December 2014 Volume 9 Number 2 ISSN 1557-6418

JAIF 4

Journal of Advances in Information Fusion

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

ISIF

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 (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.

From the Editor-in-Chief:

December 2014

JAIF Special Issues

In order to make the journal more attractive, JAIF aims at regularly publishing and encouraging proposals for special issues from highly qualified guest editors. Special issues are an effective means for communicating state-of-the-art knowledge in a specific research field to interested readers, and are intended to cover hot research topics with a large audience in the area of information fusion.

In addition to being of high value to readers, a special issue also gives distinct advantages for the guest editors and authors of individual papers. The guest editors gain international visibly in the research field and build their reputation, which is especially helpful for junior scientists. For authors, special issues provide a convenient publication platform for their research results, where the embedding into similar research topics typically leads to an increased number of readers and a higher level of citations.

Articles in a special issue are both invited and openly submitted based on the general Call-for-Papers. These may include regular papers and short correspondence papers, as deemed appropriate by the guest editors. Organizers of special sessions at ISIF's annual FUSION conference are welcome to propose a JAIF Special Issue based on extended versions of special session papers presented at the conference. All manuscripts for a special issue must meet the same standards as regular submissions and undergo exactly the same peer review procedure.

Potential guest editors should submit a formal proposal for expressing their desire to organize a special issue. These proposals should provide the biographical information of the guest editors, the motivation for the special issue, a general call for contributed papers to the special issue, a candidate list of papers with authors for direct invitation, and a schedule for delivery and review of the papers for the special issue. Typically, one or two guest editors will be responsible for overseeing the peer reviews of each special issue, and they will be supported by a small team of predefined reviewers in order to expedite the review process. Additional reviewers will be recruited from the general JAIF reviewer database.

The editorial board of JAIF makes an accept/reject decision based on the information in the formal proposal. The selection is based on the relevance of the proposed topic, its fit to the ISIF community, its interest to readers of JAIF, its potential for both attracting new authors and increasing the readership, the broadness and quality of the papers in the candidate list, and the qualifications of the guest editors. In addition, a special issue should be well-timed and focus on topics with significant publications in the last five years. Topics already treated in special issues of other journals in recent years should be avoided.

After acceptance of the proposal, an official Call-for-Papers is assembled by the editorial board in close cooperation with the guest editors based on the material provided. The Call-for-Papers will then be posted at http://www.isif.org and distributed to all ISIF members.

The timeline for a special issue details the steps required from posting the Call-for-Papers to the final publication. This includes a submission deadline for potential authors, review deadlines, and deadlines for the revision of accepted papers. All deadlines should include an official date and a buffer. In addition, a time slot for final copy editing of accepted papers should be allocated that involves a reading/annotation phase by a copy editor and another subsequent revision phase by the authors. A typical publication timeline for JAIF manuscripts can be found in the editorial of JAIF, Volume 5, Number 1.

A published special issue will contain the final accepted papers together with an introduction provided by the guest editors. This introduction should comprise a summary of the topic, a synopsis for each article, and biographical information of the guest editors.

Currently, there is an open Call-for-Papers for a special issue on "Nonlinear Derivative-Free Filters: Theory and Applications". Guest editors are Jindrich Dunik and Ondrej Straka from the Department of Cybernetics, Faculty of Applied Sciences, University of West Bohemia in Pilsen, Czech Republic. The deadline for paper submission is March 31, 2015.

Future special issues are planned in the areas of "Extended Object and Group Tracking", "Estimation Involving Directional Quantities", and "Probabilistic RGBD Data Fusion".

Uwe D. Hanebeck
Editor-in-Chief

Bias Estimation and Observability for Optical Sensor Measurements with Targets of Opportunity

DJEDJIGA BELFADEL
RICHARD W. OSBORNE, III
YAAKOV BAR-SHALOM

In order to carry out data fusion, registration error correction is crucial in multisensor systems. This requires estimation of the sensor measurement biases. It is important to correct for these bias errors so that the multiple sensor measurements and/or tracks can be referenced as accurately as possible to a common tracking coordinate system. This paper provides a solution for bias estimation of multiple passive sensors using common targets of opportunity. The measurements provided by these sensors are assumed time-coincident (synchronous) and perfectly associated. The Line of Sight (LOS) measurements from the sensors can be fused into "composite" measurements, which are Cartesian target positions, i.e., linear in the target state. We evaluate the Cramér-Rao Lower Bound (CRLB) on the covariance of the bias estimates, which serves as a quantification of the available information about the biases. Statistical tests on the results of simulations show that this method is statistically efficient, even for small sample sizes (as few as three sensors and three points on the trajectory of a single target of opportunity). We also show that the Root Mean Squared (RMS) position error is significantly improved with bias estimation compared with the target position estimation using the original biased measurements. Bias observability issues, which arise in the case of two sensors, are also discussed.

Manuscript received August 12, 2013; revised November 18, 2013 and February 13, 2014; released for publication April 14, 2014.

Refereeing of this contribution was handled by Ramona Georgescu.

Funding provided by ARO W911NF-10-1-0369.

Authors' address: Electrical and Computer Engineering, University of Connecticut, Storrs, CT, U.S.A. Email: (djedjiga.belfadel@uconn.edu, rosborne@engr.uconn.edu, ybs@engr.uconn.edu)

I. INTRODUCTION

Multisensor systems use fusion of data from multiple sensors to form accurate estimates of a target track. To fuse multiple sensor data the individual sensor data must be expressed in a common reference frame. A problem encountered in multisensor systems is the presence of errors due to sensor bias. Some sources of bias errors include: measurement biases due to the deterioration of initial sensor calibration over time; attitude errors caused by biases in the gyros of the inertial measurement units of (airborne, seaborne, or spaceborne) sensors; and timing errors due to the biases in the onboard clock of each sensor platform [11].

