GeoEND, Bouguenais, France (E-mail: theo.dezert@univ-eiffel.fr).

Jean Dezert is with The French Aerospace Lab (ONERA/DTIS), Palaiseau, France (E-mail: jean.dezert@onera.fr).

Florentin Smarandache is with Department of Mathematics, University of New Mexico, Gallup, NM, USA (E-mail: smarand@unm.edu).

1557-6418/21/\$17.00 © 2021 JAIF

I. INTRODUCTION

There exist different theories based on distinct representations and modelings of uncertainty to deal with uncertain information to conduct information fusion [1]. The theory of probability [2], [3], the theory of fuzzy sets [4], [5], the possibility theory [6], [7], and the theory of belief functions [8]–[10] are the most well-known ones. This paper addresses the problem of information fusion in the mathematical framework of the belief functions introduced by Shafer from Dempster's works [11], [12]. The belief functions are often used in decision-making support applications because the experts are generally able to express only a belief in a hypothesis (or a set of hypotheses) from their partial knowledge, experience, and from their own perception of the reality. To conduct information fusion, we need some efficient rules of combination that are able to manage the conflicting sources of evidence (if any), or expert opinions expressed in terms of belief functions. Readers interested in belief functions can find classical related papers in [13] and in the special issue [14], which includes also a list of good selected papers. It is worth to mention that the recent book of Cuzzolin [15] includes 2137 references, with many of them related to belief functions.

In this paper, we adopt the notion of conflict introduced by Shafer in [8] (p. 65). This notion of conflict is often adopted by researchers working with belief functions, as in [16] (p. 17) for instance, because this notion is quite simple to understand. Different definitions and interpretations of conflict can be also found in [17]–[27] for readers interested in this topic. In this paper, two (or more) sources are said conflicting if they support incompatible (disjoint, or contradictory) hypotheses. We also work with distinct sources of evidences that are considered as (cognitively) independent and reliable. We neither consider, nor apply discounting techniques of belief assessments listed in [14] before combining them to keep the presentation and notations as simple as possible.¹

While the conjunctive rule makes it possible to combine information between different sources of information by estimating the level of existing conflict, Dempster–Shafer (DS) rule [8], [16] proposes a distribution of this conflict on the hypotheses characterized by the sources of information. The normalization carried out by the DS rule may, however, be considered counter-intuitive especially when the level of conflict between the sources of information is high [28], [29], but also in some situations where the level of conflict between sources is low as shown in [30] showing a dictatorial behavior of DS rule. The Proportional Conflict Redistribution rules no. 5 (PCR5) [31] and no. 6 (PCR6) [32], [33] have been proposed to circumvent the problem of the DS rule to make a more judicious management of the conflict.

¹Of course discounted belief assignments can also be combined by the rules presented in this paper.

In this paper, we put forward a flawed behavior of these combination rules in some cases attributed to the non-neutrality of the vacuous Basic Belief Assignment (BBA), and we propose an improvement of these two combination rules (denoted by PCR5⁺ and PCR6⁺) in order to ensure the neutrality property of the vacuous BBA (VBBA). This is achieved by discarding specific elements implied in the partial conflict and which are not useful for making the conflict redistribution.

In the Proportional Conflict Redistribution (PCR) rules [32]–[34], one redistributes the product of masses of belief of incompatible (i.e., conflicting) elements whose intersection is empty only to elements involved in this product and proportionally to their mass of belief. For instance, let's consider two elements A and B of a frame of discernment (FoD) with $A \cap B = \emptyset$, and three BBAs $m_1(\cdot)$, $m_2(\cdot)$, and $m_3(\cdot)$ defined on this FoD with $m_1(A) > 0$, $m_2(B) > 0$, and $m_3(A \cup B) > 0$. The product $m_1(A)m_2(B)m_3(A \cup B) > 0$ is called a conflicting product hereafter because $A \cap B \cap (A \cup B) = \emptyset$. Based on PCR5 (and PCR6) rule, we will redistribute the value of this product back to the focal² elements A , B , and $A \cup B$, and proportionally to $m_1(A)$, $m_2(B)$, and $m_3(A \cup B)$. In the improved PCR rules developed in this paper, we will redistribute this conflicting product only to the focal elements A and B since the focal element $A \cup B$ is neither in conflict with A , nor with B . Such an improvement in the PCR is made possible by defining a binary-keeping index for each focal element involved in the conflicting product. This index will allow the identification of elements of the conflicting product that will have an effective role in the proportional redistribution of conflicting product. All elements (if any) having a binary-keeping index equal to zero are discarded of the conflict redistribution process. This main idea is developed in this paper and illustrated with several examples. It allows to preserve the neutrality of the total ignorant source of evidence in the improved versions of PCR5 and PCR6 rules, which is often considered as a desirable property for a rule of combination of distinct and reliable sources of evidence.

For the reader not immersed in the belief mathematics notion, the comparative numerical examples of Example 1 of Section III-B as compared with Example 1 revisited of Section VII, provide a quick verification of the improvements.

This paper is organized as follows. We give the basics of belief functions in Section II. We present the PCR5 and PCR6 rules of combination in Section III with new general formulas in Subsection III-C, and associated examples in Section IV. The flawed behavior of PCR5 and PCR6 rules are highlighted in Section V through specific examples. Then, Section VI proposes the mathematical expression of the new improved PCR5⁺ and PCR6⁺

rules of combination, as well as the very detailed procedure to select the focal elements for these new proportional redistributions. Finally, comparative results for relevant examples are shown in Section VII in order to compare the PCR5 and PCR6 results with the PCR5⁺ and PCR6⁺ results. Concluding remarks are given in Section VIII. For convenience, two MatlabTM routines are also given in Appendix 3 of this paper for PCR5⁺ and PCR6⁺ rules of combination.

II. BASICS OF BELIEF FUNCTIONS

We consider a given finite set Θ of $n > 1$ distinct elements $\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}$ corresponding to the FoD of the fusion problem, or the decision-making problem, under concern. All elements of Θ are mutually exclusive³ and each element is an elementary choice of the potential decision to take. The power set of Θ is the set of all subsets of Θ (including empty set \emptyset and Θ) and it is usually denoted 2^Θ because its cardinality equals $2^{|\Theta|}$. We adopt Shafer's formalism whereby propositions are represented by subsets [8] (Chap. 2, pp. 35–37). Hence, the propositions under concern are in one-to-one correspondence with subsets of Θ . We also use classical notations of set theory [35], i.e. \emptyset for the empty set, $A \cup B$ for the union⁴ of sets A and B (which is the set of all objects that are a member of the set A , or the set B , or both), $A \cap B$ for their intersection (which is the set of all objects that are members of both A and B), etc. A BBA given by a source of evidence is defined by Shafer [8] in his Mathematical Theory of Evidence (known also as Dempster–Shafer Theory (DST)) as $m(\cdot) : 2^\Theta \rightarrow [0, 1]$ satisfying

$$\begin{cases} m(\emptyset) = 0 \\ \sum_{A \in 2^\Theta} m(A) = 1, \end{cases} \quad (1)$$

where $m(A)$ is the mass of belief exactly committed to A , what we usually call the mass of A . A BBA is said proper (or normal) if it satisfies Shafer's definition (1). The subset $A \subseteq \Theta$ is called a focal element of the BBA $m(\cdot)$ if and only if $m(A) > 0$. The empty set is not a focal element of a BBA because $m(\emptyset) = 0$ according to definition (1). The set of all focal elements of a BBA $m(\cdot)$ is denoted $\mathcal{F}(m)$. Its mathematical definition is $\mathcal{F}(m) = \{X \in 2^\Theta | m(X) > 0\}$. The cardinality $|\mathcal{F}(m)|$ of the set $\mathcal{F}(m)$ is denoted \mathcal{F}_m . The order of focal elements of $\mathcal{F}(m)$ does not matter and all the focal elements are different. The set $\mathcal{F}(m)$ of focal elements of $m(\cdot)$ has at least one focal element, and at most $2^{|\Theta|} - 1$ focal elements.

³This standard assumption is called *Shafer's model of FoD* in Dezert–Smarandache theory (DSmT) framework [34].

⁴We prefer the notation $A \cup B$ for denoting the union of sets A and B , which is a formal mathematical notation for the union of two sets, instead of the notations AB or $\{A, B\}$ used by some authors.

²A focal element is an element (i.e., a subset) having a strictly positive mass of belief committed to it—see Section II elements.

Belief and plausibility functions are, respectively, defined from $m(\cdot)$ by [8]

$$Bel(A) = \sum_{X \in 2^\Theta | X \subseteq A} m(X) \quad (2)$$

and

$$Pl(A) = \sum_{X \in 2^\Theta | A \cap X \neq \emptyset} m(X) = 1 - Bel(\bar{A}). \quad (3)$$

where \bar{A} represents the complement of A in Θ .

$Bel(A)$ and $Pl(A)$ are usually interpreted, respectively, as lower and upper bounds of an unknown (subjective) probability measure $P(A)$ [11], [12]. The functions $m(\cdot)$, $Bel(\cdot)$ and $Pl(\cdot)$ are one-to-one. A belief function $Bel(\cdot)$ is *Bayesian* if all Bel's focal elements are singletons [8] (Theorem 2.8, p. 45). In this case, $m(X) = Bel(X)$ for any (singleton) focal element X , and $m(\cdot)$ is called a *Bayesian BBA*. Corresponding $Bel(\cdot)$ function is equal to $Pl(\cdot)$ and these functions can be interpreted as a same (possibly subjective) probability measure $P(\cdot)$. The VBBA representing a totally ignorant source is defined as $m_v(\Theta) = 1$.

III. COMBINATION OF BBAS

This section presents at first the conjunctive rule of combination which is one of the main rules to combine reliable sources of evidence and which allows to identify the conflicting information among the sources. Then we present the PCR5 [31] and PCR6 [32], [33] as alternatives of Dempster's rule of combination [8]. The development of these rules has been motivated by the counter-intuitive behavior of Dempster's rule [8] when combining high conflicting sources of evidences, but also when combining low conflicting sources of evidences as well⁵. The reader interested in this topic can refer to [13], [28]–[30] to see theoretical justifications and examples. In the following, and for simplicity, we restrain our presentation to the classical framework of belief functions, and we work with BBAs defined only on the power set 2^Θ of a FoD Θ . PCR rules have been defined originally for working with Dedekind's lattice as well, see Chapter 1 of [34] (Volume 2). In this paper, we present simple general expressions of PCR5 and PCR6 fusion rules because they are easier to understand than the original general formulas, and they afford expressions of the improved PCR5⁺ and PCR6⁺ rules in a direct and useful manner.

After a brief presentation of the main notations used in this paper, we will recall both PCR5 and PCR6 rules for historical and technical reasons. PCR5 has been developed at first, and then PCR6 has been proposed based on a modified redistribution principle inspired by PCR5. In this paper, we follow the logical and historical development of these PCR5 and PCR6 rules to make

the presentation of their improved versions PCR5⁺ and PCR6⁺. It seems easier to understand PCR6⁺ fusion formula once the PCR5⁺ formula will have been established. By presenting both rules, we offer to the readers a global deeper view on how these new rules work and their fundamental and mathematical differences in their conflict redistribution principles. In the sequel, all the introduced examples assume the model of Shafer's FoD as in the classical DST framework.

A. Notations

When we make the combination of $S \geq 2$ BBAs by the conjunctive rule, or by the PCR5 and PCR6 fusion rules, we have to compute the product of the masses of the focal elements composing any possible S -tuple of focal elements. Each possible S -tuple is noted by⁶

$$\mathbf{X}_j \triangleq (X_{j_1}, X_{j_2}, \dots, X_{j_S}) \in \mathcal{F}(m_1) \times \mathcal{F}(m_2) \times \dots \times \mathcal{F}(m_S),$$

where $j_1 \in \{1, 2, \dots, \mathcal{F}_{m_1}\}$, $j_2 \in \{1, 2, \dots, \mathcal{F}_{m_2}\}$, ..., $j_S \in \{1, 2, \dots, \mathcal{F}_{m_S}\}$. The element X_{j_i} is the focal element of $m_i(\cdot)$ that makes the i -th component of the j -th S -tuple \mathbf{X}_j .

For notation convenience also, the cartesian product $\mathcal{F}(m_1) \times \mathcal{F}(m_2) \times \dots \times \mathcal{F}(m_S)$ is denoted by $\mathcal{F}(m_1, \dots, m_S)$ in the sequel.

We have $\mathcal{F} \triangleq |\mathcal{F}(m_1, \dots, m_S)| = \prod_{i=1}^S |\mathcal{F}(m_i)| = \prod_{i=1}^S \mathcal{F}_{m_i}$ products of masses of focal elements to consider and to calculate because we have \mathcal{F}_{m_1} focal elements in $\mathcal{F}(m_1)$, \mathcal{F}_{m_2} focal elements in $\mathcal{F}(m_2)$, ..., and \mathcal{F}_{m_S} focal elements in $\mathcal{F}(m_S)$. Each product for $j = 1$ to \mathcal{F} is of the form

$$\pi_j(X_{j_1} \cap X_{j_2} \cap \dots \cap X_{j_S}) \triangleq \prod_{i=1}^S m_i(X_{j_i}). \quad (4)$$

There are two types of products:

- $\pi_j(X_{j_1} \cap X_{j_2} \cap \dots \cap X_{j_S})$ is called a *non-conflicting (mass) product* if

$$X_{j_1} \cap X_{j_2} \cap \dots \cap X_{j_S} = X \neq \emptyset.$$

In this case, $\pi_j(X_{j_1} \cap X_{j_2} \cap \dots \cap X_{j_S})$ is also noted by $\pi_j(X)$ for short.

- $\pi_j(X_{j_1} \cap X_{j_2} \cap \dots \cap X_{j_S})$ is called a *conflicting (mass) product* if

$$X_{j_1} \cap X_{j_2} \cap \dots \cap X_{j_S} = \emptyset.$$

In this case, $\pi_j(X_{j_1} \cap X_{j_2} \cap \dots \cap X_{j_S})$ is also noted by $\pi_j(\emptyset)$ for short.

It is worth noting that an element $X \in 2^\Theta \setminus \{\emptyset\}$ may belong to sets of focal elements of the different BBAs to combine, and therefore a S -tuple \mathbf{X}_j can have duplicate components. Because all the BBAs are normalized, we

⁵Which is known as the dictatorial behavior of Dempster's rule [30].

⁶The symbol \triangleq means "equals by definition."

always have

$$\sum_{j=1}^{\mathcal{F}} \pi_j(X_{j_1} \cap X_{j_2} \cap \dots \cap X_{j_s}) = 1. \quad (5)$$

As a simple example to illustrate our notations, let us consider two BBAs $m_1(\cdot)$ and $m_2(\cdot)$ defined over the FoD $\Theta = \{A, B, C\}$ with, respectively, two and three focal elements, say $\mathcal{F}(m_1) = \{A, B \cup C\}$ and $\mathcal{F}(m_2) = \{B, C, A \cup C\}$. Here $\mathcal{F}_{m_1} = |\mathcal{F}(m_1)| = 2$ and $\mathcal{F}_{m_2} = |\mathcal{F}(m_2)| = 3$. For $j_1 = 1$ (the first focal element of $m_1(\cdot)$) one has $X_{j_1} = A$, and for $j_1 = 2$ (the second focal element of $m_1(\cdot)$) one has $X_{j_1} = B \cup C$. Similarly, for $j_2 = 1$ (the first focal element of $m_2(\cdot)$) one has $X_{j_2} = B$, for $j_2 = 2$ (the 2nd focal element of $m_2(\cdot)$) one has $X_{j_2} = C$, and $j_2 = 3$ (the 3rd focal element of $m_2(\cdot)$) one has $X_{j_2} = A \cup C$. In this case we have $\mathcal{F} = \mathcal{F}_{m_1} \cdot \mathcal{F}_{m_2} = 6$ products of masses to consider in the conjunctive fusion rule (see next sub-section) which are

$$\begin{aligned} \pi_1(A \cap B) &= m_1(A)m_2(B), \\ \pi_2(A \cap C) &= m_1(A)m_2(C), \\ \pi_3(A \cap (A \cup C)) &= m_1(A)m_2(A \cup C), \\ \pi_4((B \cup C) \cap B) &= m_1(B \cup C)m_2(B), \\ \pi_5((B \cup C) \cap C) &= m_1(B \cup C)m_2(C), \\ \pi_6((B \cup C) \cap (A \cup C)) &= m_1(B \cup C)m_2(A \cup C). \end{aligned}$$

The products π_1 and π_2 are called *conflicting products* because

- for π_1 , the focal elements A and B involved in π_1 are incompatible (i.e., disjoint) because $A \cap B = \emptyset$. $\pi_1(A \cap B)$ is of course equivalent to $\pi_j(X_{j_1} \cap X_{j_2})$ with $j = 1$ by taking $X_{j_1} = A$ and $X_{j_2} = B$; and
- for π_2 , one has $A \cap C = \emptyset$. $\pi_2(A \cap C)$ is equivalent to $\pi_j(X_{j_1} \cap X_{j_2})$ with $j = 2$ by taking $X_{j_1} = A$ and $X_{j_2} = C$, etc.

The products π_3, \dots , and π_6 are not conflicting products because the focal elements involved in each product have non-empty intersection. Because $m_1(A) + m_1(B \cup C) = 1$ and $m_2(B) + m_2(C) + m_2(A \cup C) = 1$, one has $(m_1(A) + m_1(B \cup C))(m_2(B) + m_2(C) + m_2(A \cup C)) = 1$, and therefore $\sum_{j=1}^6 \pi_j = 1$. This illustrates the formula (5).

In this paper, $i \in \{1, \dots, S\}$ represents the index of the i -th source of evidence characterized by the BBA $m_i(\cdot)$, and $j \in \{1, \dots, \mathcal{F}\}$ represents the index of the j -th product $\pi_j(X_{j_1} \cap X_{j_2} \cap \dots \cap X_{j_s})$.

B. The conjunctive rule of combination

Let's consider $S \geq 2$ distinct reliable sources of evidence characterized by their BBA $m_s(\cdot)$ ($s = 1, \dots, S$)

defined on 2^Θ . Their conjunctive fusion⁷ is defined for all $A \in 2^\Theta$ by

$$\begin{aligned} m_{1,2,\dots,S}^{\text{Conj}}(A) &= \sum_{\substack{\mathbf{X}_j \in \mathcal{F}(m_1, \dots, m_S) \\ X_{j_1} \cap \dots \cap X_{j_s} = A}} \pi_j(X_{j_1} \cap X_{j_2} \cap \dots \cap X_{j_s}) \\ &= \sum_{\substack{\mathbf{X}_j \in \mathcal{F}(m_1, \dots, m_S) \\ X_{j_1} \cap \dots \cap X_{j_s} = A}} \prod_{i=1}^S m_i(X_{j_i}). \end{aligned} \quad (6)$$

The symbol \odot is also used in the literature, for instance in [36], to note the conjunctive fusion operator, i.e., $m_{1,2,\dots,S}^{\text{Conj}}(A) = [m_1 \odot m_2 \odot \dots \odot m_S](A)$.

The total conflicting mass between the S sources of evidence, denoted $m_{1,2,\dots,S}^{\text{Conj}}(\emptyset)$, is nothing but the sum of all existing conflicting mass products, that is

$$\begin{aligned} m_{1,2,\dots,S}^{\text{Conj}}(\emptyset) &= \sum_{\substack{\mathbf{X}_j \in \mathcal{F}(m_1, \dots, m_S) \\ X_{j_1} \cap \dots \cap X_{j_s} = \emptyset}} \pi_j(X_{j_1} \cap X_{j_2} \cap \dots \cap X_{j_s}) \\ &= 1 - \sum_{A \in 2^\Theta \setminus \{\emptyset\}} m_{1,2,\dots,S}^{\text{Conj}}(A). \end{aligned} \quad (7)$$

Note that the combined BBA $m_{1,2,\dots,S}^{\text{Conj}}(\cdot)$ given in (6) is not a proper BBA because it does not satisfy Shafer's definition (1). In general, the S sources of evidence to combine do not fully agree, and we have consequently $m_{1,2,\dots,S}^{\text{Conj}}(\emptyset) > 0$.

Dempster's rule of combination (called also *orthogonal sum* by Shafer [8], p. 6) coincides with the normalized version of the conjunctive rule. It is defined by $m_{1,2,\dots,S}^{\text{DS}}(A) = m_{1,2,\dots,S}^{\text{Conj}}(A)/(1 - m_{1,2,\dots,S}^{\text{Conj}}(\emptyset))$, assuming $m_{1,2,\dots,S}^{\text{Conj}}(\emptyset) \neq 1$. The DS upper notation refers to initials of Dempster and Shafer names because Dempster's rule has gained its popularity through Shafer's works on belief functions. Shafer uses the symbol \oplus to note Dempster's fusion operator, i.e., $m_{1,2,\dots,S}^{\text{DS}}(A) = [m_1 \oplus m_2 \oplus \dots \oplus m_S](A)$ for $A \neq \emptyset$, and $m_{1,2,\dots,S}^{\text{DS}}(\emptyset) = 0$. A probabilistic analysis of Dempster's rule of combination can be found in [37], and the geometry of Dempster's rule is analyzed in [38].

Example 1: Consider $\Theta = \{A, B\}$ and two following BBAs

$$m_1(A) = 0.1 \quad m_1(B) = 0.2 \quad m_1(A \cup B) = 0.7$$

$$m_2(A) = 0.4 \quad m_2(B) = 0.3 \quad m_2(A \cup B) = 0.3$$

We have $m_{1,2}^{\text{Conj}}(\emptyset) = 0.11$, and

$$m_{1,2}^{\text{Conj}}(A) = 0.35, \quad m_{1,2}^{\text{Conj}}(B) = 0.33, \quad m_{1,2}^{\text{Conj}}(\Theta) = 0.21.$$

⁷The conjunctive fusion rule is also called Smets' rule of combination by some authors because it has been widely used by Philippe Smets in his works related to belief functions. But Smets himself call it conjunctive rule, see his last paper [20] (p. 388).

Symbolically we denote the conjunctive fusion of S sources as $m_{1,2,\dots,S}^{\text{Conj}} = \text{Conj}(m_1, m_2, \dots, m_S)$. This conjunctive rule is commutative and associative. This means that the sources can be combined altogether in one step, or sequentially in any order and it does not matter. Also, the total ignorant source represented by the vacuous (non-informative) BBA has no impact in the fusion result—see Lemma 1 below.

Lemma 1: The VBBA m_v has a neutral impact in the conjunctive rule of combination, that is

$$\text{Conj}(m_1, m_2, \dots, m_S, m_v) = \text{Conj}(m_1, m_2, \dots, m_S). \quad (8)$$

Proof: see Appendix 1.

The main drawback of this fusion rule is that it does not generate a proper BBA because $m_{1,2,\dots,S}^{\text{Conj}}(\emptyset) > 0$ in general, and also it can provide a fusion result $m_{1,2,\dots,S}^{\text{Conj}}(\emptyset)$ that quickly tends to one after only few steps of a sequential fusion processing of the sources which is not very useful for decision-making support. This is because the empty set \emptyset is the absorbing element for the conjunctive operation since $\emptyset \cap A = \emptyset$ for all $A \in 2^\Theta$ so that the mass committed to the empty set always increases through the repeated conjunctive fusion rule. The main interest of this rule is its ability to identify the partial conflicts and to provide a measure of the total level of conflict $m_{1,2,\dots,S}^{\text{Conj}}(\emptyset)$ between the sources which can be used to manage (select or discard) the sources in the fusion process if one prefers, see [39] for an application in geophysics for instance.

C. PCR5 and PCR6 rules of combination

The PCR rules have been developed originally in the framework of DSMT [31], [32], [34] but they can work also in the classical framework of Shafer's belief functions as well. Six rules have been proposed and they are referred as PCR1, ..., PCR6 rules of combination having different complexities, PCR1 being the most simplest (but less effective) one. All these rules share the same general principle which consists of three steps:

- apply the conjunctive rule (6);
- calculate the conflicting mass products $\pi_j(\emptyset)$; and
- redistribute the conflicting mass products $\pi_j(\emptyset)$ proportionally on all non-empty sets involved in the conflict.

The way the conflicting mass product $\pi_j(\emptyset)$ is redistributed yields to different versions of PCR combination rules that work for any degree of conflict. The sophistication/complexity and preciseness of PCR rules increases from the first PCR1 rule up to the last rule PCR6. The main disadvantage of these rules, aside their complexity, is their non-associativity properties which impose to combine all the BBAs altogether with PCR rules rather than sequentially to expect the best fusion result.

In this paper, we focus on the presentation of PCR5 and PCR6 only because they are the most well-known advanced fusion rules used so far in the belief functions community. A detailed presentation of other rules of combination encountered in the literature can be found in [40]. Symbolically, the PCR5 fusion and the PCR6 fusion of $S \geq 2$ BBAs are, respectively, denoted $m_{1,2,\dots,S}^{\text{PCR5}} = \text{PCR5}(m_1, m_2, \dots, m_S)$, and $m_{1,2,\dots,S}^{\text{PCR6}} = \text{PCR6}(m_1, m_2, \dots, m_S)$.

Readers familiar with PCR rules could quickly read the example 1 given in section III-B, and the results obtained with classical and improved PCR5 and PCR6 rules in section VII to appreciate the discussion throughout the paper.

The PCR5 rule of combination [31]: This rule transfers the conflicting mass $\pi_j(\emptyset)$ to all the elements involved in this conflict and proportionally to their individual masses, so that a more sophisticated and specific distribution is done with the PCR5 fusion process with respect to other existing rules (including Dempster's rule). The PCR5 rule is presented in details (with justification and examples) in [34] (Vol. 2 and Vol. 3).

- The PCR5 fusion of two BBAs is obtained by $m_{1,2}^{\text{PCR5}}(\emptyset) = 0$, and for all $A \in 2^\Theta \setminus \{\emptyset\}$ by

$$m_{1,2}^{\text{PCR5}}(A) = m_{1,2}^{\text{Conj}}(A) + \sum_{\substack{X \in 2^\Theta \\ X \cap A = \emptyset}} \left[\frac{m_1(A)^2 m_2(X)}{m_1(A) + m_2(X)} + \frac{m_2(A)^2 m_1(X)}{m_2(A) + m_1(X)} \right], \quad (9)$$

where $m_{1,2}^{\text{Conj}}(A)$ is the conjunctive rule formula (6) with $S = 2$, and where all denominators in (9) are different from zero. If a denominator is zero, that fraction is discarded. All propositions/sets are in a canonical form. We take the disjunctive normal form, which is a disjunction of conjunctions, and it is unique in Boolean algebra and simplest. For example, $X = A \cap B \cap (A \cup B \cup C)$ it is not in a canonical form, but we simplify the formula and $X = A \cap B$ is in a canonical form.

The PCR5 formula (9) for two BBAs can also be expressed by considering only the focal elements of $m_1(\cdot)$ and $m_2(\cdot)$ as follows

$$m_{1,2}^{\text{PCR5}}(A) = m_{1,2}^{\text{Conj}}(A) + \sum_{\substack{(X_{j_1}, X_{j_2}) \in \mathcal{F}(m_1) \times \mathcal{F}(m_2) \\ X_{j_1} \cap X_{j_2} = \emptyset \\ X_{j_1} = A}} m_1(X_{j_1}) \cdot \frac{m_1(X_{j_1}) m_2(X_{j_2})}{m_1(X_{j_1}) + m_2(X_{j_2})} + \sum_{\substack{(X_{j_1}, X_{j_2}) \in \mathcal{F}(m_1) \times \mathcal{F}(m_2) \\ X_{j_1} \cap X_{j_2} = \emptyset \\ X_{j_2} = A}} m_2(X_{j_2}) \cdot \frac{m_1(X_{j_1}) m_2(X_{j_2})}{m_1(X_{j_1}) + m_2(X_{j_2})}, \quad (10)$$

or equivalently, with shorthand π_j notations, as

$$m_{1,2}^{\text{PCR5}}(A) = m_{1,2}^{\text{Conj}}(A) + \sum_{\substack{j \in \{1, \dots, \mathcal{F}\} | \mathbf{X}_j \in \mathcal{F}(m_1, m_2) \\ X_{j_1} \cap X_{j_2} = \emptyset \\ A \in \mathbf{X}_j}} \left[m_{i \in \{1,2\} | X_{j_i} = A}(X_{j_i}) \cdot \frac{\pi_j(X_{j_1} \cap X_{j_2})}{m_1(X_{j_1}) + m_2(X_{j_2})} \right], \quad (11)$$

where $\mathcal{F} = |\mathcal{F}(m_1)| \cdot |\mathcal{F}(m_2)|$ is the total number of products $\pi_j(X_{j_1} \cap X_{j_2}) = m_1(X_{j_1})m_2(X_{j_2})$, and $A \in \mathbf{X}_j$ means that at least one component of \mathbf{X}_j equals A .

- The explicit formula of the PCR5 fusion of three BBAs is given in [41].

- A simple formulation of the general expression of the PCR5 fusion of $S > 2$ BBAs is obtained by redistributing each conflicting product $\pi_j(\emptyset) = \pi_j(X_{j_1} \cap \dots \cap X_{j_s} = \emptyset) = \prod_{i=1}^S m_i(X_{j_i})$ to some elements of the power set of the FoD that are involved in the conflict. Each $\pi_j(\emptyset)$ is redistributed proportionally to elements involved in this conflict based on the PCR5 redistribution principle. When an element $A \in 2^\Theta$ is not involved in a conflicting product $\pi_j(\emptyset)$, i.e. $A \notin \mathbf{X}_j$, the conflicting product $\pi_j(\emptyset)$ is not redistributed to A . If an element A is involved in the conflict $X_{j_1} \cap \dots \cap X_{j_s} = \emptyset$, i.e. $A \in \mathbf{X}_j$ and $\pi_j(\emptyset)$ occur, then the proportional redistribution of $\pi_j(\emptyset)$ to A is given by

$$x_j(A) \triangleq \left(\prod_{i \in \{1, \dots, S\} | X_{j_i} = A} m_i(X_{j_i}) \right) \cdot \frac{\pi_j(\emptyset)}{\sum_{X \in \mathbf{X}_j} \left(\prod_{i \in \{1, \dots, S\} | X_{j_i} = X} m_i(X_{j_i}) \right)}, \quad (12)$$

where $A \in \mathbf{X}_j$ means that at least one component of the S -tuple $\mathbf{X}_j = (X_{j_1}, \dots, X_{j_s}) \in \mathcal{F}(m_1, \dots, m_S)$ equals A .

Finally the mass value of A obtained by the PCR5 rule is calculated by

$$m_{1,2,\dots,S}^{\text{PCR5}}(A) = m_{1,2,\dots,S}^{\text{Conj}}(A) + \sum_{j \in \{1, \dots, \mathcal{F}\} | A \in \mathbf{X}_j \wedge \pi_j(\emptyset)} x_j(A), \quad (13)$$

where $A \in \mathbf{X}_j \wedge \pi_j(\emptyset)$ is a shorthand notation meaning that at least one component of the S -tuple \mathbf{X}_j equals A and the components of \mathbf{X}_j are conflicting, i.e., $X_{j_1} \cap \dots \cap X_{j_s} = \emptyset$.

Therefore the general PCR5 formula can be expressed as $m_{1,2,\dots,S}^{\text{PCR5}}(\emptyset) = 0$, and for $A \in 2^\Theta \setminus \{\emptyset\}$ by

$$m_{1,2,\dots,S}^{\text{PCR5}}(A) = m_{1,2,\dots,S}^{\text{Conj}}(A) + \sum_{j \in \{1, \dots, \mathcal{F}\} | A \in \mathbf{X}_j \wedge \pi_j(\emptyset)} \left[\left(\prod_{i \in \{1, \dots, S\} | X_{j_i} = A} m_i(X_{j_i}) \right) \cdot \frac{\pi_j(\emptyset)}{\sum_{X \in \mathbf{X}_j} \left(\prod_{i \in \{1, \dots, S\} | X_{j_i} = X} m_i(X_{j_i}) \right)} \right]. \quad (14)$$

It is worth noting that the formula (14) is a generalization of the formula (11), i.e., (14) coincides with (11) when $S = 2$.

This general PCR5 formula is equivalent to the original PCR5 formula given in [31] but it involves only the focal elements of the BBAs to combine which makes the derivation more efficient (less computationally demanding) than the original general PCR5 formula, specially when each BBA has only few focal elements. We use this new general PCR5 formula because it is relatively simple and easy to improve it into PCR5⁺ formula—see section VI-B. The extension of PCR5 for combining qualitative⁸ BBAs can be found in [34] (Vol. 2 and 3) and in [33]. PCR5 rule is not associative and the best fusion result is obtained by combining the sources altogether at the same time when possible. A suboptimal fast fusion method using PCR5-based canonical decomposition [42] can be found in [43].

The PCR6 rule of combination [32]: A variant of PCR5 rule, called PCR6 rule, has been proposed by Martin and Osswald in [32], [33] for combining $S > 2$ sources. Because PCR6 coincides with PCR5 when one combines two sources, we do not provide the PCR6 formula for two sources which is the same as (9). The difference between PCR5 and PCR6 lies in the way the PCR is done as soon as three (or more) sources are involved in the fusion as it will be shown in the example 2 introduced in the next section. The explicit formula of the PCR6 fusion of three BBAs is given in [41] for convenience.

The PCR6 fusion of $S > 2$ BBAs is obtained by $m_{1,2,\dots,S}^{\text{PCR6}}(\emptyset) = 0$, and for all $A \in 2^\Theta \setminus \{\emptyset\}$ by⁹

$$m_{1,2,\dots,S}^{\text{PCR6}}(A) = m_{1,2,\dots,S}^{\text{Conj}}(A) + \sum_{j \in \{1, \dots, \mathcal{F}\} | A \in \mathbf{X}_j \wedge \pi_j(\emptyset)} \left[\left(\sum_{i \in \{1, \dots, S\} | X_{j_i} = A} m_i(X_{j_i}) \right) \cdot \frac{\pi_j(\emptyset)}{\sum_{X \in \mathbf{X}_j} \left(\sum_{i \in \{1, \dots, S\} | X_{j_i} = X} m_i(X_{j_i}) \right)} \right]. \quad (15)$$

The difference between the general PCR5 and PCR6 formulas is that the PCR5 proportional redistribution involves the products $\prod_{i \in \{1, \dots, S\} | X_{j_i} = A} m_i(X_{j_i})$ of multiple same focal elements A (if any) in the conflict, whereas the PCR6 conflict redistribution principle works with their sum $\sum_{i \in \{1, \dots, S\} | X_{j_i} = A} m_i(X_{j_i})$ instead. The next section presents some examples for PCR5 and PCR6 rules of combinations.

We use this general PCR6 formula instead of the original Martin-Osswald's PCR6 formula [32] because

⁸A qualitative BBA is a BBA whose values are labels (e.g., low, medium, and high) instead of real numbers.

⁹We wrote this PCR6 general formula in the style of PCR5 formula (14).

it is easier to improve it into PCR6⁺ formula—see Section VI-B. From the implementation point of view, PCR6 is simpler to implement than PCR5. From the decision-making standpoint, PCR6 is better than PCR5 when $S > 2$ as reported by Martin and Osswald in [32] (see also the Example 3 in the next section) in their applications. For convenience, some MatlabTM codes of PCR5 and PCR6 fusion rules can be found in the appendix of [44], also in Chap. 7 of [34] (Vol. 3), or from Arnaud Martin's web page [45]. PCR6 code (in R programming language) can be found also in iBelief package developed by Kuang Zhou and Arnaud Martin from the BFAS¹⁰ repository [46], or directly from [47] as well. When we have only two BBAs to combine, PCR5 and PCR6 rules provide the same result because formulas (14) and (15) coincide for $S = 2$.

In this paper, we have voluntarily chosen to present the two rules, PCR5 and PCR6, and their improved versions mainly for historical reasons and because these two rules have strong theoretical links as we have shown. By doing this, we offer the possibility to readers (and potential users) to test each of these advanced fusion methods and evaluate their performances on their own applications. Even though PCR6 is posterior to PCR5, since some researchers have implemented and are using PCR5 fusion rule, it appears important to introduce the improved version of this rule. Furthermore, PCR5 goes back exactly on the tracks of the conjunctive rule, while PCR6 does not.

IV. EXAMPLES FOR PCR5 AND PCR6 FUSION RULES

Here we provide two simple examples showing the difference of the results between PCR5 and PCR6 rules. For convenience, all numerical values given in the examples of this paper have been rounded to six decimal places when necessary.

Example 2: We consider the simplest FoD $\Theta = \{A, B\}$, and the three following BBAs

$$m_1(A) = 0.6, m_1(B) = 0.1, m_1(A \cup B) = 0.3$$

$$m_2(A) = 0.5, m_2(B) = 0.3, m_2(A \cup B) = 0.2$$

$$m_3(A) = 0.4, m_3(B) = 0.1, m_3(A \cup B) = 0.5$$

Because $\mathcal{F}_{m_1} = |\mathcal{F}(m_1)| = 3$, $\mathcal{F}_{m_2} = |\mathcal{F}(m_2)| = 3$ and $\mathcal{F}_{m_3} = |\mathcal{F}(m_3)| = 3$, we have $\mathcal{F} = \mathcal{F}_{m_1} \cdot \mathcal{F}_{m_2} \cdot \mathcal{F}_{m_3} = 27$ products to consider. Fifteen products are non-conflicting and will enter in the calculation of $m_{1,2,3}^{\text{Conj}}(A)$, $m_{1,2,3}^{\text{Conj}}(B)$, and $m_{1,2,3}^{\text{Conj}}(A \cup B)$, and 12 products are conflicting products that will need to be proportionally redistributed. The conjunctive combination of these three BBAs is

$$m_{1,2,3}^{\text{Conj}}(A) = m_1(A)m_2(A)m_3(A) \\ + m_1(A)m_2(A)m_3(A \cup B)$$

$$+ m_1(A)m_2(A \cup B)m_3(A) \\ + m_1(A \cup B)m_2(A)m_3(A) \\ + m_1(A)m_2(A \cup B)m_3(A \cup B) \\ + m_1(A \cup B)m_2(A)m_3(A \cup B) \\ + m_1(A \cup B)m_2(A \cup B)m_3(A) \\ = 0.5370,$$

$$m_{1,2,3}^{\text{Conj}}(B) = m_1(B)m_2(B)m_3(B) \\ + m_1(B)m_2(B)m_3(A \cup B) \\ + m_1(B)m_2(A \cup B)m_3(B) \\ + m_1(A \cup B)m_2(B)m_3(B) \\ + m_1(B)m_2(A \cup B)m_3(A \cup B) \\ + m_1(A \cup B)m_2(B)m_3(A \cup B) \\ + m_1(A \cup B)m_2(A \cup B)m_3(B) \\ = 0.0900,$$

$$m_{1,2,3}^{\text{Conj}}(A \cup B) = m_1(A \cup B)m_2(A \cup B)m_3(A \cup B) \\ = 0.3 \cdot 0.2 \cdot 0.5 = 0.0300,$$

and

$$m_{1,2,3}^{\text{Conj}}(\emptyset) = 1 - m_{1,2,3}^{\text{Conj}}(A) - m_{1,2,3}^{\text{Conj}}(B) - m_{1,2,3}^{\text{Conj}}(A \cup B) \\ = 0.3430,$$

In this example, we have 12 partial conflicts, noted $\pi_j(\emptyset)$ ($j = 1, \dots, 12$), which are given by the following products

$$\begin{aligned} \pi_1(\emptyset) &= m_1(A)m_2(A)m_3(B) = 0.0300, \\ \pi_2(\emptyset) &= m_1(A)m_2(B)m_3(A) = 0.0720, \\ \pi_3(\emptyset) &= m_1(B)m_2(A)m_3(A) = 0.0200, \\ \pi_4(\emptyset) &= m_1(B)m_2(B)m_3(A) = 0.0120, \\ \pi_5(\emptyset) &= m_1(B)m_2(A)m_3(B) = 0.0050, \\ \pi_6(\emptyset) &= m_1(A)m_2(B)m_3(B) = 0.0180, \\ \pi_7(\emptyset) &= m_1(A \cup B)m_2(A)m_3(B) = 0.0150, \\ \pi_8(\emptyset) &= m_1(A \cup B)m_2(B)m_3(A) = 0.0360, \\ \pi_9(\emptyset) &= m_1(B)m_2(A)m_3(A \cup B) = 0.0250, \\ \pi_{10}(\emptyset) &= m_1(A)m_2(B)m_3(A \cup B) = 0.0900, \\ \pi_{11}(\emptyset) &= m_1(A)m_2(A \cup B)m_3(B) = 0.0120, \\ \pi_{12}(\emptyset) &= m_1(B)m_2(A \cup B)m_3(A) = 0.0080. \end{aligned}$$

In applying the PCR5 formula (14), and the PCR6 formula (15), we obtain finally $m_{1,2,3}^{\text{PCR5}}(\emptyset) = m_{1,2,3}^{\text{PCR6}}(\emptyset) = 0$,

¹⁰Belief Functions and Applications Society.

and¹¹

$$\begin{aligned} m_{1,2,3}^{\text{PCR5}}(A) &\approx 0.723281, \\ m_{1,2,3}^{\text{PCR5}}(B) &\approx 0.182460, \\ m_{1,2,3}^{\text{PCR5}}(A \cup B) &\approx 0.094259, \end{aligned}$$

and

$$\begin{aligned} m_{1,2,3}^{\text{PCR6}}(A) &\approx 0.743496, \\ m_{1,2,3}^{\text{PCR6}}(B) &\approx 0.162245, \\ m_{1,2,3}^{\text{PCR6}}(A \cup B) &\approx 0.094259. \end{aligned}$$

We see a difference between the BBAs $m_{1,2,3}^{\text{PCR5}}$ and $m_{1,2,3}^{\text{PCR6}}$, which is normal because the PCR principles are quite different. Using the PCR5 fusion rule the first partial conflicting mass $\pi_1(\emptyset) = m_1(A)m_2(A)m_3(B) = 0.03$ will be redistributed back to A and B proportionally to $m_1(A)m_2(A)$ and to $m_3(B)$ as follows

$$\frac{x_1(A)}{m_1(A)m_2(A)} = \frac{x_1(B)}{m_3(B)} = \frac{\pi_1(\emptyset)}{m_1(A)m_2(A) + m_3(B)},$$

whence

$$\begin{aligned} x_1(A) &= \frac{m_1(A)m_2(A)\pi_1(\emptyset)}{m_1(A)m_2(A) + m_3(B)} = 0.0225, \\ x_1(B) &= \frac{m_3(B)\pi_1(\emptyset)}{m_1(A)m_2(A) + m_3(B)} = 0.0075. \end{aligned}$$

We can verify $\pi_1(\emptyset) = x_1(A) + x_1(B) = 0.03$.

Using the PCR6 fusion rule the first partial conflicting mass $\pi_1(\emptyset) = 0.03$ will be redistributed back to A and B proportionally to $(m_1(A) + m_2(A))$ and to $m_3(B)$. So we will get the following redistributions $x_1(A) = 0.0275$ for A and $x_1(B) = 0.0025$ for B because

$$\frac{x_1(A)}{m_1(A) + m_2(A)} = \frac{x_1(B)}{m_3(B)} = \frac{\pi_1(\emptyset)}{m_1(A) + m_2(A) + m_3(B)},$$

whence

$$\begin{aligned} x_1(A) &= \frac{(m_1(A) + m_2(A))\pi_1(\emptyset)}{m_1(A) + m_2(A) + m_3(B)} = 0.0275, \\ x_1(B) &= \frac{m_3(B)\pi_1(\emptyset)}{m_1(A) + m_2(A) + m_3(B)} = 0.0025. \end{aligned}$$

We can verify $\pi_1(\emptyset) = x_1(A) + x_1(B) = 0.03$.