The effect of biases introduced in the process of converting sensor measurements from polar (or spherical) coordinates to Cartesian coordinates has been discussed extensively in [2] together with the limit of validity of the standard transformation. If the conversion process is unbiased, the performance of a converted measurement Kalman filter is superior to a mixed coordinate Extended Kalman Filter EKF (i.e., target motion in Cartesian coordinates and measurements in polar coordinates) [2]. The approaches for conversion include the conventional conversion, the Unbiased Converted Measurement (UCM), the Modified Unbiased Converted Measurement (MUCM), and the Unscented Transform (UT). Recently, a decorrelated version of the UCM technique (DUCM) has been developed to address both conversion and estimation bias [8], [9]. Another example of biased measurement conversion is the estimation of range-rate from a moving platform. To measure range rate using the Doppler effect, it is necessary to nullify the impact of platform motion. The conventional nullification approach suffers from a similar bias problem as the position measurement conversion [3]. A novel scheme was proposed in [6] and [7] by applying the DUCM technique to own-Doppler nullification to eliminate this bias.

Time varying bias estimation based on a nonlinear least squares formulation and the singular value decomposition using truth data was presented in [11]. However, this work did not discuss the CRLB for bias estimation. An approach using Maximum a Posteriori (MAP) data association for concurrent bias estimation and data association based on sensor-level track state estimates was proposed in [12] and extended in [13]. Estimation of location biases only for passive sensors was discussed in [10]. The estimation of range, azimuth, and location biases for active sensors was presented in [14].

For angle-only sensors, imperfect registration leads to LOS angle measurement biases in azimuth and elevation. If uncorrected, registration error can lead to large tracking errors and potentially to the formation of multiple tracks (ghosts) on the same target.

In the present paper, bias estimation is investigated when only targets of opportunity are available. We assume the sensors are synchronized, their locations are

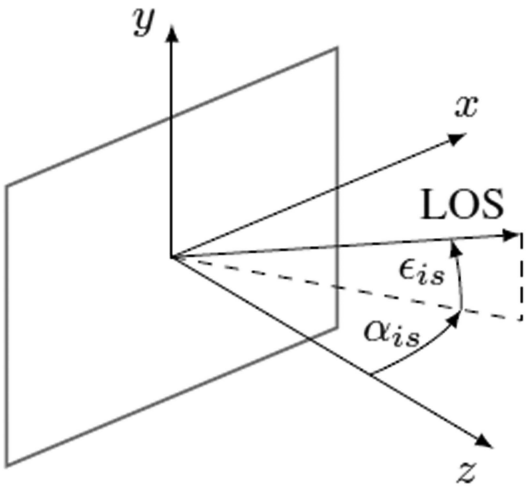


Fig. 1. Optical sensor coordinate system with the origin in the center of the focal plane.

fixed and known, the data association is correct, and we estimate their orientation biases. We investigate the use of the minimum possible number of optical sensors (which can not be less than two sensors). Two cases are considered. In the first case we use three optical sensors to observe three points on the trajectory of a single target of opportunity [4], in the second case we estimate the position of six points on the trajectory of a single target of opportunity simultaneously with the biases of two optical sensors [3]. First, we discuss the observability issues related to the bias estimation. Namely, it is shown that for two fixed sensors there is an inherent ambiguity due to a certain rotation that does not affect the measurements, i.e., one can not have complete observability of the sensor biases with targets of opportunity. For three fixed sensors, the biases are completely observable. We evaluate the Cramér-Rao Lower Bound (CRLB) on the covariance of the bias estimates (for the observable biases), which is the quantification of the available information on the sensor biases and show via statistical tests that the estimation is statistically efficient—it meets the CRLB. Section II presents the problem formulation and solution in detail. Section III describes the simulations performed and gives the results. Finally, Section IV gives the conclusions.

II. PROBLEM FORMULATION

The fundamental frame of reference used in this paper is a 3D Cartesian Common Coordinate System (CCS) defined by the orthogonal set of unit vectors $\{e_x, e_y, e_z\}$. In a multisensor scenario, sensor platform s will typically have a sensor reference frame associated with it (measurement frame of the sensor) defined by the orthogonal set of unit vectors $\{e_{\xi_s}, e_{\eta_s}, e_{\zeta_s}\}$. The origin of the measurement frame of the sensor is a translation of the CCS origin, and its axes are rotated with respect to the CCS axes. The rotation between these frames can be described by a set of Euler angles. We will refer to these angles $\phi_s + \phi_s^n$, $\rho_s + \rho_s^n$, and $\psi_s + \psi_s^n$ of sensor s, as roll, pitch, and yaw respectively [11], where ϕ_s^n is the nominal roll angle, ϕ_s is the roll bias, etc.

Each angle defines a rotation about a prescribed axis, in order to align the sensor frame axes with the CCS axes. The xyz rotation sequence is chosen, which is accomplished by first rotating about the x axis by ϕ_s^n, then rotating about the y axis by ρ_s^n, and finally rotating about the z axis by ψ_s^n. The rotations sequence can be expressed by the matrices

$$\begin{aligned} T_s(\psi_s^n, \rho_s^n, \phi_s^n) &= T_z(\psi_s^n) \cdot T_y(\rho_s^n) \cdot T_x(\phi_s^n) \\ &= \begin{bmatrix} \cos\psi_s^n & \sin\psi_s^n & 0 \\ -\sin\psi_s^n & \cos\psi_s^n & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &\cdot \begin{bmatrix} \cos\rho_s^n & 0 & -\sin\rho_s^n \\ 0 & 1 & 0 \\ \sin\rho_s^n & 0 & \cos\rho_s^n \end{bmatrix} \\ &\cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_s^n & \sin\phi_s^n \\ 0 & -\sin\phi_s^n & \cos\phi_s^n \end{bmatrix} \end{aligned} \quad (1)$$

Assume there are N_S synchronized passive sensors with known fixed position in the CCS, $\boldsymbol{\xi}_s = [\xi_s, \eta_s, \zeta_s]'$, $s = 1, 2, \ldots, N_S$, and N_t targets, located at $\mathbf{x}_i = [x_i, y_i, z_i]'$, $i = 1, 2, \ldots, N_t$, in the same CCS. With the previous convention, the operations needed to transform the position of a given target i expressed in the CCS coordinate into the sensor s coordinate system is

$$\mathbf{x}_{is}^n = T(\boldsymbol{\omega}_s)(\mathbf{x}_i - \boldsymbol{\xi}_s), \quad i = 1, 2, \ldots, N_t, \quad s = 1, 2, \ldots, N_S \quad (2)$$

where $\boldsymbol{\omega}_s = [\phi_s^n, \rho_s^n, \psi_s^n]'$ is the nominal orientation of sensor s and $T(\boldsymbol{\omega}_s)$ is the appropriate rotation matrix and the translation is the difference between the vector position of the target i and the vector position of the sensor s, both expressed in the CCS. The superscript n in (2) indicates that the rotation matrix is based on the nominal sensor orientation.