Note that for all the partial conflicts having no duplicate element involved in the conflicting product $\pi_j(\emptyset)$ we make the same redistribution with PCR5 rule and with PCR6 rule. For instance, for $\pi_7(\emptyset) = m_1(A \cup B)m_2(A)m_3(B) = 0.0150$ we get

$$\begin{aligned} \frac{x_7(A \cup B)}{m_1(A \cup B)} &= \frac{x_7(A)}{m_2(A)} = \frac{x_7(B)}{m_3(B)} \\ &= \frac{\pi_7(\emptyset)}{m_1(A \cup B) + m_2(A) + m_3(B)}, \end{aligned}$$

whence $\pi_7(\emptyset) = x_7(A \cup B) + x_7(A) + x_7(B) = 0.0150$ with

$$\begin{aligned} x_7(A \cup B) &= \frac{m_1(A \cup B)\pi_7(\emptyset)}{m_1(A \cup B) + m_2(A) + m_3(B)} = 0.0050, \\ x_7(A) &= \frac{m_2(A)\pi_7(\emptyset)}{m_1(A \cup B) + m_2(A) + m_3(B)} \approx 0.0083, \\ x_7(B) &= \frac{m_3(B)\pi_7(\emptyset)}{m_1(A \cup B) + m_2(A) + m_3(B)} \approx 0.0017. \end{aligned}$$

The next example shows also the difference between PCR5 and PCR6 rules, and it justifies why PCR6 rule is usually preferred to PCR5 rule in applications.

Example 3: we consider the FoD $\Theta = \{A, B, C\}$, and the four very simple BBAs defined by

$$m_1(A \cup B) = 1, m_2(B) = 1, m_3(A \cup B) = 1, \text{ and } m_4(C) = 1.$$

These BBAs are in conflict because the intersection of their focal elements is $(A \cup B) \cap A \cap (A \cup B) \cap C = \emptyset$. In this example, one has only one product of masses to calculate, which is $\pi_1((A \cup B) \cap A \cap (A \cup B) \cap C) = m_1(A \cup B)m_2(A)m_3(A \cup B)m_4(C) = 1$. In fact this product is a conflicting product denoted $\pi_1(\emptyset)$. We can also denote it $\pi(\emptyset)$ because the index $j = 1$ is useless in this case. Moreover, these BBAs are also in total conflict because $\pi(\emptyset) = m_1(A \cup B)m_2(A)m_3(A \cup B)m_4(C) = 1$.

If one applies the PCR5 rule principle we get

$$\begin{aligned} \frac{x(A \cup B)}{m_1(A \cup B)m_3(A \cup B)} &= \frac{x(B)}{m_2(B)} = \frac{x(C)}{m_4(C)} \\ &= \frac{\pi(\emptyset)}{m_1(A \cup B)m_3(A \cup B) + m_2(B) + m_4(C)}, \end{aligned}$$

whence $x(A \cup B) = 1/3, x(B) = 1/3$ and $x(C) = 1/3$ so that

$$\begin{aligned} m_{1,2,3,4}^{\text{PCR5}}(A \cup B) &= x(A \cup B) = 1/3, \\ m_{1,2,3,4}^{\text{PCR5}}(B) &= x(B) = 1/3, \\ m_{1,2,3,4}^{\text{PCR5}}(C) &= x(C) = 1/3. \end{aligned}$$

This PCR5 result appears counter-intuitive because three sources among the four sources exclude definitely the hypothesis C because one has $Pl_1(C) = Pl_2(C) = Pl_3(C) = 0$, so it is intuitively expected that after the combination of all the four BBAs the mass committed to C should not be greater than $1/4 = 0.25$.

If one applies the PCR6 rule principle, we get

$$\begin{aligned} \frac{x(A \cup B)}{m_1(A \cup B) + m_3(A \cup B)} &= \frac{x(B)}{m_2(B)} = \frac{x(C)}{m_4(C)} \\ &= \frac{\pi(\emptyset)}{m_1(A \cup B) + m_3(A \cup B) + m_2(B) + m_4(C)}, \end{aligned}$$

whence $x(A \cup B) = 2/4, x(B) = 1/4$ and $x(C) = 1/4$ so that

¹¹The symbol \approx means "approximately equal to."

$$\begin{aligned}
m_{1,2,3,4}^{\text{PCR6}}(A \cup B) &= x(A \cup B) = 0.5, \\
m_{1,2,3,4}^{\text{PCR6}}(B) &= x(B) = 0.25, \\
m_{1,2,3,4}^{\text{PCR6}}(C) &= x(C) = 0.25,
\end{aligned}$$

which is in better agreement with what we intuitively expect because $m_{1,2,3,4}^{\text{PCR6}}(C)$ is not greater than $1/4$. Of course in this example, Dempster's rule of combination cannot be simply applied because the conflict is total yielding a division by zero in Dempster's rule formula [8], but by using eventually some discounting methods to modify the BBAs to combine.

V. FLAWED BEHAVIOR OF PCR5 AND PCR6 RULES

The PCR5 and PCR6 rules of combination are not associative which means that the fusion of the BBAs must be done using general formulas (14) or (15) if one has more than two BBAs to combine, which is not very convenient. Therefore, the sequential PCR5 or PCR6 combination of $S > 2$ BBAs are not in general equal to the global PCR5 or PCR6 fusion of the S BBAs altogether because the order of the combination of the sources does matter in the sequential combination. In general (i.e. when conflicts exist between the sources of evidence to combine) one has for $S > 2$

$$\begin{aligned}
\text{PCR5}(m_1, m_2, \dots, m_S) &\neq \\
\text{PCR5}(\text{PCR5}(\text{PCR5}(m_1, m_2), m_3), \dots, m_S) &\quad (16)
\end{aligned}$$

and

$$\begin{aligned}
\text{PCR6}(m_1, m_2, \dots, m_S) &\neq \\
\text{PCR6}(\text{PCR6}(\text{PCR6}(m_1, m_2), m_3), \dots, m_S), &\quad (17)
\end{aligned}$$

and also for $S > 2$ PCR5 fusion result is generally different of PCR6 fusion result that is

$$\text{PCR5}(m_1, m_2, \dots, m_S) \neq \text{PCR6}(m_1, m_2, \dots, m_S). \quad (18)$$

Formula (18) says that in general PCR5 is different from PCR6, of course except the case when we combine only two sources. PCR5 and PCR6 rules can become computationally intractable for combining a large number of sources and for working with large FoD. This is a well-known limitation of these rules, but this is the price to pay to get better results than with classical rules.

Aside the complexity of these rules, it is worth to mention that the neutral impact property of the VBBA m_v is lost in general when considering the PCR5 or PCR6 combination of $S > 2$ BBAs altogether, that is

$$\text{PCR5}(m_1, \dots, m_{S-1}, m_v) \neq \text{PCR5}(m_1, \dots, m_{S-1}) \quad (19)$$

and

$$\text{PCR6}(m_1, \dots, m_{S-1}, m_v) \neq \text{PCR6}(m_1, \dots, m_{S-1}) \quad (20) \quad \text{and}$$

Formula (19) and (20) show that in general PCR5 and PCR6 do not have the ignorant source as a neutral element. This is due to the redistribution principles used in PCR5 and in PCR6 rules. Example 4 shows the non-neutral impact of the VBBA in PCR5 and PCR6 rules for convenience. Note that the VBBA has a neutral impact in the fusion result if and only if one has only two BBAs to combine with PCR5, or PCR6, and one of them is the VBBA because in this case there is no possible (partial) conflict to redistribute between any BBA $m(\cdot)$ defined over the FoD Θ and the VBBA $m_v(\cdot)$. That is, for any BBA $m_1(\cdot)$ one always has

$$\text{PCR5}(m_1, m_v) = \text{PCR6}(m_1, m_v) = m_1. \quad (21)$$

Example 4: we consider the FoD $\Theta = \{A, B\}$ having only two elements, and the following four BBAs as follows:

$$\begin{aligned}
m_1(A) &= 0.6, m_1(B) = 0.1, m_1(A \cup B) = 0.3, \\
m_2(A) &= 0.5, m_2(B) = 0.3, m_2(A \cup B) = 0.2, \\
m_3(A) &= 0.4, m_3(B) = 0.1, m_3(A \cup B) = 0.5, \\
m_4(A \cup B) &= 1.
\end{aligned}$$

BBAs m_1, m_2 , and m_3 are as in example 2, and the BBA m_4 is nothing but the VBBA m_v defined over this FoD Θ .

In example 2, we did obtain with $\text{PCR5}(m_1, m_2, m_3)$ and with $\text{PCR5}(m_1, m_2, m_3, m_4)$ the following resulting BBAs

$$\begin{aligned}
m_{1,2,3}^{\text{PCR5}}(A) &\approx 0.723281, \\
m_{1,2,3}^{\text{PCR5}}(B) &\approx 0.182460, \\
m_{1,2,3}^{\text{PCR5}}(A \cup B) &\approx 0.094259,
\end{aligned}$$

and

$$\begin{aligned}
m_{1,2,3,4}^{\text{PCR5}}(A) &\approx 0.654604, \\
m_{1,2,3,4}^{\text{PCR5}}(B) &\approx 0.144825, \\
m_{1,2,3,4}^{\text{PCR5}}(A \cup B) &\approx 0.200571.
\end{aligned}$$

Clearly, $\text{PCR5}(m_1, m_2, m_3) \neq \text{PCR5}(m_1, m_2, m_3, m_4)$ even if m_4 is the VBBA.

Analogously, we did obtain with $\text{PCR6}(m_1, m_2, m_3)$ and with $\text{PCR6}(m_1, m_2, m_3, m_4)$

$$\begin{aligned}
m_{1,2,3}^{\text{PCR6}}(A) &\approx 0.743496, \\
m_{1,2,3}^{\text{PCR6}}(B) &\approx 0.162245, \\
m_{1,2,3}^{\text{PCR6}}(A \cup B) &\approx 0.094259,
\end{aligned}$$

and

$$m_{1,2,3,4}^{\text{PCR6}}(A) \approx 0.647113,$$

$$m_{1,2,3,4}^{\text{PCR6}}(B) \approx 0.128342,$$

$$m_{1,2,3,4}^{\text{PCR6}}(A \cup B) \approx 0.224545.$$

Hence, $\text{PCR6}(m_1, m_2, m_3) \neq \text{PCR6}(m_1, m_2, m_3, m_4)$, even if m_4 is the VBBA.

This example 4 shows clearly that the VBBA does not have a neutral impact in the PCR5 and PCR6 rules of combination. In fact, adding more VBBA's m_v in the PCR5 or PCR6 fusion will increase more and more the mass of $A \cup B$ while decreasing more and more the masses of A and of B with PCR5, and PCR6. When the number of VBBA's m_v increases, we will have¹² $m_{1,2,3,m_v,\dots,m_v}^{\text{PCR5/6}}(A \cup B) \rightarrow 1$, $m_{1,2,3,m_v,\dots,m_v}^{\text{PCR5/6}}(A) \rightarrow 0$, and $m_{1,2,3,m_v,\dots,m_v}^{\text{PCR5/6}}(B) \rightarrow 0$.

This is unsatisfactory because the VBBA brings no useful information to exploit, and it is naturally expected that it must not impact the fusion result in the combination of BBAs. This can be seen as a flaw of the behavior of PCR5 and PCR6 rules of combination.

To emphasize this flaw, we give in the example 5 a case where the mass committed to some partial uncertainties can increase more than necessary with PCR5 and with PCR6 rules of combination. This is detrimental for the quality of the fusion result and for decision-making because the result is more uncertain than it should be, and consequently the decision is more difficult to make.

Example 5: we consider the FoD $\Theta = \{A, B, C, D, E\}$, and the following three BBAs

$$\begin{cases} m_1(A \cup B) = 0.70 \\ m_1(C \cup D) = 0.06 \\ m_1(A \cup B \cup C \cup D) = 0.15 \\ m_1(E) = 0.09 \end{cases}$$

and

$$\begin{cases} m_2(A \cup B) = 0.06 \\ m_2(C \cup D) = 0.50 \\ m_2(A \cup B \cup C \cup D) = 0.04 \\ m_2(E) = 0.40 \end{cases}$$

and

$$\begin{cases} m_3(B) = 0.01 \\ m_3(A \cup B \cup C \cup D \cup E) = 0.99. \end{cases}$$

Note that the BBA m_3 is not equal to the VBBA but it is very close to the VBBA because $m_3(\Theta)$ is close to one.

If we make the $\text{PCR6}(m_1, m_2)$ fusion of only the two BBAs m_1 and m_2 altogether, which is also equal to $\text{PCR5}(m_1, m_2)$, we obtain

$$\begin{cases} m_{1,2}^{\text{PCR6}}(A \cup B) \approx 0.465309 \\ m_{1,2}^{\text{PCR6}}(C \cup D) \approx 0.296299 \\ m_{1,2}^{\text{PCR6}}(A \cup B \cup C \cup D) \approx 0.023471 \\ m_{1,2}^{\text{PCR6}}(E) \approx 0.214921 \end{cases}$$

If we make the $\text{PCR6}(m_1, m_2, m_3)$ fusion of all these three BBAs altogether we obtain

$$\begin{cases} m_{1,2,3}^{\text{PCR6}}(B) \approx 0.000962 \\ m_{1,2,3}^{\text{PCR6}}(A \cup B) \approx 0.286107 \\ m_{1,2,3}^{\text{PCR6}}(C \cup D) \approx 0.203454 \\ m_{1,2,3}^{\text{PCR6}}(A \cup B \cup C \cup D) \approx 0.012203 \\ m_{1,2,3}^{\text{PCR6}}(E) \approx 0.116038 \\ m_{1,2,3}^{\text{PCR6}}(A \cup B \cup C \cup D \cup E) \approx 0.381236 \end{cases}$$

One sees that combining the BBAs m_1, m_2 with the BBA m_3 (where m_3 is close to VBBA, and therefore m_3 is almost non-informative) generates a big increase of the belief of the uncertainty in the resulting BBA. This behaviour is clearly counter-intuitive because if the source is almost vacuous, only a small degradation of the uncertainty is expected and in the limit case when m_3 is the VBBA no impact of m_3 on the fusion result should occur. Note that this behavior also occurs with $\text{PCR5}(m_1, m_2, m_3)$ because one has for this example

$$\begin{cases} m_{1,2,3}^{\text{PCR5}}(B) \approx 0.001103 \\ m_{1,2,3}^{\text{PCR5}}(A \cup B) \approx 0.286107 \\ m_{1,2,3}^{\text{PCR5}}(C \cup D) \approx 0.203384 \\ m_{1,2,3}^{\text{PCR5}}(A \cup B \cup C \cup D) \approx 0.012203 \\ m_{1,2,3}^{\text{PCR5}}(E) \approx 0.115967 \\ m_{1,2,3}^{\text{PCR5}}(A \cup B \cup C \cup D \cup E) \approx 0.381236 \end{cases}$$

The deep analysis of the partial conflict redistributions done in this interesting example reveals clearly the flaw of the principles of PCR5 and PCR6 rules of combination. Indeed, for this example, one has $\mathcal{F}_{m_1} \cdot \mathcal{F}_{m_2} \cdot \mathcal{F}_{m_3} = 4 \cdot 4 \cdot 2 = 32$ products $\pi_j(X_{j_1} \cap X_{j_2} \cap X_{j_3}) = m_1(X_{j_1})m_2(X_{j_2})m_3(X_{j_3})$ to calculate, where $X_{j_1} \in \mathcal{F}(m_1) = \{A \cup B, C \cup D, A \cup B \cup C \cup D, E\}$, $X_{j_2} \in \mathcal{F}(m_2) = \{A \cup B, C \cup D, A \cup B \cup C \cup D, E\}$, and $X_{j_3} \in \mathcal{F}(m_3) = \{B, A \cup B \cup C \cup D \cup E\}$. Among these 32 possible conjunctions of focal elements, 20 products correspond to partial conflicts when $X_{j_1} \cap X_{j_2} \cap X_{j_3} = \emptyset$, which need to be redistributed properly to some elements of $2^\Theta \setminus \{\emptyset\}$ according to the PCR5, or the PCR6 redistribution principles.

More precisely, we have to consider all the following products π_j for calculating the result

$$\pi_1(B) = m_1(A \cup B)m_2(A \cup B)m_3(B) = 0.00042,$$

$$\pi_2(A \cup B) = m_1(A \cup B)m_2(A \cup B)m_3(\Theta) = 0.04158,$$

$$\pi_3(\emptyset) = m_1(A \cup B)m_2(C \cup D)m_3(B) = 0.0035,$$

$$\pi_4(\emptyset) = m_1(A \cup B)m_2(C \cup D)m_3(\Theta) = 0.3465,$$

$$\pi_5(B) = m_1(A \cup B)m_2(A \cup B \cup C \cup D)m_3(B) = 0.00028,$$

$$\pi_6(A \cup B) = m_1(A \cup B)m_2(A \cup B \cup C \cup D)m_3(\Theta) = 0.02772,$$

$$\pi_7(\emptyset) = m_1(A \cup B)m_2(E)m_3(B) = 0.0028,$$

$$\pi_8(\emptyset) = m_1(A \cup B)m_2(E)m_3(\Theta) = 0.2772,$$

$$\pi_9(\emptyset) = m_1(C \cup D)m_2(A \cup B)m_3(B) = 0.000036,$$

$$\pi_{10}(\emptyset) = m_1(C \cup D)m_2(A \cup B)m_3(\Theta) = 0.003564,$$

¹²The notation $m^{\text{PCR5/6}}$ indicates " m^{PCR5} or m^{PCR6} " for convenience.

$$\begin{aligned}
\pi_{11}(\emptyset) &= m_1(C \cup D)m_2(C \cup D)m_3(B) = 0.0003, \\
\pi_{12}(C \cup D) &= m_1(C \cup D)m_2(C \cup D)m_3(\Theta) = 0.0297, \\
\pi_{13}(\emptyset) &= m_1(C \cup D)m_2(A \cup B \cup C \cup D)m_3(B) = 0.000024, \\
\pi_{14}(C \cup D) &= m_1(C \cup D)m_2(A \cup B \cup C \cup D)m_3(\Theta) \\
&= 0.002376, \\
\pi_{15}(\emptyset) &= m_1(C \cup D)m_2(E)m_3(B) = 0.00024, \\
\pi_{16}(\emptyset) &= m_1(C \cup D)m_2(E)m_3(\Theta) = 0.02376, \\
\pi_{17}(B) &= m_1(A \cup B \cup C \cup D)m_2(A \cup B)m_3(B) = 0.00009, \\
\pi_{18}(A \cup B) &= m_1(A \cup B \cup C \cup D)m_2(A \cup B)m_3(\Theta) = 0.00891, \\
\pi_{19}(\emptyset) &= m_1(A \cup B \cup C \cup D)m_2(C \cup D)m_3(B) = 0.00075, \\
\pi_{20}(C \cup D) &= m_1(A \cup B \cup C \cup D)m_2(C \cup D)m_3(\Theta) \\
&= 0.07425, \\
\pi_{21}(B) &= m_1(A \cup B \cup C \cup D)m_2(A \cup B \cup C \cup D)m_3(B) \\
&= 0.00006, \\
\pi_{22}(A \cup B \cup C \cup D) &= m_1(A \cup B \cup C \cup D)m_2(A \cup B \cup C \cup D) \\
&\quad \cdot m_3(\Theta) = 0.00594, \\
\pi_{23}(\emptyset) &= m_1(A \cup B \cup C \cup D)m_2(E)m_3(B) = 0.0006, \\
\pi_{24}(\emptyset) &= m_1(A \cup B \cup C \cup D)m_2(E)m_3(\Theta) = 0.0594, \\
\pi_{25}(\emptyset) &= m_1(E)m_2(A \cup B)m_3(B) = 0.000054, \\
\pi_{26}(\emptyset) &= m_1(E)m_2(A \cup B)m_3(\Theta) = 0.005346, \\
\pi_{27}(\emptyset) &= m_1(E)m_2(C \cup D)m_3(B) = 0.00045, \\
\pi_{28}(\emptyset) &= m_1(E)m_2(C \cup D)m_3(\Theta) = 0.04455, \\
\pi_{29}(\emptyset) &= m_1(E)m_2(A \cup B \cup C \cup D)m_3(B) = 0.000036, \\
\pi_{30}(\emptyset) &= m_1(E)m_2(A \cup B \cup C \cup D)m_3(\Theta) = 0.003564, \\
\pi_{31}(\emptyset) &= m_1(E)m_2(E)m_3(B) = 0.00036, \\
\pi_{32}(E) &= m_1(E)m_2(E)m_3(\Theta) = 0.03564.
\end{aligned}$$

The conjunctive rule gives

$$m_{1,2,3}^{\text{Conj}}(B) = \pi_1(B) + \pi_5(B) + \pi_{17}(B) + \pi_{21}(B) = 0.00085,$$

$$\begin{aligned}
m_{1,2,3}^{\text{Conj}}(A \cup B) &= \pi_2(A \cup B) + \pi_6(A \cup B) + \pi_{18}(A \cup B) \\
&= 0.07821,
\end{aligned}$$

$$\begin{aligned}
m_{1,2,3}^{\text{Conj}}(C \cup D) &= \pi_{12}(C \cup D) + \pi_{14}(C \cup D) + \pi_{20}(C \cup D) \\
&= 0.106326,
\end{aligned}$$

$$m_{1,2,3}^{\text{Conj}}(A \cup B \cup C \cup D) = \pi_{22}(A \cup B \cup C \cup D) = 0.00594,$$

$$m_{1,2,3}^{\text{Conj}}(E) = \pi_{32}(E) = 0.03564.$$

The total conflicting mass between these three BBAs is

$$\begin{aligned}
m_{1,2,3}^{\text{Conj}}(\emptyset) &= \sum_{j=3,4,7,\dots,11,13,15,16,19,23,\dots,31} \pi_j(\emptyset) \\
&= 1 - m_{1,2,3}^{\text{Conj}}(B) - m_{1,2,3}^{\text{Conj}}(A \cup B) - m_{1,2,3}^{\text{Conj}}(C \cup D) \\
&\quad - m_{1,2,3}^{\text{Conj}}(A \cup B \cup C \cup D) - m_{1,2,3}^{\text{Conj}}(E) = 0.773034.
\end{aligned}$$

Let us examine how the $m_{1,2,3}^{\text{PCR5}}(\Theta) \approx 0.381236$ value is obtained based on the PCR5 redistribution principle. Based on the structures of $\pi_j(\emptyset)$ products, we have to consider only products involving a proportional redistribution to Θ . So we get a proportional redistribution to Θ only from the following products

$$\begin{aligned}
\pi_4(\emptyset) &= m_1(A \cup B)m_2(C \cup D)m_3(\Theta) = 0.3465, \\
\pi_8(\emptyset) &= m_1(A \cup B)m_2(E)m_3(\Theta) = 0.2772, \\
\pi_{10}(\emptyset) &= m_1(C \cup D)m_2(A \cup B)m_3(\Theta) = 0.003564, \\
\pi_{16}(\emptyset) &= m_1(C \cup D)m_2(E)m_3(\Theta) = 0.02376, \\
\pi_{24}(\emptyset) &= m_1(A \cup B \cup C \cup D)m_2(E)m_3(\Theta) = 0.0594, \\
\pi_{26}(\emptyset) &= m_1(E)m_2(A \cup B)m_3(\Theta) = 0.005346, \\
\pi_{28}(\emptyset) &= m_1(E)m_2(C \cup D)m_3(\Theta) = 0.04455, \\
\pi_{30}(\emptyset) &= m_1(E)m_2(A \cup B \cup C \cup D)m_3(\Theta) = 0.003564.
\end{aligned}$$

Because there is no duplicate focal elements in each of these products, the PCR5 and PCR6 redistributions to Θ will be the same in this example.

The proportional redistribution of $\pi_4(\emptyset)$ to Θ is

$$x_4(\Theta) = \frac{m_3(\Theta)\pi_4(\emptyset)}{m_1(A \cup B) + m_2(C \cup D) + m_3(\Theta)} \approx 0.156637.$$

The proportional redistribution of $\pi_8(\emptyset)$ to Θ is

$$x_8(\Theta) = \frac{m_3(\Theta)\pi_8(\emptyset)}{m_1(A \cup B) + m_2(E) + m_3(\Theta)} \approx 0.131305.$$

The proportional redistribution of $\pi_{10}(\emptyset)$ to Θ is

$$x_{10}(\Theta) = \frac{m_3(\Theta)\pi_{10}(\emptyset)}{m_1(C \cup D) + m_2(A \cup B) + m_3(\Theta)} \approx 0.003179.$$

The proportional redistribution of $\pi_{16}(\emptyset)$ to Θ is

$$x_{16}(\Theta) = \frac{m_3(\Theta)\pi_{16}(\emptyset)}{m_1(C \cup D) + m_2(E) + m_3(\Theta)} \approx 0.016222.$$

The proportional redistribution of $\pi_{24}(\emptyset)$ to Θ is

$$x_{24}(\Theta) = \frac{m_3(\Theta)\pi_{24}(\emptyset)}{m_1(A \cup B \cup C \cup D) + m_2(E) + m_3(\Theta)} \approx 0.038186.$$

The proportional redistribution of $\pi_{26}(\emptyset)$ to Θ is

$$x_{26}(\Theta) = \frac{m_3(\Theta)\pi_{26}(\emptyset)}{m_1(E) + m_2(A \cup B) + m_3(\Theta)} \approx 0.004643.$$

The proportional redistribution of $\pi_{28}(\emptyset)$ to Θ is

$$x_{28}(\Theta) = \frac{m_3(\Theta)\pi_{28}(\emptyset)}{m_1(E) + m_2(C \cup D) + m_3(\Theta)} \approx 0.027914.$$

The proportional redistribution of $\pi_{30}(\emptyset)$ to Θ is

$$x_{30}(\Theta) = \frac{m_3(\Theta)\pi_{30}(\emptyset)}{m_1(E) + m_2(A \cup B \cup C \cup D) + m_3(\Theta)} \approx 0.003150.$$

Therefore, we finally obtain the quite big value for the mass committed to Θ

$$\begin{aligned}
m_{1,2,3}^{\text{PCR5}}(\Theta) &= x_4(\Theta) + x_8(\Theta) + x_{10}(\Theta) + x_{16}(\Theta) + x_{24}(\Theta) \\
&\quad + x_{26}(\Theta) + x_{28}(\Theta) + x_{30}(\Theta) \\
&\approx 0.381236.
\end{aligned}$$

We see clearly why PCR5 (and PCR6) redistributes some mass to uncertainty Θ although the focal element Θ is not in conflict with other focal elements involved in each product $\pi_4(\emptyset)$, $\pi_8(\emptyset)$, $\pi_{10}(\emptyset)$, $\pi_{16}(\emptyset)$, $\pi_{24}(\emptyset)$, $\pi_{26}(\emptyset)$, $\pi_{28}(\emptyset)$, and $\pi_{30}(\emptyset)$, which is an undesirable behavior that we want to avoid. That is why we propose in the next section some improvement of PCR5 and PCR6 rules of combination.

VI. IMPROVEMENT OF PCR5 AND PCR6 RULES

To circumvent the weakness of the original PCR5 and PCR6 redistribution principles, we propose an improvement of these rules that will be denoted as PCR5⁺ and PCR6⁺ in the sequel. These new rules are not redundant with PCR5 nor with PCR6 when combining more than two BBAs altogether.

The very simple and basic idea to improve PCR5 and PCR6 redistribution principles is to discard the elements that contain all the other elements implied in the partial conflict $\pi_j(\emptyset)$ calculation. Indeed, the elements discarded are regarded as non-informative and not useful for making the conflict redistribution.

For instance, if we consider the previous example 5, the conflicting mass with PCR5⁺ and PCR6⁺ for the conflicting product $\pi_4(\emptyset) = m_1(A \cup B)m_2(C \cup D)m_3(\Theta)$ will be proportionally redistributed back only to $A \cup B$ and to $C \cup D$ but not to Θ because $A \cup B \subseteq \Theta$ and $C \cup D \subseteq \Theta$. Thus, with PCR5⁺ and PCR6⁺ rules, we will make the following redistribution:

$$\frac{x_4(A \cup B)}{m_1(A \cup B)} = \frac{x_4(C \cup D)}{m_2(C \cup D)} = \frac{\pi_4(\emptyset)}{m_1(A \cup B) + m_2(C \cup D)}$$

Here, $x_4(\Theta)$ is set to 0 with PCR5⁺ and PCR6⁺ principles because no proportion of $\pi_4(\emptyset)$ must be redistributed to Θ .

However, with PCR5 and PCR6 rule we make the redistributions according to

$$\begin{aligned}
\frac{x_4(A \cup B)}{m_1(A \cup B)} &= \frac{x_4(C \cup D)}{m_2(C \cup D)} = \frac{x_4(\Theta)}{m_3(\Theta)} \\
&= \frac{\pi_4(\emptyset)}{m_1(A \cup B) + m_2(C \cup D) + m_3(\Theta)}.
\end{aligned}$$

A. Selection of focal elements for proportional redistribution

The main issue to improve PCR5 and PCR6 rules of combination is how to identify in each conflicting product $\pi_j(\emptyset)$ the set of elements to keep for making the improved proportional redistribution.

In this section, we propose a solution of this problem that can be easily implemented. For convenience, we give also the basic MatlabTM codes of PCR5⁺ and PCR6⁺ in Appendix 3.

Let us consider $\pi_j(\emptyset) = m_1(X_{j_1})m_2(X_{j_2}) \dots m_S(X_{j_S})$ a conflicting product¹³ where $X_{j_1} \cap X_{j_2} \cap \dots \cap X_{j_S} = \emptyset$. We denote by $\mathcal{X}_j = \{X_1, \dots, X_{s_j}, s_j \leq S\}$ the set of all distinct components of the S -tuple \mathbf{X}_j related with the conflicting product $\pi_j(\emptyset)$. The order of the elements in \mathcal{X}_j does not matter. The number s_j of elements in \mathcal{X}_j can be less than S because it is possible to have duplicate focal elements in $\pi_j(\emptyset)$. We consider in \mathcal{X}_j only the distinct focal elements involved in $\pi_j(\emptyset)$ (see the next example) and we will define their binary *keeping-index indicator* which will allow to know if each element of \mathcal{X}_j needs to be kept in the PCR, or not, in the improved PCR5 and PCR6 rules of combination.

For each element $X_l \in \mathcal{X}_j$ we first define its binary *containing indicator* $\delta_j(X_{l'}, X_l)$ with respect to $X_{l'} \in \mathcal{X}_j$ to characterize if X_l contains (includes) $X_{l'}$ in wide sense, or not. Therefore, we take $\delta_j(X_{l'}, X_l) = 1$ if $X_{l'} \cap X_l = X_{l'}$, or equivalently if $X_{l'} \subseteq X_l$, and $\delta_j(X_{l'}, X_l) = 0$ otherwise. The definition of this binary *containing indicator* is summarized by the formula

$$\delta_j(X_{l'}, X_l) \triangleq \begin{cases} 1 & \text{if } X_{l'} \subseteq X_l, \\ 0 & \text{if } X_{l'} \not\subseteq X_l. \end{cases} \quad (22)$$

Of course $\delta_j(X_l, X_l) = 1$ because $X_l \cap X_l = X_l$, and we have $\delta_j(X_{l'}, X_l) = 0$ as soon as $|X_{l'}| > |X_l|$, where $|X_{l'}|$ and $|X_l|$ are the cardinalities of $X_{l'}$ and X_l , respectively. We have also $\delta_j(X_{l'}, X_l) = 0$ when $X_{l'} \cap X_l \neq X_{l'}$. For $X_l = \Theta$, we have $\delta_j(X_{l'}, X_l) = \delta_j(X_{l'}, \Theta) = 1$ for any $X_{l'} \in \mathcal{X}_j$.

To know if a focal element $X_{j_i} \in \mathbf{X}_j$ must be kept, or not, in the proportional redistribution of the j -th conflicting mass $\pi_j(\emptyset)$ with PCR5⁺ and PCR6⁺ rules, we have to determinate its binary *keeping-index* $\kappa_j(X_{j_i})$. For this, we define $\kappa_j(X_{j_i}) \in \{0, 1\}$ as follows

$$\kappa_j(X_{j_i}) \triangleq 1 - \prod_{\substack{X_{l'}, X_l \in \mathcal{X}_j \\ X_{l'} \neq X_l \\ |X_{j_i}| \leq |X_l| \\ |X_{l'}| \leq |X_l|}} \delta_j(X_{l'}, X_l) \quad (23)$$

The value $\kappa_j(X_{j_i}) = 1$ stipulates that the focal element $X_{j_i} \in \mathbf{X}_j$ must receive some proportional redistribution from the conflicting mass $\pi_j(\emptyset)$. The value $\kappa_j(X_{j_i}) = 0$ indicates that the focal element X_{j_i} will not be involved in the proportional redistribution of the conflicting mass $\pi_j(\emptyset)$.

¹³We consider $S > 2$ BBAs because for $S = 2$ BBAs, no improper increasing of uncertainty occurs with PCR5 or PCR6.

The binary keeping-index can also be defined equivalently as

$$\kappa_j(X_{j_i}) = \begin{cases} 1 & \text{if } c(X_{j_i}) \text{ is true} \\ 1 - \prod_{\substack{X_{l'} \in \mathcal{X}_j \\ X_{l'} \neq X_{j_i} \\ |X_{l'}| \leq |X_{j_i}|}} \delta_j(X_{l'}, X_{j_i}) & \text{if } c(X_{j_i}) \text{ is false,} \end{cases} \quad (24)$$

where the condition $c(X_{j_i})$ is defined as

$$c(X_{j_i}) \triangleq \exists X_l \in \mathcal{X}_j \text{ such } |X_l| > |X_{j_i}| \text{ and } \kappa_j(X_l) = 1.$$

Because this second definition of $\kappa_j(X_{j_i})$ is self-referencing, we need to calculate the binary keeping indexes iteratively starting by the element of \mathcal{X}_j of highest cardinality (say X), then for elements of \mathcal{X}_j of cardinality $|X| - 1$ (if any), then for elements of \mathcal{X}_j of cardinality $|X| - 2$ (if any), etc. From the implementation standpoint the definition (24) is more efficient than the direct definition (23).

Remark 1: We always have $\kappa_j(\Theta) = 0$ if $\Theta \in \mathcal{X}_j$ because Θ always includes all other focal elements of \mathcal{X}_j and Θ has the highest cardinality, so $\delta_j(X_{l'}, \Theta) = 1$ for all $X_{l'} \in \mathcal{X}_j$. Therefore the binary keeping index formula (23) reduces to

$$\kappa_j(\Theta) = 1 - \prod_{X_{l'} \in \mathcal{X}_j} \delta_j(X_{l'}, \Theta) = 1 - \underbrace{1 \cdot 1 \cdot \dots \cdot 1}_{|\mathcal{X}_j| \text{ terms}} = 0.$$

Remark 2: For a given FoD and a given number of BBAs to combine, it is always possible to calculate offline the values of the binary keeping indexes of focal elements of all possible combinations of focal elements involved in conflicting products $\pi_j(\emptyset) > 0$ because the binary keeping index depends only on the structure of the focal elements, and not on the numerical mass values of the focal elements. This remark is important, especially in applications where we have thousands or millions of fusion steps to make because we will not have to recalculate in each fusion step the binary keeping-indexes for each $\pi_j(\emptyset)$ even if the input BBAs values to combine change.

Remark 3: It is worth to recall that PCR5⁺ and PCR6⁺ have interest if and only if we have more than two ($S > 2$) BBAs to combine. If we have only two BBAs to combine ($S = 2$) we always get $m_{\text{PCR5}} = m_{\text{PCR5}^+} = m_{\text{PCR6}} = m_{\text{PCR6}^+}$ because in this case the PCR5, PCR5⁺, PCR6, and PCR6⁺ rules coincide.

For convenience, we illustrate the calculation of these binary keeping-indexes based on the direct calculation (23) for different examples.

Example 6: We consider the FoD $\Theta = \{A, B, C, D\}$, six BBAs, and the j -th conflicting (assumed strictly positive) product whose structure is as follows

$$\begin{aligned} \pi_j(\emptyset) &= m_1(A)m_2(B \cup C)m_3(A \cup C)m_4(B \cup C) \\ &\quad \cdot m_5(A \cup B \cup C)m_6(A \cup B \cup C \cup D). \end{aligned}$$

In this product $\pi_j(\emptyset)$, we have the duplicate focal element $B \cup C$ because it appears both in $m_2(B \cup C)$

and in $m_4(B \cup C)$. The focal elements entering in each BBA of $\pi_j(\emptyset)$ are respectively $X_{j_1} = A$, $X_{j_2} = B \cup C$, $X_{j_3} = A \cup C$, $X_{j_4} = B \cup C$, $X_{j_5} = A \cup B \cup C$, and $X_{j_6} = A \cup B \cup C \cup D = \Theta$. So we have to consider only the following set of distinct focal elements for this $\pi_j(\emptyset)$ product

$$\begin{aligned} \mathcal{X}_j &= \{X_1 = A, X_2 = B \cup C, X_3 = A \cup C, \\ &\quad X_4 = A \cup B \cup C, X_5 = A \cup B \cup C \cup D\}. \end{aligned}$$

Therefore, considering only $X_{l'} \neq X_l$ and $|X_{l'}| \leq |X_l|$ that are conditions entering in formula (23), we have the following binary containing indicator $\delta_j(X_{l'}, X_l)$ values:

$$\begin{aligned} \delta_j(X_1, X_2) &= 0 \text{ because } (X_1 = A) \not\subseteq (X_2 = B \cup C), \\ \delta_j(X_1, X_3) &= 1 \text{ because } (X_1 = A) \subseteq (X_3 = A \cup C), \\ \delta_j(X_1, X_4) &= 1 \text{ because } (X_1 = A) \subseteq (X_4 = A \cup B \cup C), \\ \delta_j(X_1, X_5) &= 1 \text{ because } (X_1 = A) \subseteq (X_5 = \Theta), \\ \delta_j(X_2, X_3) &= 0 \text{ because } (X_2 = B \cup C) \not\subseteq (X_3 = A \cup C), \\ \delta_j(X_2, X_4) &= 1 \text{ because } (X_2 = B \cup C) \subseteq (X_4 = A \cup B \cup C), \\ \delta_j(X_2, X_5) &= 1 \text{ because } (X_2 = B \cup C) \subseteq (X_5 = \Theta), \\ \delta_j(X_3, X_2) &= 0 \text{ because } (X_3 = A \cup C) \not\subseteq (X_2 = B \cup C), \\ \delta_j(X_3, X_4) &= 1 \text{ because } (X_3 = A \cup C) \subseteq (X_4 = A \cup B \cup C), \\ \delta_j(X_3, X_5) &= 1 \text{ because } (X_3 = A \cup C) \subseteq (X_5 = \Theta), \\ \delta_j(X_4, X_5) &= 1 \text{ because } (X_4 = A \cup B \cup C) \subseteq (X_5 = \Theta). \end{aligned}$$

The binary keeping indexes $\kappa_j(X_{j_i})$ for $i = 1, 2, \dots, 6$ are calculated based on the formula (23) as follows:

- For the focal element $X_{j_1} = A = X_1$ of \mathcal{X}_j having $|X_{j_1}| = 1$, we get

$$\begin{aligned} \kappa_j(A) &= 1 - \prod_{\substack{X_{l'}, X_l \in \mathcal{X}_j \\ X_{l'} \neq X_l \\ |X_{j_1}| \leq |X_l| \\ |X_{l'}| \leq |X_l|}} \delta_j(X_{l'}, X_l) \\ &= 1 - [\delta_j(X_1, X_2)\delta_j(X_1, X_3)\delta_j(X_1, X_4)\delta_j(X_1, X_5) \\ &\quad \cdot \delta_j(X_2, X_3)\delta_j(X_2, X_4)\delta_j(X_2, X_5)\delta_j(X_3, X_2) \\ &\quad \cdot \delta_j(X_3, X_4)\delta_j(X_3, X_5)\delta_j(X_4, X_5)] \\ &= 1 - 0 \cdot 1 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 1 = 1. \end{aligned}$$

Hence, the focal element $X_{j_1} = A$ will be kept in the proportional redistribution of the conflicting mass $\pi_j(\emptyset)$.

- For the focal element $X_{j_2} = B \cup C = X_2$ of \mathcal{X}_j having $|X_{j_2}| = 2$, we get

$$\begin{aligned} \kappa_j(B \cup C) &= 1 - \prod_{\substack{X_{l'}, X_l \in \mathcal{X}_j \\ X_{l'} \neq X_l \\ |X_{j_2}| \leq |X_l| \\ |X_{l'}| \leq |X_l|}} \delta_j(X_{l'}, X_l) \\ &= 1 - [\delta_j(X_1, X_2)\delta_j(X_1, X_3)\delta_j(X_1, X_4) \end{aligned}$$

$$\begin{aligned}
& \cdot \delta_j(X_1, X_5)\delta_j(X_2, X_3)\delta_j(X_2, X_4) \\
& \cdot \delta_j(X_2, X_5)\delta_j(X_3, X_2)\delta_j(X_3, X_4) \\
& \cdot \delta_j(X_3, X_5)\delta_j(X_4, X_5)] \\
& = 1 - 0 \cdot 1 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 1 = 1.
\end{aligned}$$

Hence, the focal element $X_{j_2} = B \cup C$ will be kept in the proportional redistribution of the conflicting mass $\pi_j(\emptyset)$.

- For the focal element $X_{j_3} = A \cup C = X_3$ of \mathcal{X}_j having $|X_{j_3}| = 2$, we get

$$\begin{aligned}
\kappa_j(A \cup C) &= 1 - \prod_{\substack{X_{l'}, X_l \in \mathcal{X}_j \\ X_{l'} \neq X_l \\ |X_{j_3}| \leq |X_l| \\ |X_{l'}| \leq |X_l|}} \delta_j(X_{l'}, X_l) \\
&= 1 - [\delta_j(X_1, X_2)\delta_j(X_1, X_3)\delta_j(X_1, X_4) \\
& \quad \cdot \delta_j(X_1, X_5)\delta_j(X_2, X_3)\delta_j(X_2, X_4) \\
& \quad \cdot \delta_j(X_2, X_5)\delta_j(X_3, X_2)\delta_j(X_3, X_4) \\
& \quad \cdot \delta_j(X_3, X_5)\delta_j(X_4, X_5)] \\
&= 1 - 0 \cdot 1 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 1 = 1.
\end{aligned}$$

Hence, the focal element $X_{j_3} = A \cup C$ will be kept in the proportional redistribution of the conflicting mass $\pi_j(\emptyset)$.

- For the duplicate focal element $X_{j_4} = B \cup C$ of \mathcal{X}_j having $|X_{j_4}| = 2$, we have $\kappa_j(X_{j_4}) = 1$ because $X_{j_4} = X_{j_2}$ and $\kappa_j(X_{j_2}) = 1$.
- For the focal element $X_{j_5} = A \cup B \cup C = X_4$ of \mathcal{X}_j having $|X_{j_5}| = 3$, we get

$$\begin{aligned}
\kappa_j(A \cup B \cup C) &= 1 - \prod_{\substack{X_{l'}, X_l \in \mathcal{X}_j \\ X_{l'} \neq X_l \\ |X_{j_5}| \leq |X_l| \\ |X_{l'}| \leq |X_l|}} \delta_j(X_{l'}, X_l) \\
&= 1 - [\delta_j(X_1, X_4)\delta_j(X_1, X_5) \\
& \quad \cdot \delta_j(X_2, X_4)\delta_j(X_2, X_5)\delta_j(X_3, X_4) \\
& \quad \cdot \delta_j(X_3, X_5)\delta_j(X_4, X_5)] \\
&= 1 - 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 0.
\end{aligned}$$

Hence, the focal element $X_{j_5} = A \cup B \cup C$ will be discarded in the proportional redistribution of the conflicting mass $\pi_j(\emptyset)$.

- For the focal element $X_{j_6} = A \cup B \cup C \cup D = \Theta = X_5$ of \mathcal{X}_j having $|X_{j_6}| = 4$, we get

$$\kappa_j(\Theta) = 1 - \prod_{\substack{X_{l'}, X_l \in \mathcal{X}_j \\ X_{l'} \neq X_l \\ |X_{j_6}| \leq |X_l| \\ |X_{l'}| \leq |X_l|}} \delta_j(X_{l'}, X_l)$$

$$\begin{aligned}
& = 1 - \delta_j(X_1, X_5)\delta_j(X_2, X_5)\delta_j(X_3, X_5)\delta_j(X_4, X_5) \\
& = 1 - 1 \cdot 1 \cdot 1 \cdot 1 = 0.
\end{aligned}$$

This result illustrates the validity of the aforementioned remark 1. Hence, the focal element $X_{j_5} = A \cup B \cup C \cup D = \Theta$ will be discarded in the proportional redistribution of the conflicting mass $\pi_j(\emptyset)$.