As shown in Figure 1, the azimuth angle α_{is} is the angle in the sensor xz plane between the sensor z axis and the line of sight to the target, while the elevation angle ϵ_{is} is the angle between the line of sight to the target and its projection onto the xz plane, that is

$$\begin{bmatrix} \alpha_{is} \\ \epsilon_{is} \end{bmatrix} = \begin{bmatrix} \tan^{-1}\left(\dfrac{x_{is}}{z_{is}}\right) \\ \tan^{-1}\left(\dfrac{y_{is}}{\sqrt{x_{is}^2 + z_{is}^2}}\right) \end{bmatrix} \quad (3)$$

The model for the biased noise-free LOS measurements is then

$$\begin{bmatrix} \boldsymbol{\alpha}_{is}^b \\ \boldsymbol{\epsilon}_{is}^b \end{bmatrix} = \begin{bmatrix} g_1(\mathbf{x}_i, \boldsymbol{\xi}_s, \boldsymbol{\omega}_s, \mathbf{b}_s) \\ g_2(\mathbf{x}_i, \boldsymbol{\xi}_s, \boldsymbol{\omega}_s, \mathbf{b}_s) \end{bmatrix} = \mathbf{g}(\mathbf{x}_i, \boldsymbol{\xi}_s, \boldsymbol{\omega}_s, \mathbf{b}_s) \quad (4)$$

where g_1 and g_2 denote the sensor Cartesian coordinates-to-azimuth/elevation angle mapping that can be found by inserting equations (2) and (3) into (4). The bias vector of sensor s is

$$\mathbf{b}_s = [\phi_s, \rho_s, \psi_s]' \tag{5}$$

For a given target, each sensor provides the noisy LOS measurements

$$\mathbf{z}_{is} = \mathbf{g}(\mathbf{x}_i, \boldsymbol{\xi}_s, \boldsymbol{\omega}_s, \mathbf{b}_s) + \mathbf{w}_{is} \tag{6}$$

where

$$\mathbf{w}_{is} = [w_{is}^{\alpha}, w_{is}^{\epsilon}]' \tag{7}$$

The measurement noises $\mathbf{w}_{is}$ are zero-mean, white Gaussian with

$$R_s = \begin{bmatrix} (\sigma_s^{\alpha})^2 & 0 \\ 0 & (\sigma_s^{\epsilon})^2 \end{bmatrix} \tag{8}$$

and are assumed mutually independent.

The problem is to estimate the bias vectors for all sensors and the positions of the targets of opportunity. We shall obtain the Maximum Likelihood (ML) estimate of the augmented parameter vector

$$\boldsymbol{\theta} = [\mathbf{x}_1', \ldots, \mathbf{x}_{N_t}', \mathbf{b}_1', \ldots, \mathbf{b}_{N_S}']' \tag{9}$$

consisting of the (unknown) position of target i and the biases of sensor s, $i = 1, \ldots, N_t, s = 1, \ldots, N_S$, by maximizing the likelihood function

$$\Lambda(\boldsymbol{\theta}) = \prod_{i=1}^{N_t} \prod_{s=1}^{N_S} p(\mathbf{z}_{is} \mid \boldsymbol{\theta}) \tag{10}$$

where

$$p(\mathbf{z}_{is} \mid \boldsymbol{\theta}) = |2\pi R_s|^{-1/2} \cdot \exp(-\tfrac{1}{2}[\mathbf{z}_{is} - \mathbf{h}_{is}(\boldsymbol{\theta})]' R_s^{-1} [\mathbf{z}_{is} - \mathbf{h}_{is}(\boldsymbol{\theta})]) \tag{11}$$

and

$$\mathbf{h}_{is}(\boldsymbol{\theta}) \triangleq \mathbf{g}(\mathbf{x}_i, \boldsymbol{\xi}_s, \boldsymbol{\omega}_s, \mathbf{b}_s) \tag{12}$$

The ML Estimate (MLE) is then

$$\hat{\boldsymbol{\theta}}^{ML} = \arg\max_{\boldsymbol{\theta}} \Lambda(\boldsymbol{\theta}) \tag{13}$$

In order to find the MLE, one has to solve a nonlinear least squares problem for the exponent in (11). This will be done using a numerical search via the Iterated Least Squares (ILS) technique [1].

A. Requirements for Bias Estimability

First requirement for bias estimability. For a given target we have a two-dimensional measurement from each sensor (the two LOS angles to the target). We assume that each sensor sees all the targets at a common time.[1] Stacking together each measurement of N_t targets seen by N_S sensors results in an overall measurement vector of dimension $2N_tN_S$. Given that the position and bias vectors of each target are three-dimensional, and knowing that the number of equations (size of the stacked measurement vector) has to be at least equal to the number of parameters to be estimated (target positions and biases), we must have

$$2N_tN_S \geq 3(N_t + N_S) \tag{14}$$

This is a necessary condition but not sufficient because (13) has to have a unique solution, i.e., the parameter vector has to be estimable. This is guaranteed by the second requirement.

Second requirement of bias estimability. This is the invertibility of the Fisher Information Matrix (FIM) [1], to be discussed later. For example, to estimate the biases of 3 sensors (9 bias components) we need 3 targets (9 position components), i.e., the search is in an 18-dimensional space. In order to estimate the biases of 2 sensors (6 bias components) we need at least 6 targets (18 position components) to meet the necessary requirement (14). The rank of the FIM has to be equal to the number of parameters to be estimated (6 + 18 = 24). The full rank of the FIM is a necessary and sufficient condition for estimability, however, for the two fixed sensors situation this is not satisfied. This issue will be discussed further in the section III.B, where an explanation will be provided.