In summary, the conflicting product $\pi_j(\emptyset) = m_1(A)m_2(BUC)m_3(AUC)m_4(BUC)m_5(AUBUC)m_6(\Theta)$ will be redistributed only to the three focal elements A , $B \cup C$, and $A \cup C$ with the improved rules PCR5⁺ and PCR6⁺, whereas it would have been redistributed to all five focal elements A , $B \cup C$, $A \cup C$, $A \cup B \cup C$, and Θ with the classical PCR5 and PCR6 rules. Thus, two focal elements were discarded.

Example 7: This example is somehow an extension of example 6 by including a new element E in the FoD. So, the FoD is $\Theta = \{A, B, C, D, E\}$, seven BBAs, and the j -th conflicting (assumed strictly positive) product whose structure is as follows

$$\begin{aligned}
\pi_j(\emptyset) &= m_1(AUE)m_2(BUCUE)m_3(AUCUE)m_4(BUCUE) \\
& \quad \cdot m_5(AUBUCUE)m_6(AUBUCUDUE)m_7(A).
\end{aligned}$$

In this product $\pi_j(\emptyset)$, we have the duplicate focal element $BUCUE$ because it appears both in $m_2(BUCUE)$ and in $m_4(BUCUE)$. The focal elements entering in each BBA of $\pi_j(\emptyset)$ are, respectively, $X_{j_1} = A \cup E$, $X_{j_2} = B \cup C \cup E$, $X_{j_3} = A \cup C \cup E$, $X_{j_4} = B \cup C \cup E$, $X_{j_5} = A \cup B \cup C \cup E$, $X_{j_6} = A \cup B \cup C \cup D \cup E = \Theta$, and $X_{j_7} = A$. So we have to consider only the following set of distinct focal elements for this $\pi_j(\emptyset)$ product

$$\mathcal{X}_j = \{X_1 = A \cup E, X_2 = B \cup C \cup E, X_3 = A \cup C \cup E, X_4 = A \cup B \cup C \cup E, X_5 = A \cup B \cup C \cup D \cup E, X_6 = A\}.$$

Therefore, considering only $X_{l'} \neq X_l$ and $|X_{l'}| \leq |X_l|$ that are conditions entering in formula (23), we have the following binary containing indicator $\delta_j(X_{l'}, X_l)$ values:

$$\begin{aligned}
\delta_j(X_6, X_1) &= 1 \text{ because } (X_6 = A) \subseteq (X_1 = A \cup E), \\
\delta_j(X_6, X_2) &= 0 \text{ because } (X_6 = A) \not\subseteq (X_2 = B \cup C \cup E), \\
\delta_j(X_6, X_3) &= 1 \text{ because } (X_6 = A) \subseteq (X_3 = A \cup C \cup E), \\
\delta_j(X_6, X_4) &= 1 \text{ because } (X_6 = A) \subseteq (X_4 = A \cup B \cup C \cup E), \\
\delta_j(X_6, X_5) &= 1 \text{ because } (X_6 = A) \subseteq (X_5 = \Theta), \\
\delta_j(X_1, X_2) &= 0 \text{ because } (X_1 = A \cup E) \not\subseteq (X_2 = B \cup C \cup E), \\
\delta_j(X_1, X_3) &= 1 \text{ because } (X_1 = A \cup E) \subseteq (X_3 = A \cup C \cup E), \\
\delta_j(X_1, X_4) &= 1 \text{ because } (X_1 = A \cup E) \subseteq (X_4 = A \cup B \cup C \cup E), \\
\delta_j(X_1, X_5) &= 1 \text{ because } (X_1 = A \cup E) \subseteq (X_5 = \Theta), \\
\delta_j(X_2, X_3) &= 0 \text{ because } (X_2 = B \cup C \cup E) \not\subseteq (X_3 = A \cup C \cup E), \\
\delta_j(X_2, X_4) &= 1 \text{ because } (X_2 = B \cup C \cup E) \subseteq (X_4 = A \cup B \cup C \cup E), \\
\delta_j(X_2, X_5) &= 1 \text{ because } (X_2 = B \cup C \cup E) \subseteq (X_5 = \Theta), \\
\delta_j(X_3, X_2) &= 0 \text{ because } (X_3 = A \cup C \cup E) \not\subseteq (X_2 = B \cup C \cup E),
\end{aligned}$$

$\delta_j(X_3, X_4) = 1$ because $(X_3 = A \cup C \cup E) \subseteq (X_4 = A \cup B \cup C \cup E)$,

$\delta_j(X_3, X_5) = 1$ because $(X_3 = A \cup C \cup E) \subseteq (X_5 = \Theta)$,

$\delta_j(X_4, X_5) = 1$ because $(X_4 = A \cup B \cup C \cup E) \subseteq (X_5 = \Theta)$.

The binary keeping indexes $\kappa_j(X_{j_i})$ for $i = 1, 2, \dots, 7$ are calculated based on the formula (23) as follows

- For the focal element $X_{j_1} = A \cup E = X_1$ of \mathcal{X}_j having $|X_{j_1}| = 2$, we get

$$\begin{aligned} \kappa_j(X_{j_1}) &= 1 - \prod_{\substack{X_{l'}, X_l \in \mathcal{X}_j \\ X_{l'} \neq X_l \\ |X_{j_1}| \leq |X_l| \\ |X_{l'}| \leq |X_l|}} \delta_j(X_{l'}, X_l) \\ &= 1 - [\delta_j(X_1, X_2)\delta_j(X_1, X_3)\delta_j(X_1, X_4)\delta_j(X_1, X_5) \\ &\quad \cdot \delta_j(X_2, X_3)\delta_j(X_2, X_4)\delta_j(X_2, X_5)\delta_j(X_3, X_2) \\ &\quad \cdot \delta_j(X_3, X_4)\delta_j(X_3, X_5)\delta_j(X_4, X_5)\delta_j(X_6, X_1) \\ &\quad \cdot \delta_j(X_6, X_2)\delta_j(X_6, X_3)\delta_j(X_6, X_4)\delta_j(X_6, X_5)] \\ &= 1 - 0 \cdot 1 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 1 = 1. \end{aligned}$$

Hence, the focal element $X_{j_1} = A \cup E$ will be kept in the proportional redistribution of the conflicting mass $\pi_j(\emptyset)$.

- For the focal element $X_{j_2} = B \cup C \cup E = X_2$ of \mathcal{X}_j having $|X_{j_2}| = 3$, we get

$$\begin{aligned} \kappa_j(X_{j_2}) &= 1 - \prod_{\substack{X_{l'}, X_l \in \mathcal{X}_j \\ X_{l'} \neq X_l \\ |X_{j_2}| \leq |X_l| \\ |X_{l'}| \leq |X_l|}} \delta_j(X_{l'}, X_l) \\ &= 1 - [\delta_j(X_1, X_2)\delta_j(X_1, X_3)\delta_j(X_1, X_4)\delta_j(X_1, X_5) \\ &\quad \cdot \delta_j(X_2, X_3)\delta_j(X_2, X_4)\delta_j(X_2, X_5)\delta_j(X_3, X_2) \\ &\quad \cdot \delta_j(X_3, X_4)\delta_j(X_3, X_5)\delta_j(X_4, X_5)\delta_j(X_6, X_2) \\ &\quad \cdot \delta_j(X_6, X_3)\delta_j(X_6, X_4)\delta_j(X_6, X_5)] \\ &= 1 - 0 \cdot 1 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 1 \\ &= 1. \end{aligned}$$

Hence, the focal element $X_{j_2} = B \cup C \cup E$ will also be kept in the proportional redistribution of the conflicting mass $\pi_j(\emptyset)$.

- For the focal element $X_{j_3} = A \cup C \cup E = X_3$ of \mathcal{X}_j having $|X_{j_3}| = 3$, we get

$$\begin{aligned} \kappa_j(X_{j_3}) &= 1 - \prod_{\substack{X_{l'}, X_l \in \mathcal{X}_j \\ X_{l'} \neq X_l \\ |X_{j_3}| \leq |X_l| \\ |X_{l'}| \leq |X_l|}} \delta_j(X_{l'}, X_l) \\ &= 1 - [\delta_j(X_1, X_2)\delta_j(X_1, X_3)\delta_j(X_1, X_4)\delta_j(X_1, X_5) \\ &\quad \cdot \delta_j(X_2, X_3)\delta_j(X_2, X_4)\delta_j(X_2, X_5)\delta_j(X_3, X_2) \\ &\quad \cdot \delta_j(X_3, X_4)\delta_j(X_3, X_5)\delta_j(X_4, X_5)\delta_j(X_6, X_2) \\ &\quad \cdot \delta_j(X_6, X_3)\delta_j(X_6, X_4)\delta_j(X_6, X_5)] \\ &= 1 - 0 \cdot 1 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 1 \\ &= 1. \end{aligned}$$

Hence, the focal element $X_{j_3} = A \cup C \cup E$ is also kept in the redistribution.

- For the duplicate focal element $X_{j_4} = B \cup C \cup E$ having $|X_{j_4}| = 3$, we have $\kappa_j(X_{j_4}) = 1$ because $X_{j_4} = X_{j_2}$ and $\kappa_j(X_{j_2}) = 1$.

- For the focal element $X_{j_5} = A \cup B \cup C \cup E = X_4$ having $|X_{j_5}| = 4$, we get

$$\begin{aligned} \kappa_j(X_{j_5}) &= 1 - \prod_{\substack{X_{l'}, X_l \in \mathcal{X}_j \\ X_{l'} \neq X_l \\ |X_{j_5}| \leq |X_l| \\ |X_{l'}| \leq |X_l|}} \delta_j(X_{l'}, X_l) \\ &= 1 - [\delta_j(X_1, X_4)\delta_j(X_1, X_5)\delta_j(X_2, X_4)\delta_j(X_2, X_5) \\ &\quad \cdot \delta_j(X_3, X_4)\delta_j(X_3, X_5)\delta_j(X_4, X_5)\delta_j(X_6, X_4) \\ &\quad \cdot \delta_j(X_6, X_5)] \\ &= 1 - 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 0. \end{aligned}$$

Hence, the focal element $X_{j_5} = A \cup B \cup C \cup E$ must be ignored in the proportional redistribution.

- For the focal element $X_{j_6} = A \cup B \cup C \cup D \cup E = \Theta = X_5$ having $|X_{j_6}| = 5$, we get

$$\begin{aligned} \kappa_j(X_{j_6}) &= 1 - \prod_{\substack{X_{l'}, X_l \in \mathcal{X}_j \\ X_{l'} \neq X_l \\ |X_{j_6}| \leq |X_l| \\ |X_{l'}| \leq |X_l|}} \delta_j(X_{l'}, X_l) \\ &= 1 - [\delta_j(X_1, X_5)\delta_j(X_2, X_5)\delta_j(X_3, X_5)\delta_j(X_4, X_5) \\ &\quad \cdot \delta_j(X_6, X_5)] \\ &= 1 - 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 0. \end{aligned}$$

This result illustrates the validity of the aforementioned remark 1. Hence, the focal element $X_{j_6} = A \cup B \cup C \cup D \cup E$ must be ignored in the proportional redistribution.

- For the focal element $X_{j_7} = A = X_6$ having $|X_{j_7}| = 1$, we get naturally (see our previous remark 1)

$$\begin{aligned} \kappa_j(X_{j_7}) &= 1 - \prod_{\substack{X_{l'}, X_l \in \mathcal{X}_j \\ X_{l'} \neq X_l \\ |X_{j_7}| \leq |X_l| \\ |X_{l'}| \leq |X_l|}} \delta_j(X_{l'}, X_l) \\ &= 1 - [\delta_j(X_1, X_2)\delta_j(X_1, X_3)\delta_j(X_1, X_4)\delta_j(X_1, X_5) \\ &\quad \cdot \delta_j(X_2, X_3)\delta_j(X_2, X_4)\delta_j(X_2, X_5)\delta_j(X_3, X_2) \\ &\quad \cdot \delta_j(X_3, X_4)\delta_j(X_3, X_5)\delta_j(X_4, X_5)\delta_j(X_6, X_2) \\ &\quad \cdot \delta_j(X_6, X_3)\delta_j(X_6, X_4)\delta_j(X_6, X_5)] \\ &= 1 - 0 \cdot 1 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 1 \\ &= 1. \end{aligned}$$

Hence, the focal element $X_{j_7} = A$ must be kept in the proportional redistribution.

In summary, the conflicting product $\pi_j(\emptyset) = m_1(A \cup E)m_2(B \cup C \cup E)m_3(A \cup C \cup E)m_4(B \cup C \cup E)m_5(A \cup B \cup C \cup E)m_6(\Theta)m_7(A)$ will be redistributed only to focal

elements $A \cup E, B \cup C \cup E, A \cup C \cup E$, and A with the improved rules PCR5⁺ and PCR6⁺, whereas it would have been redistributed to all focal elements $A \cup E, B \cup C \cup E, A \cup C \cup E, A \cup B \cup C \cup E, \Theta$, and A with the classical PCR5 and PCR6 rules.

Example 8: This is a somehow simplified version of example 6. We consider the FoD $\Theta = \{A, B, C, D\}$, only five BBAs, and suppose that the j -th conflicting (assumed strictly positive) product is as follows

$$\pi_j(\emptyset) = m_1(A)m_2(B \cup C)m_3(A \cup C)m_4(B \cup C) \cdot m_5(A \cup B \cup C \cup D).$$

Based on (23), it can be verified¹⁴ that the binary keeping indexes of focal elements involved in conflicting products are

$$\begin{aligned} \kappa_j(A) &= 1, \\ \kappa_j(B \cup C) &= 1, \\ \kappa_j(A \cup C) &= 1, \\ \kappa_j(A \cup B \cup C \cup D) &= 0. \end{aligned}$$

Example 9: We consider the FoD $\Theta = \{A, B, C, D\}$, seven BBAs, and suppose that the j -th conflicting (assumed strictly positive) product is as follows

$$\pi_j(\emptyset) = m_1(A)m_2(B \cup C)m_3(A \cup C)m_4(B \cup C) \cdot m_5(A \cup B \cup C \cup D)m_6(A \cup B \cup C)m_7(A \cup B \cup C).$$

Based on (23), it can be verified that the binary keeping indexes of focal elements involved in conflicting products are

$$\begin{aligned} \kappa_j(A) &= 1, \\ \kappa_j(B \cup C) &= 1, \\ \kappa_j(A \cup C) &= 1, \\ \kappa_j(A \cup B \cup C \cup D) &= 0, \\ \kappa_j(A \cup B \cup C) &= 0. \end{aligned}$$

Example 10: We consider the FoD $\Theta = \{A, B, C\}$, three BBAs, and suppose that the j -th conflicting (assumed strictly positive) product is as follows

$$\pi_j(\emptyset) = m_1(A)m_2(B \cup C)m_3(A \cup C).$$

Based on (23), it can be verified that the binary keeping indexes of focal elements involved in conflicting products are

$$\begin{aligned} \kappa_j(A) &= 1, \\ \kappa_j(B \cup C) &= 1, \\ \kappa_j(A \cup C) &= 1. \end{aligned}$$

Example 11: We consider the FoD $\Theta = \{A, B, C\}$, four BBAs, and suppose that the j -th conflicting (assumed strictly positive) product is as follows

$$\pi_j(\emptyset) = m_1(A)m_2(B \cup C)m_3(A \cup C)m_4(A \cup B).$$

Based on (23), it can be verified that the binary keeping-indexes of focal elements involved in conflicting products are

$$\begin{aligned} \kappa_j(A) &= 1, \\ \kappa_j(B \cup C) &= 1, \\ \kappa_j(A \cup C) &= 1, \\ \kappa_j(A \cup B) &= 1. \end{aligned}$$

Example 12: We consider the FoD $\Theta = \{A, B, C\}$, three BBAs, and suppose that the j -th conflicting (assumed strictly positive) product is as follows

$$\pi_j(\emptyset) = m_1(A \cup B \cup C)m_2(A)m_3(B \cup C).$$

Based on (23), it can be verified that the binary keeping-indexes of focal elements involved in conflicting products are

$$\begin{aligned} \kappa_j(A \cup B \cup C) &= 0, \\ \kappa_j(A) &= 1, \\ \kappa_j(B \cup C) &= 1. \end{aligned}$$

Example 13: We consider the FoD $\Theta = \{A, B, C, D\}$, and the three following BBAs

$$\begin{aligned} m_1(A \cup B) &= 0.8, m_1(C \cup D) = 0.2, \\ m_2(A \cup B) &= 0.4, m_2(C \cup D) = 0.6, \\ m_3(B) &= 0.1, m_3(A \cup B \cup C \cup D) = 0.9. \end{aligned}$$

We have $\mathcal{F} = |\mathcal{F}(m_1)| \cdot |\mathcal{F}(m_2)| \cdot |\mathcal{F}(m_3)| = 2 \cdot 2 \cdot 2 = 8$ products π_j ($j = 1, \dots, \mathcal{F}$) entering in the fusion process as follows

$$\begin{aligned} \pi_1(B) &= m_1(A \cup B)m_2(A \cup B)m_3(B) = 0.032, \\ \pi_2(A \cup B) &= m_1(A \cup B)m_2(A \cup B)m_3(\Theta) = 0.288, \\ \pi_3(\emptyset) &= m_1(A \cup B)m_2(C \cup D)m_3(B) = 0.048, \\ \pi_4(\emptyset) &= m_1(A \cup B)m_2(C \cup D)m_3(\Theta) = 0.432, \\ \pi_5(\emptyset) &= m_1(C \cup D)m_2(A \cup B)m_3(B) = 0.008, \\ \pi_6(\emptyset) &= m_1(C \cup D)m_2(A \cup B)m_3(\Theta) = 0.072, \\ \pi_7(\emptyset) &= m_1(C \cup D)m_2(C \cup D)m_3(B) = 0.012, \\ \pi_8(C \cup D) &= m_1(C \cup D)m_2(C \cup D)m_3(\Theta) = 0.108. \end{aligned}$$

Based on (23), it can be verified¹⁵ that the binary keeping-indexes of focal elements involved in conflicting products $\pi_3(\emptyset)$ to $\pi_7(\emptyset)$ are

$$\kappa_3(A \cup B) = 1, \kappa_3(C \cup D) = 1, \kappa_3(B) = 1,$$

¹⁴The verification is left to the reader.

¹⁵The verification is left to the reader.

$$\begin{aligned}
\kappa_4(A \cup B) &= 1, \kappa_4(C \cup D) = 1, \kappa_4(\Theta) = 0, \\
\kappa_5(C \cup D) &= 1, \kappa_5(A \cup B) = 1, \kappa_5(B) = 1, \\
\kappa_6(C \cup D) &= 1, \kappa_6(A \cup B) = 1, \kappa_6(\Theta) = 0, \\
\kappa_7(C \cup D) &= 1, \kappa_7(B) = 1.
\end{aligned}$$

In summary, once the binary keeping-index of $\kappa_j(X_{j_i})$ of all focal elements X_{j_i} involved in a conflicting product $\pi_j(\emptyset)$ are calculated, we can apply PCR5 or PCR6 redistribution principle only with the focal elements for which $\kappa_j(X_{j_i}) = 1$. With this new improved method of proportional redistribution, PCR5⁺ and PCR6⁺ rules will never increase the mass of non-conflicting elements involved in each $\pi_j(\emptyset)$ (if any), and in doing this way, we will preserve the neutrality of the vacuous belief assignment in the PCR5⁺ and PCR6⁺ fusion rules, which is a very desirable behavior.

B. Expressions of PCR5⁺ and PCR6⁺ fusion rules

The expressions of PCR5⁺ and PCR6⁺ fusion rules are proper modifications of PCR5 and PCR6 formulas (14) and (15) taking into account the selection of focal elements on which the proportional redistribution must apply thanks to the value of their binary keeping index.

The PCR5⁺ fusion of $S > 2$ BBAs is obtained by $m_{1,2,\dots,S}^{\text{PCR5}^+}(\emptyset) = 0$, and for all $A \in 2^\Theta \setminus \{\emptyset\}$ by

$$\begin{aligned}
m_{1,2,\dots,S}^{\text{PCR5}^+}(A) &= m_{1,2,\dots,S}^{\text{Conj}}(A) \\
&+ \sum_{j \in \{1,\dots,\mathcal{F}\} | A \in \mathbf{X}_j \wedge \pi_j(\emptyset)} \left[(\kappa_j(A) \prod_{i \in \{1,\dots,S\} | X_{j_i} = A} m_i(X_{j_i})) \right. \\
&\cdot \left. \frac{\pi_j(\emptyset)}{\sum_{X \in \mathbf{X}_j} (\kappa_j(X) \prod_{i \in \{1,\dots,S\} | X_{j_i} = X} m_i(X_{j_i}))} \right]. \quad (25)
\end{aligned}$$

The PCR6⁺ fusion of $S > 2$ BBAs is obtained by $m_{1,2,\dots,S}^{\text{PCR6}^+}(\emptyset) = 0$, and for all $A \in 2^\Theta \setminus \{\emptyset\}$ by

$$\begin{aligned}
m_{1,2,\dots,S}^{\text{PCR6}^+}(A) &= m_{1,2,\dots,S}^{\text{Conj}}(A) \\
&+ \sum_{j \in \{1,\dots,\mathcal{F}\} | A \in \mathbf{X}_j \wedge \pi_j(\emptyset)} \left[(\kappa_j(A) \sum_{i \in \{1,\dots,S\} | X_{j_i} = A} m_i(X_{j_i})) \right. \\
&\cdot \left. \frac{\pi_j(\emptyset)}{\sum_{X \in \mathbf{X}_j} (\kappa_j(X) \sum_{i \in \{1,\dots,S\} | X_{j_i} = X} m_i(X_{j_i}))} \right], \quad (26)
\end{aligned}$$

where $\kappa_j(A)$ and $\kappa_j(X)$ are, respectively, the binary keeping indexes of elements A and X involved in the conflicting product $\pi_j(\emptyset)$, that are calculated by the formula (23) or (24).