B. Iterated Least Squares

Given the estimate $\hat{\boldsymbol{\theta}}^j$ after j iterations, the ILS estimate after the $(j+1)$th iteration will be

$$\hat{\boldsymbol{\theta}}^{j+1} = \hat{\boldsymbol{\theta}}^j + [(H^j)' R^{-1} H^j]^{-1} (H^j)' R^{-1} [\mathbf{z} - \mathbf{h}(\hat{\boldsymbol{\theta}}^j)] \tag{15}$$

where

$$\mathbf{z} = [z_{11}', \ldots, z_{is}', \ldots, z_{N_tN_S}']' \tag{16}$$

$$\mathbf{h}(\hat{\boldsymbol{\theta}}^j) = [h_{11}(\hat{\boldsymbol{\theta}}^j)', \ldots, h_{is}(\hat{\boldsymbol{\theta}}^j)', \ldots, h_{N_tN_S}(\hat{\boldsymbol{\theta}}^j)'] \tag{17}$$

R is a block diagonal matrix consisting of N_t blocks of N_S blocks of R_s

$$H^j = \left. \frac{\partial \mathbf{h}(\boldsymbol{\theta}^j)}{\partial \boldsymbol{\theta}} \right|_{\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}^j} \tag{18}$$

is the Jacobian matrix of the vector consisting of the stacked measurement functions (17) w.r.t. (9) evaluated at the ILS estimate from the previous iteration j. In this case, the Jacobian matrix is, with the iteration index omitted for conciseness,

$$H = [H_{11} \quad H_{21} \cdots H_{N_t1} \quad H_{12} \cdots H_{N_tN_S}]' \tag{19}$$

[1] This can also be the same target at different times, as long as the sensors are synchronized.

where

$$H'_{is} = \begin{bmatrix} \frac{\partial g_{1_{is}}}{\partial x_1} & \frac{\partial g_{2_{is}}}{\partial x_1} \\ \frac{\partial g_{1_{is}}}{\partial y_1} & \frac{\partial g_{2_{is}}}{\partial y_1} \\ \frac{\partial g_{1_{is}}}{\partial z_1} & \frac{\partial g_{2_{is}}}{\partial z_1} \\ \vdots & \vdots \\ \frac{\partial g_{1_{is}}}{\partial x_{N_t}} & \frac{\partial g_{2_{is}}}{\partial x_{N_t}} \\ \frac{\partial g_{1_{is}}}{\partial y_{N_t}} & \frac{\partial g_{2_{is}}}{\partial y_{N_t}} \\ \frac{\partial g_{1_{is}}}{\partial z_{N_t}} & \frac{\partial g_{2_{is}}}{\partial z_{N_t}} \\ \frac{\partial g_{1_{is}}}{\partial \psi_1} & \frac{\partial g_{2_{is}}}{\partial \psi_1} \\ \frac{\partial g_{1_{is}}}{\partial \rho_1} & \frac{\partial g_{2_{is}}}{\partial \rho_1} \\ \frac{\partial g_{1_{is}}}{\partial \phi_1} & \frac{\partial g_{2_{is}}}{\partial \phi_1} \\ \vdots & \vdots \\ \frac{\partial g_{1_{is}}}{\partial \psi_{N_S}} & \frac{\partial g_{2_{is}}}{\partial \psi_{N_S}} \\ \frac{\partial g_{1_{is}}}{\partial \rho_{N_S}} & \frac{\partial g_{2_{is}}}{\partial \rho_{N_S}} \\ \frac{\partial g_{1_{is}}}{\partial \phi_{N_S}} & \frac{\partial g_{2_{is}}}{\partial \phi_{N_S}} \end{bmatrix} \tag{20}$$

The appropriate partial derivatives are given in the Appendix.

C. Initialialization

In order to perform the numerical search via ILS, an initial estimate $\hat{\boldsymbol{\theta}}^0$ is required. Assuming that the biases are null, the LOS measurements from the first and the second sensor α_{i1}, α_{i2} and ϵ_{i1} can be used to solve for each initial Cartesian target position, in the CCS, as

$$x_i^0 = \frac{\xi_2 - \xi_1 + \zeta_1 \tan\alpha_{i1} - \zeta_2 \tan\alpha_{i2}}{\tan\alpha_{i1} - \tan\alpha_{i2}} \tag{21}$$

$$y_i^0 = \frac{\tan\alpha_{i1}(\xi_2 + \tan\alpha_{i2}(\zeta_1 - \zeta_2)) - \xi_1 \tan\alpha_{i2}}{\tan\alpha_{i1} - \tan\alpha_{i2}} \tag{22}$$

$$z_i^0 = \eta_1 + \tan\epsilon_{i1} \left| \frac{(\xi_1 - \xi_2)\cos\alpha_{i2} + (\zeta_2 - \zeta_1)\sin\alpha_{i2}}{\sin(\alpha_{i1} - \alpha_{i2})} \right| \tag{23}$$

D. Cramér-Rao Lower Bound

In order to evaluate the efficiency of the estimator, the CRLB must be calculated. The CRLB provides a lower bound on the covariance matrix of an unbiased estimator as [1]

$$E\{(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})'\} \geq J^{-1} \tag{24}$$

where J is the Fisher Information Matrix (FIM), $\boldsymbol{\theta}$ is the true parameter vector to be estimated, and $\hat{\boldsymbol{\theta}}$ is the estimate. The FIM is

$$J = E\{[\nabla_{\boldsymbol{\theta}} \ln\Lambda(\boldsymbol{\theta})][\nabla_{\boldsymbol{\theta}} \ln\Lambda(\boldsymbol{\theta})]'\}|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{true}}} \tag{25}$$

where the gradient of the log-likelihood function is

$$\lambda(\boldsymbol{\theta}) \triangleq \ln\Lambda(\boldsymbol{\theta}) \tag{26}$$

$$\nabla_{\boldsymbol{\theta}}\lambda(\boldsymbol{\theta}) = \sum_{i=1}^{N_t}\sum_{s=1}^{N_S} H'_{is} R_{is}^{-1}(\mathbf{z}_{is} - \mathbf{h}_{is}(\boldsymbol{\theta})) \tag{27}$$

which, when plugged into (25), gives

$$\begin{aligned} J &= \sum_{i=1}^{N_t}\sum_{s=1}^{N_S} H'_{is}(R_s^{-1})H_{is}|_{\theta=\boldsymbol{\theta}_{\text{true}}} \\ &= H'(R^{-1})H|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{true}}} \end{aligned} \tag{28}$$

III. SIMULATIONS

A. Three-Sensor Case

We simulated three optical sensors at various fixed and known locations observing a target at three points in time at unknown locations (which is equivalent to viewing three different targets at unknown locations). Five scenarios of three sensors are examined for a set of target locations. They are shown in Figures 2–6. Each scenario is such that each target position can be observed by all sensors. As discussed in the previous section, the three sensor biases were roll, pitch, and yaw angle offsets. The biases for each sensor were set to $1^\circ = 17.45$ mrad. We made 100 Monte Carlo runs for each scenario. In order to establish a baseline for evaluating the performance of our algorithm, we also ran the simulations without biases and with biases, but without bias estimation. The horizontal and vertical Fields of View (FOV) of each sensor are assumed to be 60°. The measurement noise standard deviation σ_s (identical across sensors for both azimuth and elevation measurements) was assumed to be 0.34 mrad (based on an assumed pixel subtended angle of 0.02° (a modest 9 megapixel FPA with FOV $60^\circ \cdot 60^\circ$; $60^\circ/\sqrt{9 \cdot 10^6} = 0.02^\circ$).

1) Description of the Scenarios. The sensors are assumed to provide LOS angle measurements. We denote by $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3$ the 3D Cartesian sensor positions, and $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ the 3D Cartesian target positions (all in CCS). The three target positions are the same for all the scenarios, and they were chosen from a trajectory of a ballistic

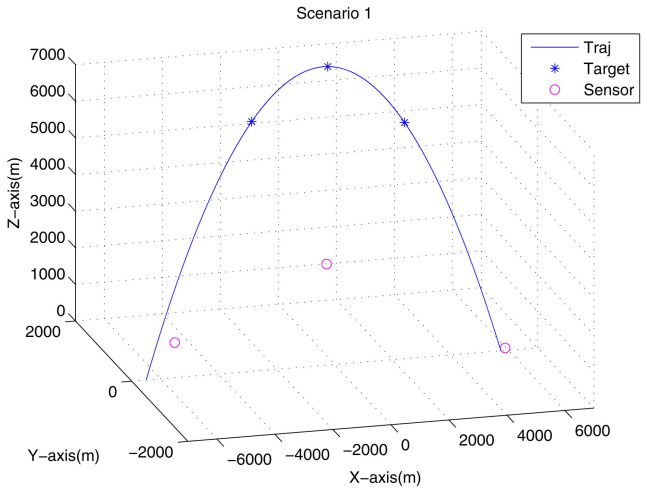


Fig. 2. Scenario 1.

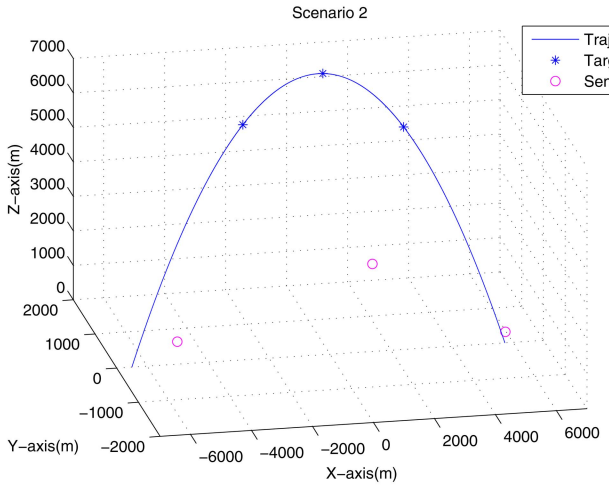


Fig. 3. Scenario 2.

target as follows (in m)

$$\mathbf{x}_1 = [-2860, 0, 6820]' \tag{29}$$

$$\mathbf{x}_2 = [-235.9, 0, 8152]' \tag{30}$$

$$\mathbf{x}_3 = [2413, 0, 6451]' \tag{31}$$

Table I summarizes the sensor positions (in m) for the five scenarios considered.

2) Statistical Efficiency of the Estimates. In order to test for the statistical efficiency of the estimate (of the 18 dimensional vector (9)), the Normalized Estimation Error Squared (NEES) [1] is used, with the CRLB as the covariance matrix. The sample average NEES over 100 Monte Carlo runs is shown in Figure 7 for all scenarios. The NEES is calculated using the FIM evaluated at both the true bias values and target positions, as well as at the estimated biases and target positions. According to the CRLB, the FIM has to be evaluated at the true parameter. Since this is not available in practice, however, it is useful to evaluate the FIM also at the estimated parameter, the only one available in real world

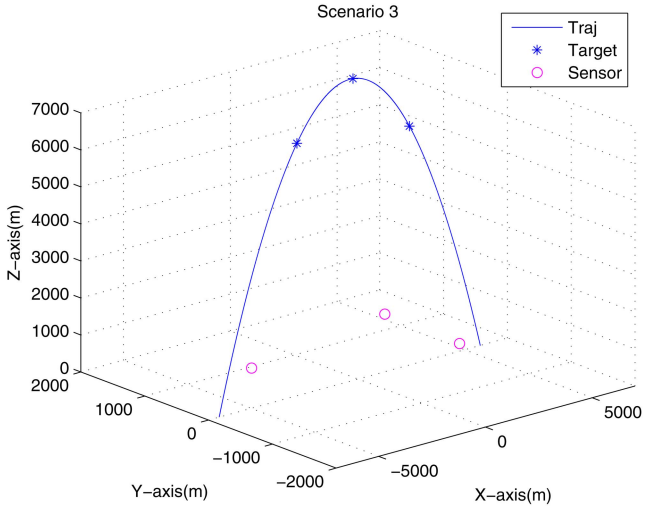


Fig. 4. Scenario 3.

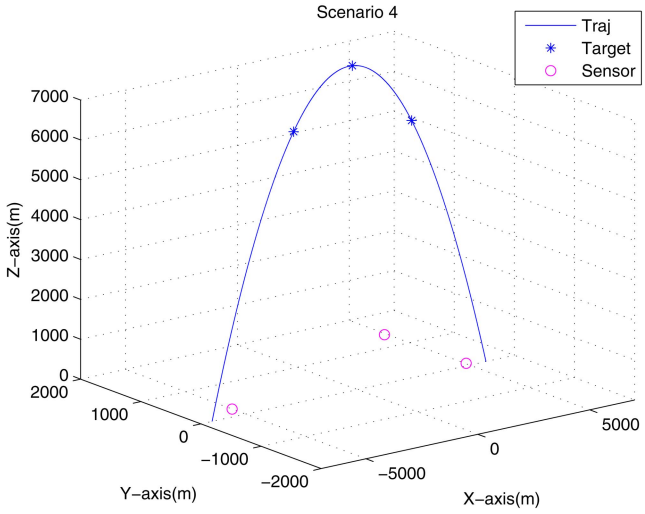


Fig. 5. Scenario 4.