Remark 4: It is worth mentioning that PCR5⁺ formula (25) is totally consistent with PCR5 formula (14) when all binary keeping-indexes are equal to one. Similarly, the PCR6⁺ formula (26) reduces to PCR6 formula (15) if all binary keeping indexes equal one.

Theorem: The VBBA m_v has a neutral impact in PCR5⁺ and PCR6⁺ rules of combination.

Proof: see Appendix 2.

C. On the complexity of PCR5⁺ and PCR6⁺ fusion rules

The complexity of PCR5 and PCR6 rules is difficult to establish precisely because the number of computations highly depends on the structure of focal elements of the BBAs to combine, but definitely it is higher than Dempster's rule of combination. What about the complexity of PCR5⁺ and PCR6⁺ fusion rules? On one hand, PCR5⁺ and PCR6⁺ seem more complex than PCR5 and PCR6 rules because one needs extra computational burden with respect to PCR5 and PCR6 rules to calculate the binary keeping indexes. But in fact, the calculation of binary keeping indexes do not depend on the mass values of focal elements but only on their structure. Hence, the binary keeping indexes can be calculated off-line once for all for many possible structures of focal elements of BBAs to combine. On the other hand, if the binary keeping index calculation is done off-line, then PCR5⁺ and PCR6⁺ become less complex than PCR5 and PCR6 rule because some elements are discarded with PCR5⁺ and PCR6⁺ making the redistribution simpler and more effective than with PCR5 and PCR6 rules. It is not possible to say for sure if globally PCR5⁺ and PCR6⁺ are more (or less) complex than PCR5 and PCR6 because it really depends on the fusion problem under consideration and the structure of focal elements of BBAs to combine. If the sources of evidence to combine generate many partial conflicts to redistribute, including many elements to discard, then PCR5⁺ and PCR6⁺ are more advantageous than PCR5 and PCR6 in terms of reduction of complexity.

VII. EXAMPLES FOR PCR5⁺ AND PCR6⁺ FUSION RULES

Here we compare the results obtained with PCR5⁺ and PCR6⁺ with respect to those drawn from PCR5 and PCR6 rules on the examples from 1 to 13 in the previous sections. Since these following examples, for PCR5⁺ and PCR6⁺ fusion rules, respectively, consider the same FoD and BBAs as those presented, they will be denoted as "revisited examples."

Example 1 (revisited): Consider $\Theta = \{A, B\}$ and two following BBAs

$$m_1(A) = 0.1 \quad m_1(B) = 0.2 \quad m_1(A \cup B) = 0.7$$

$$m_2(A) = 0.4 \quad m_2(B) = 0.3 \quad m_2(A \cup B) = 0.3$$

Because there is only two BBAs to combine, we have

$$\text{PCR5}(m_1, m_2) = \text{PCR6}(m_1, m_2),$$

$$\text{PCR5}^+(m_1, m_2) = \text{PCR6}^+(m_1, m_2).$$

We have $m_{1,2}^{\text{Conj}}(A) = 0.35$, $m_{1,2}^{\text{Conj}}(B) = 0.33$, and $m_{1,2}^{\text{Conj}}(\Theta) = 0.21$, and we have the two conflicting products $\pi_1(\emptyset) = m_1(A)m_2(B) = 0.03$ and $\pi_2(\emptyset) = m_2(A)m_1(B) = 0.08$ to redistribute.

Applying PCR5 principle for $\pi_1(\emptyset) = 0.03$ we get

$$\frac{x_1(A)}{m_1(A)} = \frac{x_1(B)}{m_2(B)} = \frac{\pi_1(\emptyset)}{m_1(A) + m_2(B)},$$

whence $x_1(A) = 0.1 \cdot \frac{0.03}{0.1+0.3} = 0.0075$ and $x_1(B) = 0.3 \cdot \frac{0.03}{0.1+0.3} = 0.0225$.

Applying PCR5 principle for $\pi_2(\emptyset) = 0.08$ we get

$$\frac{x_2(A)}{m_2(A)} = \frac{x_2(B)}{m_1(B)} = \frac{\pi_2(\emptyset)}{m_2(A) + m_1(B)},$$

whence $x_2(A) = 0.4 \cdot \frac{0.08}{0.4+0.2} \approx 0.0533$ and $x_2(B) = 0.2 \cdot \frac{0.08}{0.4+0.2} \approx 0.0267$.

Therefore we get

$$\begin{aligned} m_{1,2}^{\text{PCR5}}(A) &= m_{1,2}^{\text{PCR6}}(A) = m_{1,2}^{\text{Conj}}(A) + x_1(A) + x_2(A) \\ &= 0.35 + 0.0075 + 0.0533 = 0.4108, \\ m_{1,2}^{\text{PCR5}}(B) &= m_{1,2}^{\text{PCR6}}(B) = m_{1,2}^{\text{Conj}}(B) + x_1(B) + x_2(B) \\ &= 0.33 + 0.0225 + 0.0267 = 0.3792, \\ m_{1,2}^{\text{PCR5}}(A \cup B) &= m_{1,2}^{\text{PCR6}}(A \cup B) = m_{1,2}^{\text{Conj}}(A \cup B) = 0.21. \end{aligned}$$

If we want to apply PCR5⁺, or PCR6⁺, rule we need to compute the binary keeping indexes of each focal element entering in the conflicting products $\pi_1(\emptyset)$ and $\pi_2(\emptyset)$. In this example, for $\pi_1(\emptyset) = m_1(A)m_2(B)$, we have $\mathcal{X}_1 = \{A, B\}$, and for $\pi_2(\emptyset) = m_2(A)m_1(B)$, we have $\mathcal{X}_2 = \{A, B\}$. Applying formula (22), we get $\delta_1(A, B) = 0$ because $A \not\subseteq B$, and $\delta_1(B, A) = 0$ because $B \not\subseteq A$ (and also $\delta_2(A, B) = 0$ and $\delta_2(B, A) = 0$). Applying formula (23) we get the binary keeping indexes $\kappa_1(A) = 1$, $\kappa_1(B) = 1$, $\kappa_2(A) = 1$, and $\kappa_2(B) = 1$, indicating that the redistribution of $\pi_1(\emptyset)$ must operate on all elements of $\mathcal{X}_1 = \{A, B\}$, and the redistribution of $\pi_2(\emptyset)$ must also operate on all elements of $\mathcal{X}_2 = \{A, B\}$, so there is no element that must be discarded for making the improved redistribution in this example. Therefore PCR5⁺, or PCR6⁺ results coincide with PCR5 and PCR6 results, that is $m^{\text{PCR5}}(\cdot) = m^{\text{PCR6}}(\cdot) = m^{\text{PCR5}^+}(\cdot) = m^{\text{PCR6}^+}(\cdot)$ which is normal.

Example 2 (revisited): Consider $\Theta = \{A, B\}$ and the three following BBAs

$$\begin{aligned} m_1(A) &= 0.6, m_1(B) = 0.1, m_1(A \cup B) = 0.3, \\ m_2(A) &= 0.5, m_2(B) = 0.3, m_2(A \cup B) = 0.2, \\ m_3(A) &= 0.4, m_3(B) = 0.1, m_3(A \cup B) = 0.5. \end{aligned}$$

As shown in Section IV, for this example, one has the following 12 conflicting products to redistribute when applying PCR5 or PCR6 fusion formulas.

$$\begin{aligned} \pi_1(\emptyset) &= m_1(A)m_2(A)m_3(B) = 0.0300, \\ \pi_2(\emptyset) &= m_1(A)m_2(B)m_3(A) = 0.0720, \end{aligned}$$

$$\begin{aligned} \pi_3(\emptyset) &= m_1(B)m_2(A)m_3(A) = 0.0200, \\ \pi_4(\emptyset) &= m_1(B)m_2(B)m_3(A) = 0.0120, \\ \pi_5(\emptyset) &= m_1(B)m_2(A)m_3(B) = 0.0050, \\ \pi_6(\emptyset) &= m_1(A)m_2(B)m_3(B) = 0.0180, \\ \pi_7(\emptyset) &= m_1(A \cup B)m_2(A)m_3(B) = 0.0150, \\ \pi_8(\emptyset) &= m_1(A \cup B)m_2(B)m_3(A) = 0.0360, \\ \pi_9(\emptyset) &= m_1(B)m_2(A)m_3(A \cup B) = 0.0250, \\ \pi_{10}(\emptyset) &= m_1(A)m_2(B)m_3(A \cup B) = 0.0900, \\ \pi_{11}(\emptyset) &= m_1(A)m_2(A \cup B)m_3(B) = 0.0120, \\ \pi_{12}(\emptyset) &= m_1(B)m_2(A \cup B)m_3(A) = 0.0080. \end{aligned}$$

With PCR5 and PCR6, the products $\pi_1(\emptyset)$ to $\pi_6(\emptyset)$ are redistributed to A and B only, whereas the products $\pi_7(\emptyset)$ to $\pi_{12}(\emptyset)$ are redistributed to A, B , and $A \cup B$. Applying PCR5 formula (14) and PCR6 formula (15), we obtain $m_{1,2,3}^{\text{PCR5}}(\emptyset) = m_{1,2,3}^{\text{PCR6}}(\emptyset) = 0$ and

$$\begin{cases} m_{1,2,3}^{\text{PCR5}}(A) \approx 0.723281 \\ m_{1,2,3}^{\text{PCR5}}(B) \approx 0.182460 \\ m_{1,2,3}^{\text{PCR5}}(A \cup B) \approx 0.094259 \end{cases} \quad \text{and} \quad \begin{cases} m_{1,2,3}^{\text{PCR6}}(A) \approx 0.743496 \\ m_{1,2,3}^{\text{PCR6}}(B) \approx 0.162245 \\ m_{1,2,3}^{\text{PCR6}}(A \cup B) \approx 0.094259 \end{cases}$$

The calculation of the binary keeping indexes by the formula (23) gives in this example

$$\begin{cases} \kappa_j(A) = 1, \kappa_j(B) = 1, & \text{for } j = 1, \dots, 6 \\ \kappa_j(A) = 1, \kappa_j(B) = 1, \kappa_j(A \cup B) = 0, & \text{for } j = 7, \dots, 12. \end{cases}$$

Therefore, if we apply the PCR5⁺ and PCR6⁺ improved rules of combination, we redistribute the products $\pi_1(\emptyset)$ to $\pi_6(\emptyset)$ to A and B (as for PCR5 and PCR6 rule), but the products $\pi_7(\emptyset)$ to $\pi_{12}(\emptyset)$ will be redistributed to A, B only, and not to $A \cup B$ because $\kappa_j(A \cup B) = 0$ for $j = 7, \dots, 12$. So finally, we obtain $m_{1,2,3}^{\text{PCR5}^+}(\emptyset) = m_{1,2,3}^{\text{PCR6}^+}(\emptyset) = 0$ and

$$\begin{cases} m_{1,2,3}^{\text{PCR5}^+}(A) \approx 0.768631 \\ m_{1,2,3}^{\text{PCR5}^+}(B) \approx 0.201369 \\ m_{1,2,3}^{\text{PCR5}^+}(A \cup B) = 0.03 \end{cases} \quad \text{and} \quad \begin{cases} m_{1,2,3}^{\text{PCR6}^+}(A) \approx 0.788847 \\ m_{1,2,3}^{\text{PCR6}^+}(B) \approx 0.181153 \\ m_{1,2,3}^{\text{PCR6}^+}(A \cup B) = 0.03 \end{cases}$$

We can verify that we obtain a more precise redistribution with PCR5⁺ (respectively PCR6⁺) rule with respect to PCR5 (respectively PCR6) rule because

TABLE I
Example 5: Results of PCR5⁺ versus PCR5

Focal elements	$m_{1,2,3}^{\text{PCR5}}(\cdot)$	$m_{1,2,3}^{\text{PCR5}^+}(\cdot)$
B	0.001103	0.001107
$A \cup B$	0.286107	0.464483
$C \cup D$	0.203385	0.296186
$A \cup B \cup C \cup D$	0.012203	0.023408
E	0.115966	0.214816
$A \cup B \cup C \cup D \cup E$	0.381236	0

TABLE II
Example 5: Results of PCR6⁺ versus PCR6

Focal elements	$m_{1,2,3}^{\text{PCR6}}(\cdot)$	$m_{1,2,3}^{\text{PCR6}^+}(\cdot)$
B	0.000962	0.000967
$A \cup B$	0.286107	0.464483
$C \cup D$	0.203454	0.296255
$A \cup B \cup C \cup D$	0.012203	0.023408
E	0.116038	0.214887
$A \cup B \cup C \cup D \cup E$	0.381236	0

$m_{1,2,3}^{\text{PCR5}^+}(A \cup B) < m_{1,2,3}^{\text{PCR5}}(A \cup B)$ and also $m_{1,2,3}^{\text{PCR6}^+}(A \cup B) < m_{1,2,3}^{\text{PCR6}}(A \cup B)$.

Example 3 (revisited): we consider $\Theta = \{A, B, C\}$, and the four very simple BBAs defined by

$$m_1(A \cup B) = 1, m_2(B) = 1, m_3(A \cup B) = 1, \text{ and } m_4(C) = 1.$$

These four BBAs are in total conflict because $(A \cup B) \cap A \cap (A \cup B) \cap C = \emptyset$, and one has only one product $\pi(\emptyset) = m_1(A \cup B)m_2(A)m_3(A \cup B)m_4(C) = 1$ to consider, so $j = 1$ in this case and it can be omitted in the notations of the binary keeping indexes.

As shown previously, one has

$$\begin{cases} m_{1,2,3,4}^{\text{PCR5}}(A \cup B) = 1/3 \\ m_{1,2,3,4}^{\text{PCR5}}(B) = 1/3 \\ m_{1,2,3,4}^{\text{PCR5}}(C) = 1/3 \end{cases} \quad \text{and} \quad \begin{cases} m_{1,2,3,4}^{\text{PCR6}}(A \cup B) = 0.5 \\ m_{1,2,3,4}^{\text{PCR6}}(B) = 0.25 \\ m_{1,2,3,4}^{\text{PCR6}}(C) = 0.25 \end{cases}$$

Because all focal elements $A \cup B, A$, and C entering in $\pi(\emptyset)$ are conflicting then one has the binary keeping-indexes $\kappa(A \cup B) = 1, \kappa(A) = 1$ and $\kappa(C) = 1$, i.e., all these elements will receive a redistribution of the conflicting mass $\pi(\emptyset)$. Therefore there is no restriction for making the redistribution. Consequently, PCR5^+ result coincides with PCR5 result, and PCR6^+ result coincides with PCR6 result.

Example 4 (revisited): we consider $\Theta = \{A, B\}$, and the following four BBAs

$$\begin{aligned} m_1(A) &= 0.6, m_1(B) = 0.1, m_1(A \cup B) = 0.3, \\ m_2(A) &= 0.5, m_2(B) = 0.3, m_2(A \cup B) = 0.2, \\ m_3(A) &= 0.4, m_3(B) = 0.1, m_3(A \cup B) = 0.5, \\ m_4(A \cup B) &= 1 \quad (m_4 \text{ is the VBBA}). \end{aligned}$$

The BBAs m_1, m_2 , and m_3 are the same as in example 2, and the BBA m_4 is the VBBA. We have already shown

TABLE III
Example 6: Results of PCR5⁺ versus PCR5

Focal elements	$m_{1,2,3,4,5,6}^{\text{PCR5}}(\cdot)$	$m_{1,2,3,4,5,6}^{\text{PCR5}^+}(\cdot)$
A	1/5	1/3
$A \cup C$	1/5	1/3
$B \cup C$	1/5	1/3
$A \cup B \cup C$	1/5	0
$A \cup B \cup C \cup D$	1/5	0

TABLE IV
Example 6: Results of PCR6⁺ versus PCR6

Focal elements	$m_{1,2,3,4,5,6}^{\text{PCR6}}(\cdot)$	$m_{1,2,3,4,5,6}^{\text{PCR6}^+}(\cdot)$
A	1/6	1/4
$A \cup C$	1/6	1/4
$B \cup C$	1/3	1/2
$A \cup B \cup C$	1/6	0
$A \cup B \cup C \cup D$	1/6	0

that $\text{PCR5}(m_1, m_2, m_3) \neq \text{PCR5}(m_1, m_2, m_3, m_4)$ even if m_4 is the VBBA, and

$$\begin{cases} m_{1,2,3,4}^{\text{PCR5}}(A) \approx 0.654604 \\ m_{1,2,3,4}^{\text{PCR5}}(B) \approx 0.144825 \\ m_{1,2,3,4}^{\text{PCR5}}(A \cup B) \approx 0.200571 \end{cases}$$

Similarly, $\text{PCR6}(m_1, m_2, m_3) \neq \text{PCR6}(m_1, m_2, m_3, m_4)$, and

$$\begin{cases} m_{1,2,3,4}^{\text{PCR6}}(A) \approx 0.647113 \\ m_{1,2,3,4}^{\text{PCR6}}(B) \approx 0.128342 \\ m_{1,2,3,4}^{\text{PCR6}}(A \cup B) \approx 0.224545 \end{cases}$$

Applying the PCR5^+ formula (25) and the PCR6^+ formula (26) we will obtain $m_{1,2,3}^{\text{PCR5}^+}(\emptyset) = m_{1,2,3,4}^{\text{PCR6}^+}(\emptyset) = 0$ and

$$\begin{cases} m_{1,2,3,4}^{\text{PCR5}^+}(A) \approx 0.768631 \\ m_{1,2,3,4}^{\text{PCR5}^+}(B) \approx 0.201369 \\ m_{1,2,3,4}^{\text{PCR5}^+}(A \cup B) = 0.03 \end{cases} \quad \text{and} \quad \begin{cases} m_{1,2,3,4}^{\text{PCR6}^+}(A) \approx 0.788847 \\ m_{1,2,3,4}^{\text{PCR6}^+}(B) \approx 0.181153 \\ m_{1,2,3,4}^{\text{PCR6}^+}(A \cup B) = 0.03 \end{cases}$$

One has $\text{PCR5}^+(m_1, m_2, m_3, m_4) = \text{PCR5}^+(m_1, m_2, m_3)$ and $\text{PCR6}^+(m_1, m_2, m_3, m_4) = \text{PCR6}^+(m_1, m_2, m_3)$ because with the improved proportional redistribution of PCR5^+ and PCR6^+ rules, the VBBA has always a neutral impact in the fusion result, which is what we intuitively expect.

Example 5 (revisited): we consider $\Theta = \{A, B, C, D, E\}$, and the following three BBAs

$$\begin{cases} m_1(A \cup B) = 0.70 \\ m_1(C \cup D) = 0.06 \\ m_1(A \cup B \cup C \cup D) = 0.15 \\ m_1(E) = 0.09 \end{cases}$$

TABLE V
Example 7: Results of PCR5⁺ versus PCR5

Focal elements	$m_{1,2,3,4,5,6,7}^{\text{PCR5}}(\cdot)$	$m_{1,2,3,4,5,6,7}^{\text{PCR5}^+}(\cdot)$
A	1/6	1/4
$A \cup E$	1/6	1/4
$A \cup C \cup E$	1/6	1/4
$B \cup C \cup E$	1/6	1/4
$A \cup B \cup C \cup E$	1/6	0
$A \cup B \cup C \cup D \cup E$	1/6	0

TABLE VI
Example 7: Results of PCR6⁺ versus PCR6

Focal elements	$m_{1,2,3,4,5,6,7}^{\text{PCR6}}(\cdot)$	$m_{1,2,3,4,5,6,7}^{\text{PCR6}^+}(\cdot)$
A	1/7	1/5
$A \cup E$	1/7	1/5
$A \cup C \cup E$	1/7	1/5
$B \cup C \cup E$	2/7	2/5
$A \cup B \cup C \cup E$	1/7	0
$A \cup B \cup C \cup D \cup E$	1/7	0

and

$$\begin{cases} m_2(A \cup B) = 0.06 \\ m_2(C \cup D) = 0.50 \\ m_2(A \cup B \cup C \cup D) = 0.04 \\ m_2(E) = 0.40 \end{cases}$$

and

$$\begin{cases} m_3(B) = 0.01 \\ m_3(A \cup B \cup C \cup D \cup E) = 0.99 \end{cases}$$

Note that the BBA m_3 is not equal to the VBBA but it is very close to the VBBA because $m_3(\Theta)$ is close to one.

If we consider the fusion of only the two first BBAs m_1 and m_2 , we have $\text{PCR6}(m_1, m_2) = \text{PCR6}^+(m_1, m_2) = \text{PCR5}(m_1, m_2) = \text{PCR5}^+(m_1, m_2)$ because all these rules coincide when combining two BBAs.

$$\begin{cases} m_{1,2}^{\text{PCR6}}(A \cup B) \approx 0.465309 \\ m_{1,2}^{\text{PCR6}}(C \cup D) \approx 0.296299 \\ m_{1,2}^{\text{PCR6}}(A \cup B \cup C \cup D) \approx 0.023471 \\ m_{1,2}^{\text{PCR6}}(E) \approx 0.214921 \end{cases}$$

If we make the PCR5, PCR5⁺, PCR6, and PCR6⁺ fusion of these three BBAs altogether we obtain now different results which is normal, because for $S > 2$, one has $\text{PCR5}^+(m_1, \dots, m_S) \neq \text{PCR5}(m_1, \dots, m_S)$ and $\text{PCR6}^+(m_1, \dots, m_S) \neq \text{PCR6}(m_1, \dots, m_S)$ in general. So, in this example 5, we get results shown in Tables I and II.

These values highlight the great ignorance of the results proposed by PCR5 and PCR6 when the third (almost vacuous) source of information is taken into account. Indeed, $m_{1,2,3}^{\text{PCR5}}(\Theta) = m_{1,2,3}^{\text{PCR6}}(\Theta)$ is the greatest mass among the set of hypotheses, whereas the results proposed with PCR5⁺ and PCR6⁺ combina-

TABLE VII
Example 8: Results of PCR5⁺ versus PCR5

Focal elements	$m_{1,2,3,4,5}^{\text{PCR5}}(\cdot)$	$m_{1,2,3,4,5}^{\text{PCR5}^+}(\cdot)$
A	1/4	1/3
$A \cup C$	1/4	1/3
$B \cup C$	1/4	1/3
$A \cup B \cup C \cup D$	1/4	0

TABLE VIII
Example 8: Results of PCR6⁺ versus PCR6

Focal elements	$m_{1,2,3,4,5}^{\text{PCR6}}(\cdot)$	$m_{1,2,3,4,5}^{\text{PCR6}^+}(\cdot)$
A	1/5	1/4
$A \cup C$	1/5	1/4
$B \cup C$	2/5	1/2
$A \cup B \cup C \cup D$	1/5	0

tion rules discard the ignorant information and propose results closer to those obtained by merging two sources. Indeed, the largest mass is allocated to $A \cup B$.