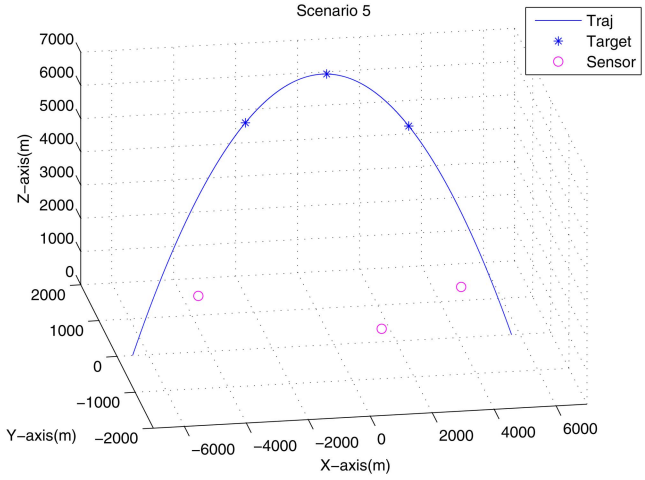


Fig. 6. Scenario 5.

implementations [15], [16]. The results are practically identical regardless of which values are chosen for evaluation of the FIM. The 95% probability region for the

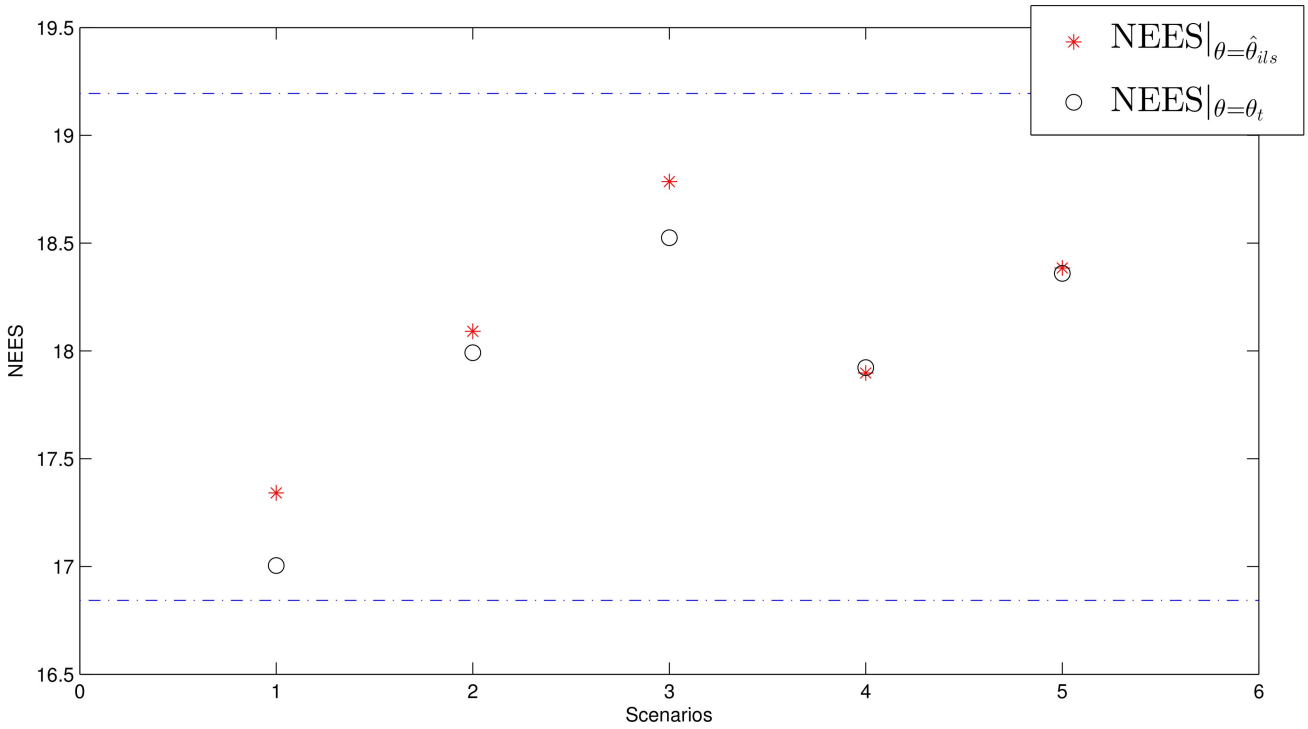


Fig. 7. Sample average NEES over 100 Monte Carlo runs for all five scenarios (three-sensor case).

TABLE I
Sensor positions (in m) for the scenarios considered.

Scenario	First Sensor			Second Sensor			Third Sensor		
	ξ	η	ζ	ξ	η	ζ	ξ	η	ζ
1	−5500	15	950	−230	45	2720	5900	20	50
2	−4900	145	505	1230	−220	2765	5900	200	110
3	−4900	25	1050	1330	25	1585	4900	45	150
4	−5600	5	200	1230	10	1220	4900	20	50
5	−3500	1500	25	1230	−520	1265	4900	1350	20

TABLE II
Sample average position RMSE (in m) for the three targets, over 100 Monte Carlo runs, for the three estimation schemes (three-sensor case).

Scheme	First Target	Second Target	Third Target
	RMSE	RMSE	RMSE
1	3.33	3.51	2.82
2	146.61	167.43	134.80
3	38.93	43.82	37.68

100 sample average NEES of the 18 dimensional parameter vector is $[16.84, 19.19]$. For all five scenarios, the NEES is found to be within this interval and the MLE is therefore statistically efficient. Figure 8 shows the individual bias component NEES for all scenarios, The 95% probability region for the 100 sample average single component NEES is $[0.74, 1.29]$. For all five scenarios these NEES are found to be within this interval.

The RMS position errors for the three targets are summarized in Table II. In this table, the first estimation scheme was established as a baseline using bias-free LOS measurements to estimate the target positions.[2] For the second scheme, we used biased LOS measurements but we only estimated target positions. In the last scheme, we used biased LOS measurements and we simultaneously estimated the target positions and sensor biases. Bias estimation yields significantly improved target RMS position errors in the presence of biases.

Each component of $\boldsymbol{\theta}$ should also be individually consistent with its corresponding σ_{CRLB} (the square root of the corresponding diagonal element of the inverse of FIM). In this case, the sample average bias RMSE over 100 Monte Carlo runs should be within 15% of its corresponding bias standard deviation from the

[2] As shown in [15], [16] the unbiased LOS measurements yield composite measurements (full position MLEs) whose errors are zero-mean and their covariance is equal to the corresponding CRLB.

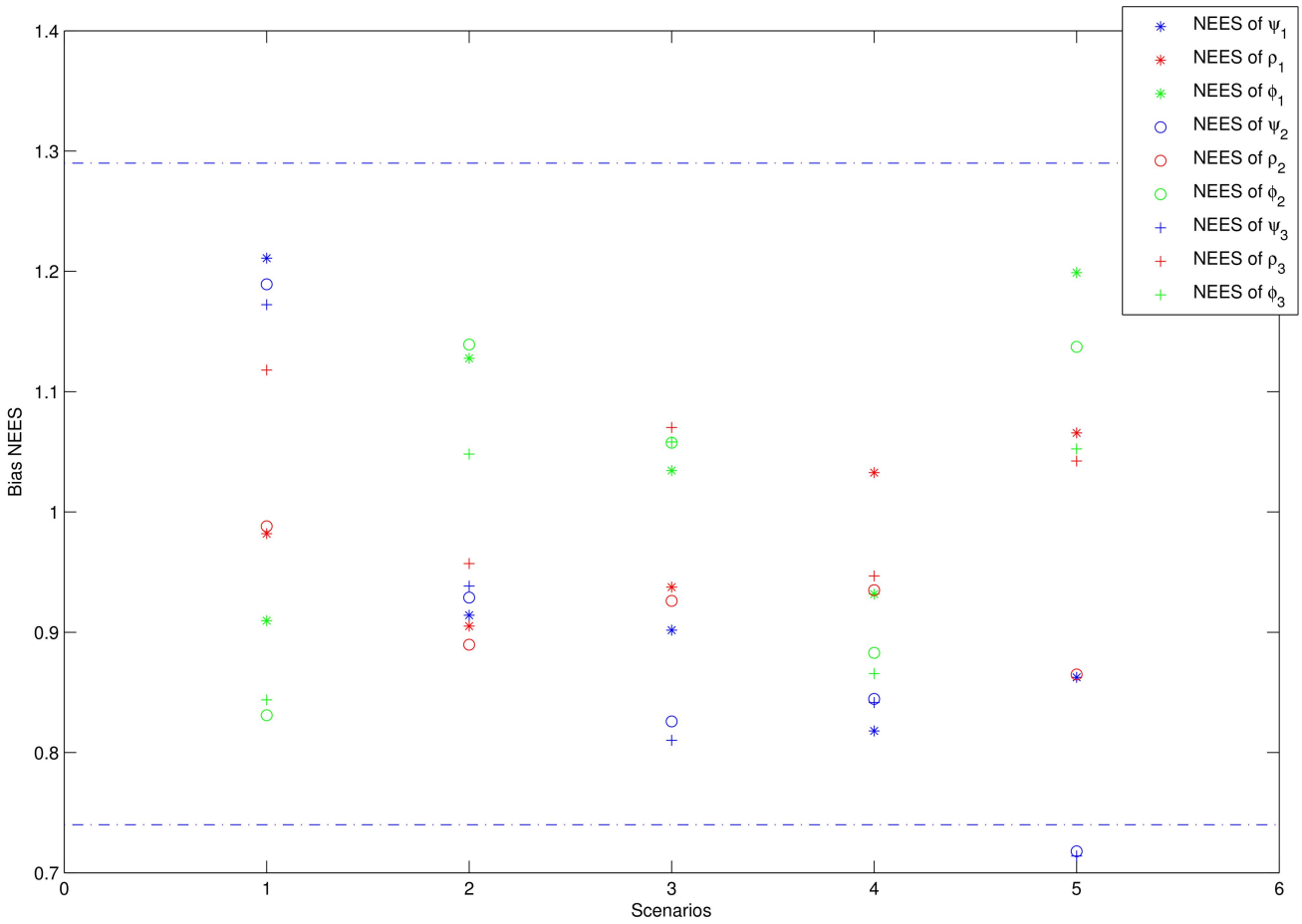


Fig. 8. Sample average bias NEES (CRLB evaluated at the estimate), for each of the 9 biases, over 100 Monte Carlo runs for all five scenarios (three-sensor case).

TABLE III
Sample average bias RMSE over 100 Monte Carlo runs and the corresponding bias standard deviation from the CRLB (σ_{CRLB}), for all configurations (in mrad) (three-sensor case).

Scenario		First Sensor			Second Sensor			Third Sensor		
		ψ	ρ	ϕ	ψ	ρ	ϕ	ψ	ρ	ϕ
1	RMSE	3.168	1.173	2.558	7.358	1.121	3.321	3.210	1.419	2.261
	σ_{CRLB}	2.872	1.183	2.679	6.721	1.129	3.639	2.954	1.341	2.459
2	RMSE	1.935	1.133	2.642	7.573	1.069	3.352	4.224	1.335	1.881
	σ_{CRLB}	2.028	1.190	2.485	7.855	1.129	3.138	4.355	1.362	1.835
3	RMSE	2.473	1.089	5.923	6.475	1.084	6.675	4.504	1.266	5.272
	σ_{CRLB}	2.600	1.124	5.780	7.054	1.140	6.455	4.969	1.239	5.105
4	RMSE	2.512	1.257	5.950	6.472	1.161	6.522	4.579	1.351	5.218
	σ_{CRLB}	2.801	1.243	6.198	7.094	1.201	6.976	5.024	1.388	5.634
5	RMSE	3.102	1.697	4.418	5.979	2.124	5.609	4.238	2.195	3.979
	σ_{CRLB}	3.334	1.646	4.034	7.078	2.295	5.253	5.011	2.150	3.869

CRLB with 95% probability. Table III demonstrates the consistency of the individual bias estimates. This complements the NEES evaluations from Figure 8.

To confirm that the bias estimates are unbiased, the average bias error $\tilde{\bar{b}}$, from Table IV, over 100 Monte Carlo runs confirms that $|\tilde{\bar{b}}|$ is less then $2\sigma_{\text{CRLB}}/\sqrt{N}$ (which it should be with 95% probability), i.e., these bias estimates are unbiased.