The next examples 6–12 are very simple examples involving only categorical BBAs so that only one conflicting product (equals to one) needs to be redistributed based on PCR5, PCR6, PCR5⁺, and PCR6⁺ rules. These examples offer the possibility to the reader to do the derivations manually for making a verification of our results.

Example 6 (revisited): we consider $\Theta = \{A, B, C, D\}$, and the following categorical BBAs $m_1(A) = 1, m_2(B \cup C) = 1, m_3(A \cup C) = 1, m_4(B \cup C) = 1, m_5(A \cup B \cup C) = 1$, and $m_6(A \cup B \cup C \cup D) = 1$. If we make the PCR5, PCR5⁺, PCR6, and PCR6⁺ fusion of these six BBAs altogether, we obtain results given in Tables III and IV.

In this example, we have only one conflicting product $\pi_1(\emptyset)$ to redistribute which is given by

$$\begin{aligned} \pi_1(\emptyset) &= m_1(A)m_2(B \cup C)m_3(A \cup C)m_4(B \cup C) \\ &\quad \cdot m_5(A \cup B \cup C)m_6(A \cup B \cup C \cup D). \end{aligned}$$

Because $\kappa_1(A \cup B \cup C) = 0$ and $\kappa_1(A \cup B \cup C \cup D) = 0$, these two disjunctions are discarded and more mass is committed to $A, A \cup C$ and $B \cup C$ with PCR5⁺ and PCR6⁺ rules. There is more mass allocated to $B \cup C$ with PCR6⁺ and PCR6 than with PCR5⁺ and PCR5 because two sources of information support this hypothesis.

Example 7 (revisited): we consider $\Theta = \{A, B, C, D, E\}$, and the following seven categorical BBAs $m_1(A \cup E) = 1, m_2(B \cup C \cup E) = 1, m_3(A \cup C \cup E) = 1, m_4(B \cup C \cup E) = 1, m_5(A \cup B \cup C \cup E) = 1, m_6(A \cup B \cup C \cup D \cup E) = 1$, and $m_7(A) = 1$. If we make the PCR5, PCR5⁺, PCR6,

TABLE IX
Example 9: Results of PCR5⁺ versus PCR5

Focal elements	$m_{1,2,3,4,5,6,7}^{\text{PCR5}}(\cdot)$	$m_{1,2,3,4,5,6,7}^{\text{PCR5}^+}(\cdot)$
A	1/5	1/3
$A \cup C$	1/5	1/3
$B \cup C$	1/5	1/3
$A \cup B \cup C$	1/5	0
$A \cup B \cup C \cup D$	1/5	0

TABLE X
Example 9: Results of PCR6⁺ versus PCR6

Focal elements	$m_{1,2,3,4,5,6,7}^{\text{PCR6}}(\cdot)$	$m_{1,2,3,4,5,6,7}^{\text{PCR6}^+}(\cdot)$
A	1/7	1/4
$A \cup C$	1/7	1/4
$B \cup C$	2/7	1/2
$A \cup B \cup C$	2/7	0
$A \cup B \cup C \cup D$	1/7	0

and PCR6⁺ fusion of these seven BBAs altogether, we obtain results given in Tables V and VI.

In this example 7, we have only one conflicting product $\pi_1(\emptyset)$ to redistribute which is given by

$$\begin{aligned} \pi_1(\emptyset) = & m_1(A \cup E)m_2(B \cup C \cup E)m_3(A \cup C \cup E) \\ & \cdot m_4(B \cup C \cup E)m_5(A \cup B \cup C \cup E) \\ & \cdot m_6(A \cup B \cup C \cup D \cup E)m_7(A). \end{aligned}$$

Because $\kappa_1(A \cup B \cup C \cup E) = 0$ and $\kappa_1(A \cup B \cup C \cup D \cup E) = 0$, these two disjunctions are discarded and more mass is committed to $A, A \cup E, A \cup C \cup E$, and $B \cup C \cup E$ with PCR5⁺ and PCR6⁺ rules. There is more mass allocated to $B \cup C \cup E$ with PCR6⁺ and PCR6 than with PCR5⁺ and PCR5 because two sources of information support this hypothesis.

Example 8 (revisited): we consider $\Theta = \{A, B, C, D\}$, and the following categorical BBAs $m_1(A) = 1, m_2(B \cup C) = 1, m_3(A \cup C) = 1, m_4(B \cup C) = 1$, and $m_5(A \cup B \cup C \cup D) = 1$. If we make the PCR5, PCR5⁺, PCR6, and PCR6⁺ fusion of these seven BBAs altogether we obtain results given in Tables VII and VIII.

Because $\kappa_1(A \cup B \cup C \cup D) = 0$, this disjunction is discarded and more mass is committed to $A, A \cup C$, and $B \cup C$ with PCR5⁺ and PCR6⁺ rules. There is more mass allocated to $B \cup C$ with PCR6⁺ and PCR6 than with PCR5⁺ and PCR5 because two sources of information support this hypothesis.

Example 9 (revisited): we consider $\Theta = \{A, B, C, D\}$, and the following seven categorical BBAs $m_1(A) = 1, m_2(B \cup C) = 1, m_3(A \cup C) = 1, m_4(B \cup C) = 1, m_5(A \cup B \cup C \cup D) = 1, m_6(A \cup B \cup C) = 1$, and $m_7(A \cup B \cup C) = 1$. If we make the PCR5, PCR5⁺, PCR6, and PCR6⁺ fusion of these seven BBAs altogether, we obtain results given in Tables IX and X.

Because $\kappa_1(A \cup B \cup C \cup D) = 0$ and $\kappa_1(A \cup B \cup C) = 0$, these disjunctions are discarded and more mass is com-

TABLE XI
Example 10: Results of PCR5, PCR5⁺, PCR6, PCR6⁺

Focal elements	$m_{1,2,3}^{\text{PCR5}}(\cdot)$	$m_{1,2,3}^{\text{PCR5}^+}(\cdot)$	$m_{1,2,3}^{\text{PCR6}}(\cdot)$	$m_{1,2,3}^{\text{PCR6}^+}(\cdot)$
A	1/3	1/3	1/3	1/3
$A \cup C$	1/3	1/3	1/3	1/3
$B \cup C$	1/3	1/3	1/3	1/3

TABLE XII
Example 12: Results of PCR5, PCR5⁺

Focal elements	$m_{1,2,3}^{\text{PCR5}}(\cdot)$	$m_{1,2,3}^{\text{PCR5}^+}(\cdot)$
A	1/3	1/2
$B \cup C$	1/3	1/2
$A \cup B \cup C$	1/3	0

mitted to $A, A \cup C$ and $B \cup C$ with PCR5⁺ and PCR6⁺ rules. There is more mass allocated to $B \cup C$ with PCR6⁺ and PCR6 than with PCR5⁺ and PCR5 because two sources of information support this hypothesis. Similarly, more mass is allocated to $(A \cup B \cup C)$ with PCR6 than PCR5 since two sources of information support this hypothesis.

Example 10 (revisited): we consider $\Theta = \{A, B, C\}$, and the following three categorical BBAs $m_1(A) = 1, m_2(B \cup C) = 1$, and $m_3(A \cup C) = 1$. We have only one conflicting product $\pi_1(\emptyset) = m_1(A)m_2(B \cup C)m_3(A \cup C) = 1$ to redistribute, and for this example, we have $\kappa_1(A) = 1, \kappa_1(A \cup C) = 1$, and $\kappa_1(B \cup C) = 1$, which means that all focal elements $A, A \cup C$, and $B \cup C$ must be kept, and they must receive a mass through the proportional redistribution principle. Hence, in this example, we have $m_{1,2,3}^{\text{PCR5}} = m_{1,2,3}^{\text{PCR6}} = m_{1,2,3}^{\text{PCR5}^+} = m_{1,2,3}^{\text{PCR6}^+}$, and the combined masses are evenly distributed as shown in the Table XI.

Example 11 (revisited): we consider $\Theta = \{A, B, C\}$, and the following four categorical BBAs $m_1(A) = 1, m_2(B \cup C) = 1, m_3(A \cup C) = 1$, and $m_4(A \cup B) = 1$. Because we have only one conflicting product $\pi_1(\emptyset) = m_1(A)m_2(B \cup C)m_3(A \cup C)m_4(A \cup B) = 1$ and $\kappa_1(A) = 1, \kappa_1(A \cup B) = 1, \kappa_1(A \cup C) = 1$ and $\kappa_1(B \cup C) = 1$, no hypothesis is discarded in the PCR, and we get $m_{1,2,3,4}^{\text{PCR5}} = m_{1,2,3,4}^{\text{PCR6}} = m_{1,2,3,4}^{\text{PCR5}^+} = m_{1,2,3,4}^{\text{PCR6}^+}$ with the merged masses being evenly distributed, that is $m_{1,2,3,4}^{\text{PCR5}}(A) = 1/4, m_{1,2,3,4}^{\text{PCR5}}(A \cup B) = 1/4, m_{1,2,3,4}^{\text{PCR5}}(A \cup C) = 1/4$, and $m_{1,2,3,4}^{\text{PCR5}}(B \cup C) = 1/4$.

Example 12 (revisited): we consider $\Theta = \{A, B, C\}$, and the following three categorical BBAs, $m_1(A \cup B \cup C) = 1, m_2(A) = 1$, and $m_3(B \cup C) = 1$. If we make the PCR5 fusion and the PCR5⁺ fusion of these three BBAs altogether, we obtain results given in Table XII. Because $\pi_1(\emptyset) = m_1(A \cup B \cup C)m_2(A)m_3(B \cup C)$, we

TABLE XIII
Example 13: Results of PCR5⁺ versus PCR5

Focal elements	$m_{1,2,3}^{\text{PCR5}}(\cdot)$	$m_{1,2,3}^{\text{PCR5}^+}(\cdot)$
B	0.041797	0.041797
$A \cup B$	0.487632	0.613029
$C \cup D$	0.258327	0.345174
$A \cup B \cup C \cup D$	0.212244	0

TABLE XIV
Example 13: Results of PCR6⁺ versus PCR6

Focal elements	$m_{1,2,3}^{\text{PCR6}}(\cdot)$	$m_{1,2,3}^{\text{PCR6}^+}(\cdot)$
B	0.037676	0.037676
$A \cup B$	0.487632	0.613029
$C \cup D$	0.262448	0.349295
$A \cup B \cup C \cup D$	0.212244	0

get $\kappa_1(A \cup B \cup C) = 0$, $\kappa_1(A) = 1$, and $\kappa_1(B \cup C) = 1$ based on (23). Therefore, using the PCR5⁺ combination rule, we get a redistribution of the conflicting mass $\pi_1(\emptyset) = 1$ only between A and $B \cup C$. In this example we have $m_{1,2,3}^{\text{PCR5}} = m_{1,2,3}^{\text{PCR6}}$, and $m_{1,2,3}^{\text{PCR5}^+} = m_{1,2,3,4}^{\text{PCR6}^+}$ because no mass is allocated on the same hypothesis by two different sources.

Example 13 (revisited): we consider $\Theta = \{A, B, C, D\}$, and the three following BBAs

$$\begin{aligned} m_1(A \cup B) &= 0.8, m_1(C \cup D) = 0.2, \\ m_2(A \cup B) &= 0.4, m_2(C \cup D) = 0.6, \\ m_3(B) &= 0.1, m_3(A \cup B \cup C \cup D) = 0.9. \end{aligned}$$

If we make the PCR5, PCR5⁺, PCR6, and PCR6⁺ fusion of these seven BBAs altogether, we obtain results given in Tables XIII and XIV.

Because $\kappa_j(\Theta) = 0$ for any conflicting product $\pi_j(\emptyset)$ involving Θ , this hypothesis is discarded in the redistribution of $\pi_4(\emptyset)$ and of $\pi_6(\emptyset)$ (see example 13 in Subsection VI-A for details), and therefore more mass is redistributed to $A \cup B$ and $C \cup D$ with PCR5⁺ and PCR6⁺ rules. No more mass is committed to B with PCR5⁺ and PCR6⁺, respectively, in comparison with PCR5 and PCR6. This is because B is not implied in any partial conflict with Θ (cf. Subsection VI-A for details).

VIII. CONCLUSION

In this paper, after having demonstrated the flawed behavior of PCR5 and PCR6 rules of combination for $S > 2$ BBAs (including possibly VBBAs), we proposed improvements to correct these behaviors. A computation of a binary keeping index has been detailed, which makes it possible to discard ignorant information sources for the calculation of each partial conflict. This binary keeping index has been integrated into the original formulations of PCR5 and PCR6 in order to ensure the neutrality property of the VBBA and to propose two new combination rules for a number of sources greater than 2: PCR5⁺ and PCR6⁺ rules. The interest of such combination rules could prove to be particularly important in an application case identifying many ignorant sources of information. In such a scenario, the prepon-

derant ignorance of a certain number of sources will no longer obscure a more precise characterization provided by other sources.

These new rules of combination have been already applied to risk analysis issues for geophysical and geotechnical data fusion in order to reinforce the levee protection characterizations [48].

APPENDIX 1: PROOF OF THE LEMMA 1

We prove that: $m_{1,2,\dots,S,S+1}^{\text{Conj}}(A) = m_{1,2,\dots,S}^{\text{Conj}}(A)$, for any $A \in 2^\Theta \setminus \{\emptyset\}$, where $m_{S+1}(\Theta) = 1$ is the VBBA m_v . The set of focal elements of $m_{S+1}(\cdot)$ is $\mathcal{F}(m_{S+1}) = \{\Theta\}$, therefore $\mathcal{F}_{m_{S+1}} = 1$ and $X_{j_{S+1}} = \Theta$. Based on the formula (6) written for $S + 1$ BBAs, we have

$$\begin{aligned} m_{1,2,\dots,S,S+1}^{\text{Conj}}(A) &= \sum_{\substack{\mathbf{X}_j \in \mathcal{F}(m_1, \dots, m_S, m_{S+1}) \\ X_{j_1} \cap \dots \cap X_{j_S} \cap X_{j_{S+1}} = A}} \pi_j(X_{j_1} \cap \dots \cap X_{j_S} \cap X_{j_{S+1}}) \\ &= \sum_{\substack{\mathbf{X}_j \in \mathcal{F}(m_1, \dots, m_S, m_{S+1}) \\ X_{j_1} \cap \dots \cap X_{j_S} \cap \Theta = A}} \prod_{i=1}^{S+1} m_i(X_{j_i}). \end{aligned} \quad (27)$$

Because $X_{j_{S+1}} = \Theta$ is constant and $m_{S+1}(X_{j_{S+1}}) = m_{S+1}(\Theta) = 1$, one has

$$\prod_{i=1}^{S+1} m_i(X_{j_i}) = \left(\prod_{i=1}^S m_i(X_{j_i}) \right) \cdot m_{S+1}(\Theta) = \prod_{i=1}^S m_i(X_{j_i}),$$

and $X_{j_1} \cap \dots \cap X_{j_S} \cap X_{j_{S+1}} = X_{j_1} \cap \dots \cap X_{j_S} \cap \Theta = X_{j_1} \cap \dots \cap X_{j_S}$. Therefore the formula (27) becomes

$$\begin{aligned} m_{1,2,\dots,S,S+1}^{\text{Conj}}(A) &= \sum_{\substack{\mathbf{X}_j \in \mathcal{F}(m_1, \dots, m_S, m_{S+1}) \\ X_{j_1} \cap \dots \cap X_{j_S} \cap \Theta = A}} \prod_{i=1}^{S+1} m_i(X_{j_i}) \\ &= \sum_{\substack{\mathbf{X}_j \in \mathcal{F}(m_1, \dots, m_S) \\ X_{j_1} \cap \dots \cap X_{j_S} = A}} \prod_{i=1}^S m_i(X_{j_i}) \\ &= m_{1,2,\dots,S}^{\text{Conj}}(A), \end{aligned}$$

which completes the proof of the Lemma 1.

APPENDIX 2: PROOF OF THE THEOREM

We prove that $\text{PCR5}^+(m_1, \dots, m_S, m_{S+1}) = \text{PCR5}^+(m_1, \dots, m_S)$, or equivalently that $m_{1,2,\dots,S,S+1}^{\text{PCR5}^+}(A) = m_{1,2,\dots,S}^{\text{PCR5}^+}(A)$ for any $A \in 2^\Theta \setminus \{\emptyset\}$, where $m_{S+1}(X_{j_{S+1}}) = m_{S+1}(\Theta) = 1$ is the VBBA. It is worth noting that $m_{1,2,\dots,S,S+1}^{\text{Conj}}(A) = m_{1,2,\dots,S}^{\text{Conj}}(A)$ for any $A \in 2^\Theta \setminus \{\emptyset\}$ because the VBBA $m_{S+1}(\cdot)$ is the neutral element of the conjunctive rule (see Lemma 1). It is important to note that when considering $A = \Theta$,

we have always $m_{1,2,\dots,S+1}^{\text{PCR5}^+}(\Theta) = m_{1,2,\dots,S,S+1}^{\text{Conj}}(\Theta) = m_{1,2,\dots,S}^{\text{Conj}}(\Theta) = m_{1,2,\dots,S}^{\text{PCR5}^+}(\Theta)$ because the binary keeping index of Θ is always equal to zero (see remark 1), i.e., $\kappa_j(\Theta) = 0$. Therefore all the redistribution terms to Θ in PCR5⁺ (and in PCR6⁺) formula are equal to zero when $A = \Theta$. So, we just have to consider $A \neq \Theta$ to make the proof.

Because $m_{S+1}(\cdot)$ is the VBBA, its set of focal elements is $\mathcal{F}(m_{S+1}) = \{\Theta\}$ and it contains only one focal element, i.e. $|\mathcal{F}(m_{S+1})| = 1$. Therefore

$$\mathcal{F} = |\mathcal{F}(m_1)| \cdot |\mathcal{F}(m_2)| \cdot \dots \cdot |\mathcal{F}(m_S)| \cdot |\mathcal{F}(m_{S+1})| \quad (28)$$

$$= |\mathcal{F}(m_1)| \cdot |\mathcal{F}(m_2)| \cdot \dots \cdot |\mathcal{F}(m_S)|. \quad (29)$$

This means that the number of conflicting products $\pi_j(\emptyset)$ associated to the $S + 1$ -tuple $\mathbf{X}_j = (X_{j_1}, \dots, X_{j_s}, \Theta) \in \mathcal{F}(m_1, \dots, m_S, m_{S+1})$ is equal to the number of conflicting products $\pi_j(\emptyset)$ associated to S -tuple $\mathbf{X}_j = (X_{j_1}, \dots, X_{j_s}) \in \mathcal{F}(m_1, \dots, m_S)$. Moreover, we always have

$$\prod_{i=1}^{S+1} m_i(X_{j_i}) = \left(\prod_{i=1}^S m_i(X_{j_i}) \right) \cdot m_{S+1}(\Theta) = \prod_{i=1}^S m_i(X_{j_i}).$$

Hence, we always have

$$\pi_j(X_{j_1} \cap \dots \cap X_{j_s} \cap \Theta = \emptyset) = \pi_j(X_{j_1} \cap \dots \cap X_{j_s} = \emptyset),$$

because $X_{j_1} \cap \dots \cap X_{j_s} \cap \Theta = X_{j_1} \cap \dots \cap X_{j_s}$.

Based on the formula (25) written for $S + 1$ BBAs, we have

$$\begin{aligned} m_{1,2,\dots,S,S+1}^{\text{PCR5}^+}(A) &= m_{1,2,\dots,S,S+1}^{\text{Conj}}(A) \\ &+ \sum_{j \in \{1,\dots,\mathcal{F}\} | A \in \mathbf{X}_j \wedge \pi_j(\emptyset)} \left[\left(\kappa_j(A) \prod_{i \in \{1,\dots,S+1\} | X_{j_i}=A} m_i(X_{j_i}) \right) \cdot \frac{\pi_j(X_{j_1} \cap \dots \cap X_{j_s} \cap \Theta = \emptyset)}{\sum_{X \in \mathbf{X}_j} \left(\kappa_j(X) \prod_{i \in \{1,\dots,S+1\} | X_{j_i}=X} m_i(X_{j_i}) \right)} \right], \quad (30) \end{aligned}$$

where \mathcal{F} is given by (28).

Because $X_{j_{S+1}} = \Theta$ and because we consider $A \neq \Theta$, we have always

$$\prod_{i \in \{1,\dots,S+1\} | X_{j_i}=A} m_i(X_{j_i}) = \prod_{i \in \{1,\dots,S\} | X_{j_i}=A} m_i(X_{j_i}).$$

Whether $X \in \mathbf{X}_j = (X_{j_1}, \dots, X_{j_s})$ or $X \in \mathbf{X}_j = (X_{j_1}, \dots, X_{j_s}, \Theta)$ the value of $\kappa_j(X)$ is the same since the additional binary *containing indicator* $\delta_j(X, \Theta)$ entering in the product of the computation of the binary *keeping-index* is always equal to 1 and does not modify $\kappa_j(X)$ value, and of course when $X = A$. Because the binary keeping-index entering in the numerator and denominator of formula (30) removes the factor $m_{S+1}(\Theta)$ from all products it belongs to (since Θ includes all elements of the product it belongs to), the formula (30)

reduces to the following formula

$$\begin{aligned} m_{1,2,\dots,S,S+1}^{\text{PCR5}^+}(A) &= m_{1,2,\dots,S}^{\text{Conj}}(A) \\ &+ \sum_{j \in \{1,\dots,\mathcal{F}\} | A \in \mathbf{X}_j \wedge \pi_j(\emptyset)} \left[\left(\kappa_j(A) \prod_{i \in \{1,\dots,S\} | X_{j_i}=A} m_i(X_{j_i}) \right) \cdot \frac{\pi_j(X_{j_1} \cap \dots \cap X_{j_s} = \emptyset)}{\sum_{X \in \mathbf{X}_j} \left(\kappa_j(X) \prod_{i \in \{1,\dots,S\} | X_{j_i}=X} m_i(X_{j_i}) \right)} \right] \\ &= m_{1,2,\dots,S}^{\text{PCR5}^+}(A), \quad (31) \end{aligned}$$

where \mathbf{X}_j represents now the S -tuple $(X_{j_1}, \dots, X_{j_s})$, and $\pi_j(\emptyset) = \pi_j(X_{j_1} \cap \dots \cap X_{j_s} = \emptyset)$.

So, we have proved PCR5⁺(m_1, \dots, m_S, m_{S+1}) = PCR5⁺(m_1, \dots, m_S) when m_{S+1} is the VBBA. Similarly, we can prove that PCR6⁺(m_1, \dots, m_S, m_{S+1}) = PCR6⁺(m_1, \dots, m_S) when m_{S+1} is the VBBA. This completes the proof of the theorem.

APPENDIX 3: CODES OF PCR5⁺ AND PCR6⁺ RULES

For convenience, we provide two basic MatlabTM codes for PCR5⁺ and PCR6⁺ for the fusion of $S \geq 2$ BBAs for working with 2^Θ , i.e. working with Shafer's model. No input verification of input is done in the routines. It is assumed that the input matrix BBA is correct, both in dimension and in content. The derivation of all possible combinations is done with `combvec`(Combinations, `vec`) instruction which is included in the MatlabTM neural networks toolbox. This `combvec` call can be a very time-consuming task when the size of the problem increases. A standalone version of these codes is also available upon request to the authors. The j -th column of the BBA input matrix corresponds to the (vertical) BBA vector $m_j(\cdot)$ associated with the j -th source s_j . Each element of a BBA matrix is in $[0,1]$ and the sum of each column must be one. If N is the cardinality of the frame Θ and if S is the number of sources, then the size of the BBA input matrix is $((2^N) - 1) \times S$. Each column of the BBA matrix must use the classical binary encoding of elements. For example, if $\Theta = \{A, B, C\}$, then we encode the elements of $2^\Theta \setminus \{\emptyset\}$ by the binary sequence $001 \equiv A$, $010 \equiv B$, $011 \equiv A \cup B$, ..., $111 \equiv A \cup B \cup C$. The mass of empty set is not included in the BBA vector because it is always set to zero. These codes can be used and shared for free for research purposes only. Commercial uses of these codes, or adaptation of them in any programming language, is not allowed without written agreement of the authors. These codes are provided by the copyright holders "as is" and any express or implied warranties are disclaimed. The copyright holder will not be liable for any direct, or indirect damages of the use of these codes. The authors would appreciate any feedback in the use of these codes, and publication using these codes should cite this paper in agreement for their use.

```

%=====
function [mPCR5plus]=PCR5plusfusion(BBA)
%=====
% Authors and copyrights: Theo Dezert & Jean Dezert
% Input: BBA=[m1 m2 ... mS]= Matrix of BBAs to combine with PCR5+
% Output: mPCR5plus is PCR5+(m1,m2,...,mS) fusion result
%=====
NbrSources=size(BBA,2);CardTheta=log2(size(BBA,1)+1);
if(NbrSources==1), mPCR5plus=BBA(:,1);return, end
mPCR5plus=zeros(size(BBA,1),1);FocalElem = cell(NbrSources,1);
for i=1:NbrSources, FocalElem(i)=find(BBA(:,i)> 0)';end
Combinations=combvec(FocalElem(1:NbrSources)');
for c=1:size(Combinations,1)
    PC=Combinations(c,:);masseConj=diag(BBA(PC,:))';
    masseConj=prod(diag(BBA(PC,:))',2);Intersections=PC(1);
    for s=2:NbrSources, Intersections=bitand(Intersections,PC(s)); end
    if(Intersections~=0)
        mPCR5plus(Intersections)=mPCR5plus(Intersections)+masseConj;
    else
        Binary=[];CardPC=[];KeepIndex=[];
        for i=1:NbrSources
            Binary(i,:)=bitget(PC(i),CardTheta:-1:1,'int8');
            CardPC(i,:)=sum(Binary(i,:)=1);
        end
        for j=1:NbrSources
            delta=[];
            for js=1:NbrSources
                if CardPC(js)>=CardPC(j)
                    for jp=1:NbrSources
                        if PC(jp)~=PC(js) && CardPC(jp)<=CardPC(js)
                            if sum(Binary(jp,:)<=Binary(js,:))~=CardTheta
                                delta=[delta 1];
                            else
                                delta=[delta 0];
                            end
                        end
                    end
                end
            end
            end
            if isempty(delta)==1
                KeepIndex(j,1)=1;
            else
                KeepIndex(j,1)=1-prod(delta);
            end
        end
        KeepIndex=KeepIndex';
        for i=1:NbrSources
            if KeepIndex(i)=1, KeepIndex(i)=masseConj(i); end
        end
        UQ=unique(PC);Proportions=0*UQ;DenPCR5=0;
        for u=1:size(UQ,2)
            SamePropositions=find(PC==UQ(u));
            MassProd=prod(KeepIndex(SamePropositions));
            Proportions(u)= MassProd*masseConj;DenPCR5=DenPCR5+MassProd;
        end
        Proportions=Proportions/DenPCR5;
        for u=1:size(UQ,2),mPCR5plus(UQ(u))=mPCR5plus(UQ(u))+Proportions(u); end
    end
end

%=====
function [mPCR6plus]=PCR6plusfusion(BBA)
%=====
% Authors and copyrights: Theo Dezert & Jean Dezert
% Input: BBA=[m1 m2 ... mS]= Matrix of BBAs to combine with PCR6+
% Output: mPCR6plus is PCR6+(m1,m2,...,mS) fusion result
%=====
NbrSources=size(BBA,2);CardTheta=log2(size(BBA,1)+1);
if(NbrSources==1), mPCR6plus=BBA(:,1);return, end
mPCR6plus=zeros(size(BBA,1),1);FocalElem = cell(NbrSources,1);
for i=1:NbrSources, FocalElem(i)=find(BBA(:,i)> 0)';end
Combinations=combvec(FocalElem(1:NbrSources)');
for c=1:size(Combinations,1)
    PC=Combinations(c,:);masseConj=diag(BBA(PC,:))';
    masseConj=prod(diag(BBA(PC,:))',2);Intersections=PC(1);
    for s=2:NbrSources, Intersections=bitand(Intersections,PC(s));end
    if(Intersections~=0)
        mPCR6plus(Intersections)=mPCR6plus(Intersections)+masseConj;
    else
        Binary=[];CardPC=[];KeepIndex=[];
        for i=1:NbrSources
            Binary(i,:)=bitget(PC(i),CardTheta:-1:1,'int8');
            CardPC(i,:)=sum(Binary(i,:)=1);
        end
        for j=1:NbrSources
            delta=[];
            for js=1:NbrSources
                if CardPC(js)>=CardPC(j)
                    for jp=1:NbrSources
                        if PC(jp)~=PC(js) && CardPC(jp)<=CardPC(js)
                            if sum(Binary(jp,:)<=Binary(js,:))~=CardTheta
                                delta=[delta 1];
                            else
                                delta=[delta 0];
                            end
                        end
                    end
                end
            end
            end
            if isempty(delta)==1
                KeepIndex(j,1)=1;
            else
                KeepIndex(j,1)=1-prod(delta);
            end
        end
        KeepIndex=KeepIndex';IgnoringSetOfFE=find(KeepIndex==0);
        masseConj(IgnoringSetOfFE)=[];PC(IgnoringSetOfFE)=[];
        for s=1:numel(masseConj)
            Proportion= masseConj(s)*(masseConj/(sum(masseConj,2)));
            mPCR6plus(PC(s))=mPCR6plus(PC(s))+Proportion;
        end
    end
end
end

```

REFERENCES

- [1] J.Y. Halpern
Reasoning About Uncertainty. Cambridge, MA: The MIT Press, 2003.
- [2] A. N. Kolmogorov
Foundations of the Theory of Probability. New York, NY: Chelsea Publishing Company, 1956.
- [3] A. N. Kolmogorov
“Three approaches to the quantitative definition of information,”
Int. J. Comput. Math., vol. 2, no. 1–4, pp. 157–168, Dec. 1968.
- [4] L. Zadeh
“Fuzzy sets,”
Inf. Control, vol. 8, no. 3, pp. 338–353, 1965.
- [5] D. Dubois and H. Prade,
Eds.
Fundamentals of Fuzzy Sets. New York, NY: Springer Sciences+Business Media, 2000.
- [6] D. Dubois and H. Prade
Fuzzy Sets and Systems: Theory and Applications New York, NY: Academic Press, 1980.
- [7] D. Dubois and H. Prade
“Possibility theory and its applications: where do we stand?,”
in *Springer Handbook of Computational Intelligence*, Berlin, Heidelberg: Springer, 2015, pp. 1–31.
- [8] G. Shafer
A Mathematical Theory of Evidence. Princeton, NJ: Princeton University Press, 1976.
- [9] R.R. Yager, J.M. Kacprzyk, and M. Fedrizzi (Editors)
Advances in the Dempster-Shafer Theory of Evidence. New York, NY: John Wiley & Sons, 1994.
- [10] R. R. Yager and L. Liu,
Eds.
Classical Works of the Dempster-Shafer Theory of Belief Functions. In *Studies in Fuzziness and Soft Computing*, Vol. 219, Berlin, Heidelberg: Springer, 2008.
- [11] A. Dempster
“Upper and lower probabilities induced by a multivalued mapping,”
Ann. Math. Statist., vol. 38, no. 2, pp. 325–339, Apr. 1967.
- [12] A. Dempster
“A generalization of Bayesian inference,”
J R Stat Soc Series B, vol. 30, no. 2, pp. 205–245, 1968.
- [13] R. Yager
“On the Dempster-Shafer framework and new combination rules,”
Inf. Sci., Vol. 41, no. 2, pp. 93–138, Mar. 1987.
- [14] T. Denœux
“40 years of research on Dempster-Shafer theory: a retrospective,”
Int. J. Approx. Reason, Special Issue C, vol. 79, pp. 1–6, Dec. 2016.
- [15] F. Cuzzolin
The Geometry of Uncertainty. Switzerland AG: Springer Nature, 2021.
- [16] K. Sentz and S. Ferson
Combination of evidence in Dempster-Shafer theory.
SANDIA Tech. Rep. SAND2002-0835, 96 pages, Apr. 2002.
- [17] T. George and N.R. Pal
“Quantification of conflict in Dempster-Shafer framework: a new approach,”
Int. J. Gen. Syst., vol. 24, no. 4, pp. 407–423, Mar. 1996.
- [18] M.J. Wierman
“Measuring conflict in evidence theory,”
in *Proc. of IFSA World Congr. and 20th NAFIPS Int. Conf.*, vol. 3, no. 21, pp. 1741–1745, 2001.

- [19] W. Liu
“Analyzing the degree of conflict among belief functions,”
Artif. Intell., vol. 170, pp. 909–924, 2006.
- [20] P. Smets
“Analyzing the combination of conflicting belief functions,”
Inf. Fusion, Vol. 8, no. 4, pp. 387–412, Oct. 2007.
- [21] A. Martin, A.-L. Jousselme, and C. Osswald
“Conflict measure for the discounting operation on belief functions,”
in *Proc. of Fusion 2008 Int. Conf. on Inf. Fusion*, Cologne, Germany, 2008.
- [22] A. Martin
“About conflict in the theory of belief functions,”
in *Belief Functions: Theory and Applications. Advances in Intelligent and Soft Computing*, T. Denœux & M.H. Masson, Eds., Vol 164. Springer, Berlin, Heidelberg, 2012.
- [23] J. Schubert
“The internal conflict of a belief function,”
in *Proc. of the 2nd Int. Conf. on Belief Functions*, Compiègne, France, May 2012.
- [24] S. Destercke and T. Burger
“Toward an axiomatic definition of conflict between belief functions,”
IEEE Trans. on Cybernetics, vol. 43, no. 2, pp. 585–596, Apr. 2013.
- [25] T. Burger
“Geometric views on conflicting mass functions: from distances to angles,”
Int J Approx Reason, vol. 70, no. C, pp. 36–50, Mar. 2016.
- [26] M. Daniel and V. Kratochvíl
“On hidden conflicts of belief functions,”
in *Proc. of 11th Conf. of the Eur. Soc. for Fuzzy Log. and Technol. (EUSFLAT 2019)*, Prague, Czech Republic, Sept. 2019.
- [27] A. Martin
“Conflict management in information fusion with belief functions,”
in *Information Quality in Information Fusion and Decision Making*, Eloi Bossé & Galina L. Rogova, Eds., Berlin, Heidelberg: Springer Verlag, Apr. 2019, pp. 79–97.
- [28] L.A. Zadeh
“On the validity of Dempster’s rule of combination of evidence,” ERL Memo M79/24, Department of EECS, Univ. of California, Berkeley, CA, U.S.A., 1979.
- [29] L.A. Zadeh
“A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination,”
AI Mag., vol. 7, no. 2, pp. 85–90, 1986.
- [30] J. Dezert and A. Tchamova
“On the validity of Dempster’s fusion rule and its interpretation as a generalization of Bayesian fusion rule,”
Int. J. of Intelligent Syst., vol. 29, no. 3, pp. 223–252, Dec. 2014.
- [31] F. Smarandache and J. Dezert
“Proportional conflict redistribution rules for information fusion,”
in *Advances and Applications of DSMT for Information Fusion*, Vol. 2, Rehoboth, NM, U.S.A: American Research Press, 2006, ch. 1.
- [32] A. Martin and C. Osswald
“A new generalization of the proportional conflict redistribution rule stable in terms of decision,”
in *Advances and Applications of DSMT for Information Fusion*, Vol. 2, Rehoboth, NM, U.S.A: American Research Press, 2006, ch. 2.
- [33] A. Martin, C. Osswald, J. Dezert, and F. Smarandache
“General combination rules for qualitative and quantitative beliefs,”
J. Adv. Inf. Fusion, vol. 3, no. 2, pp. 67–89, Dec. 2008.
- [34] F. Smarandache and J. Dezert, Eds.
Advances and Applications of DSMT for Information Fusion, Vols. 1–4. Rehoboth, NM, U.S.A: American Research Press, 2004–2015.
- [35] P. Halmos
Naive Set Theory. New York, NY: D. Van Nostrand Company Inc., 1960.
- [36] T. Denœux
“The cautious rule of combination for belief functions and some extensions,”
in *Proc. of Fusion 2006 Int. Inform. Fusion Conf.*, Firenze, Italy, July 2006.
- [37] I. Kramosil
“Probabilistic analysis of Dempster combination rule,”
in *Probabilistic Analysis of Belief Functions*. New York: Springer-Science+Business Media, 2001, ch. 6.
- [38] F. Cuzzolin
“Geometry of Dempster’s Rule of Combination,”
IEEE Trans. on SMC, Part B: Cybernetics, vol. 34, no. 2, pp. 961–977, Apr. 2004.
- [39] T. Dezert, Y. Fargier, S. Palma-Lopes, and P. Côte
“Levee characterization by means of data fusion of in-situ geophysical and geotechnical information,”
in *Proc. of 25th Eur. Meeting of Environ. and Eng. Geophys.*, Vol. 2019, No. 1, pp. 1–5, European Association of Geoscientists & Engineers, Sept. 2012.
- [40] F. Smarandache
“An in-depth look at quantitative information fusion rules,”
in *Advances and Applications of DSMT for Information Fusion*, vol. 2, Rehoboth, NM, U.S.A: American Research Press, 2006, ch. 8.
- [41] F. Smarandache and J. Dezert
“Importance of sources using repeated fusion with the proportional conflict redistribution rules #5 and #6,”
in *Multispace & Multistructure. Neutrosophic Transdisciplinarity (100 Collected Papers of Sciences)*, vol. IV. Finland, North-European Scientific Publishers, 2010.
- [42] J. Dezert and F. Smarandache
“Canonical decomposition of dichotomous basic belief assignment,”
Int. J. Intell. Syst., vol. 35, no. 7, pp. 1–21, Jul. 2020.
- [43] J. Dezert, F. Smarandache, A. Tchamova, and D. Han
“Fast fusion of basic belief assignments defined on a dichotomous frame of discernment,”
in *Proc. of Fusion 2020*, Pretoria, South Africa, Jul. 2020.
- [44] F. Smarandache, J. Dezert, and J.-M. Tacnet
“Fusion of sources of evidence with different importances and reliabilities,”
in *Proc. of Fusion 2010*, Edinburgh, Scotland, UK, 26–29 Jul. 2010.
- [45] Arnaud Martin’s toolbox. [Online]. Available: <http://people.irisa.fr/Arnaud.Martin/toolboxes/>
- [46] The Belief Functions and Applications Society. [Online]. Available: <https://www.bfasociety.org/#software>
- [47] The Comprehensive R Archive Network. [Online]. Available: <https://CRAN.R-project.org/package=ibelief>
- [48] T. Dezert and J. Dezert
“Improvement of proportional conflict redistribution fusion rules for levee characterization,”
in *Proc. of ESREL 2021 Int. Conf.*, Angers, France, Sept. 2021.



Théo Dezert was a Postdoctoral Researcher at the Gustave Eiffel University in Bouguenais (France) from March 2020 to May 2021 and will be a Postdoctoral Researcher for two years from October 2021, under the supervision of Professor Leif Lia and Associate Professor Fjola Gudrun Sigtryggsdottir at Norwegian University of Science and Technology (Norway). He obtained the bachelor's degree in Earth and environmental sciences from Tours University in 2014 and the master's degree in planetary geosciences from Nantes University in 2016. During his PhD, obtained in 2019, working with Cerema and Ifsttar French institutes, he developed a fusion methodology to combine spatialized geophysical and punctual geotechnical data in order to improve dike diagnosis. This methodology relies on the use of belief functions. He has collaborated with EDF (French Electric utility company) in order to apply this methodology to their hydraulic works. In 2021, he has co-organized the francophone “First National Information Fusion Seminar for the Characterization of Porous Media: Applications to Geosciences and Civil Engineering” with the French INRAE Institute and Clermont-Auvergne University. He has published three international articles and participated in over ten international and national conferences, he has also been reviewing geophysical articles for international journals.



Jean Dezert graduated from EFREI Engineering School in Paris in 1985, and obtained the Ph.D degree in automatic control and signal processing from the University Paris XI, Orsay, France in 1990. During 1986–1990, he did research in multi-sensor multi-target tracking (MS-MTT) at the French Aerospace Research Lab (ONERA), Châtillon, France. During 1991–1992, he visited the ESE Department at UConn University in USA as Post-doc Research Fellow under supervision of Professor Bar-Shalom. During 1992–1993, he was a Teaching Assistant in Electrical Engineering Department at Orléans University, France. He joined ONERA in 1993, where he is a Maître de Recherches and Senior Research Scientist. His current research interests include estimation theory with applications to MS-MTT, information fusion, plausible reasoning under uncertainty, and decision-making support. He is a Reviewer for several international journals. He has worked for the development of the International Society of Information Fusion (www.isif.org) and served in its Board. He was President of ISIF in 2016. Dr. Dezert has been involved in the Technical Program Committees of several International Conferences on Information Fusion. He gave several seminars, lectures, and tutorials on information fusion and tracking in Europe, USA, Canada, Australia, and China during the last two decades. He has published several book chapters, 160 papers in conferences, and around 70 papers in journals. He has also co-edited several books with Professor Smarandache on information fusion related to DSMT for information fusion. For more information, see www.onera.fr/fr/staff/jean-dezert.



Florentin Smarandache is a Professor of Mathematics at the University of New Mexico, USA. He got the MSc degree in mathematics and computer science from the University of Craiova, Romania, PhD in mathematics from the State University of Kishinev, and postdoctoral in applied mathematics from Okayama University of Sciences, Japan. He is the Founder of Neutrosophy (generalization of dialectics), Neutrosophic Set, Logic, Probability and Statistics since 1995 and has published hundreds of papers and books on neutrosophic physics, superluminal and instantaneous physics, unmatter, quantum paradoxes, absolute theory of relativity, redshift and blueshift due to the medium gradient and refraction index besides the Doppler effect, paradoxism, outerart, neutrosophy as a new branch of philosophy, law of included multiple-middle, multispace and multistructure, hypersoft set, degree of dependence and independence between neutrosophic components, refined neutrosophic set, neutrosophic over-under-off-set, plithogenic set, neutrosophic triplet and duplet structures, quadruple neutrosophic structures, extension of algebraic structures to NeutroAlgebras and AntiAlgebras, DSmT, and so on to many peer-reviewed international journals and many books and he presented papers and plenary lectures to many international conferences around the world. In addition, he published many books of poetry, dramas, children' stories, translations, essays, novel, folklore collections, traveling memories, and art albums. For more information, see fs.unm.edu/FlorentinSmarandache.htm.

INTERNATIONAL SOCIETY OF INFORMATION FUSION

ISIF Website: <http://www.isif.org>

2021 BOARD OF DIRECTORS*

2018–2020	2019–2021	2020–2022
Fredrik Gustafsson	Kathryn Laskey	Pieter De Villiers
X. Rong Li	Felix Govaers	Murat Efe
Zhansheng Duan	Simon Maskell	Wolfgang Koch

*Board of Directors are elected by the members of ISIF for a three year term.

PAST PRESIDENTS

Paulo Costa, 2019	Joachim Biermann, 2011	Xiao-Rong Li, 2003
Lyudmila Mihaylova, 2018	Stefano Coraluppi, 2010	Yaakov Bar-Shalom, 2002
Lyudmila Mihaylova, 2017	Elisa Shahbazian, 2009	Pramod Varshney, 2001
Jean Dezert, 2016	Darko Musicki, 2008	Yaakov Bar-Shalom, 2000
Darin Dunham, 2015	Erik Blasch, 2007	Jim Llinas, 1999
Darin Dunham, 2014	Pierre Valin, 2006	Jim Llinas, 1998
Wolfgang Koch, 2013	W. Dale Blair, 2005	
Roy Streit, 2012	Chee Chong, 2004	

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 are encouraged to submit their manuscripts for peer review <http://isif.org/journal>.

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.