In order to examine the statistical efficiency for a variety of target-sensor geometries, the sensors' locations were varied from one scenario to another in order to vary the Geometric Dilution of Precision (GDOP),

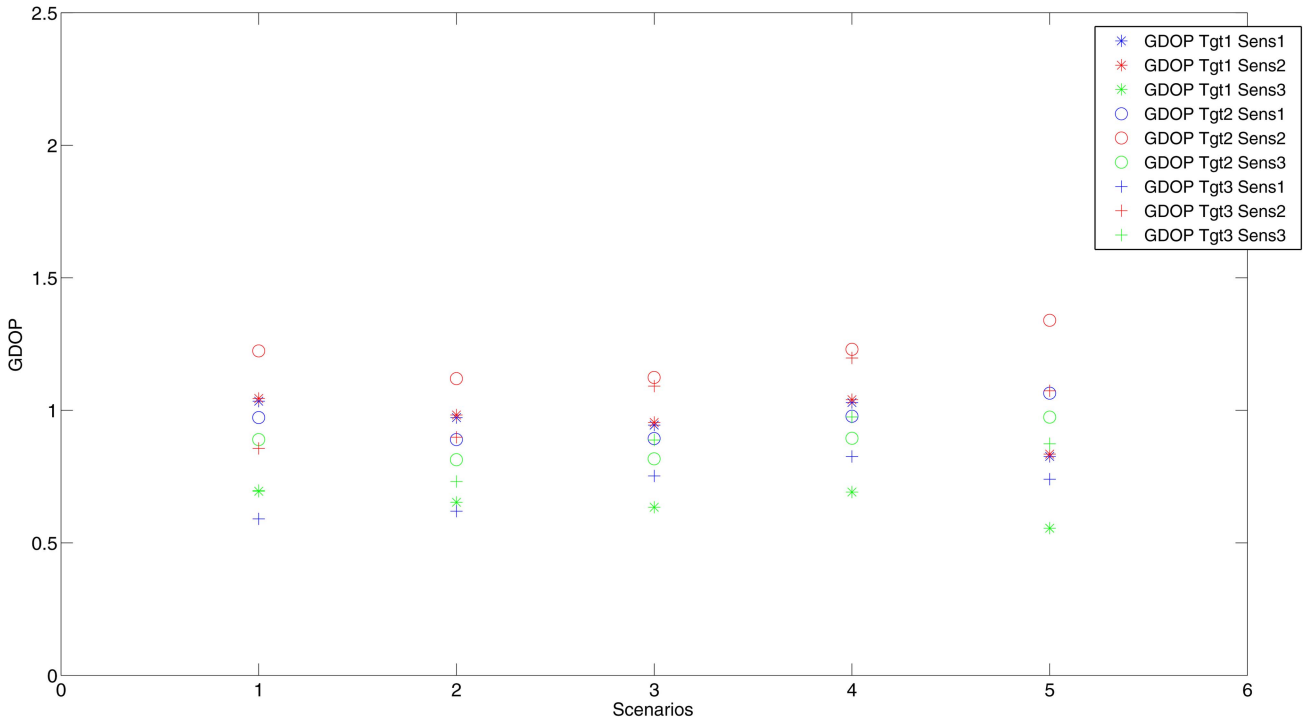


Fig. 9. GDOPs for the five scenarios considered (three-sensor case).

TABLE IV

Sample average bias error $\bar{\tilde{b}}$ over $N = 100$ Monte Carlo runs for all configurations (in mrad) (to confirm that the bias estimates are unbiased) (three-sensor case).

Scenario		First Sensor			Second Sensor			Third Sensor		
		ψ	ρ	ϕ	ψ	ρ	ϕ	ψ	ρ	ϕ
1	$\bar{\tilde{b}}$	0.336	−0.076	0.034	0.693	−0.127	0.128	0.240	−0.111	0.146
	$\frac{\sigma_{\text{CRLB}}}{\sqrt{N}}$	0.287	0.118	0.268	0.672	0.113	0.364	0.295	0.134	0.246
2	$\bar{\tilde{b}}$	−0.099	0.012	0.045	−0.356	0.002	0.017	−0.195	0.088	−0.038
	$\frac{\sigma_{\text{CRLB}}}{\sqrt{N}}$	0.203	0.119	0.248	0.785	0.113	0.314	0.436	0.136	0.184
3	$\bar{\tilde{b}}$	−0.191	0.125	0.039	−0.565	0.134	−0.076	−0.348	0.198	−0.162
	$\frac{\sigma_{\text{CRLB}}}{\sqrt{N}}$	0.260	0.112	0.578	0.705	0.114	0.645	0.497	0.124	0.510
4	$\bar{\tilde{b}}$	0.020	−0.153	−0.481	0.412	−0.094	−0.374	0.345	−0.180	−0.209
	$\frac{\sigma_{\text{CRLB}}}{\sqrt{N}}$	0.280	0.124	0.620	0.709	0.120	0.698	0.502	0.139	0.563
5	$\bar{\tilde{b}}$	0.522	−0.002	−0.058	0.823	0.038	0.034	0.576	−0.009	0.025
	$\frac{\sigma_{\text{CRLB}}}{\sqrt{N}}$	0.333	0.165	0.403	0.708	0.230	0.525	0.501	0.215	0.387

defined as

$$\text{GDOP} \triangleq \frac{\text{RMSE}}{r\sqrt{\sigma_\alpha^2 + \sigma_\epsilon^2}} \tag{32}$$

where "RMSE" is the RMS position error for a target location (in the absence of biases), r is the range to the target, and σ_α and σ_ϵ are the azimuth and elevation measurement error standard deviations, respectively. Figure 9 shows the various GDOP levels in the 9 target-sensor combinations for each of the five scenarios for which statistical efficiency was confirmed.

B. Two-Sensor Case

We simulated two optical sensors at various fixed locations observing a target at six (unknown) locations (which is equivalent to viewing six different targets at unknown locations). In this case a 24-dimensional parameter vector is to be estimated.

It was observed that the rank of the FIM was 23 which implies incomplete observability. Even with more target points there was always a deficiency of 1 in the rank of the FIM. As shown in Figure 10, this can be

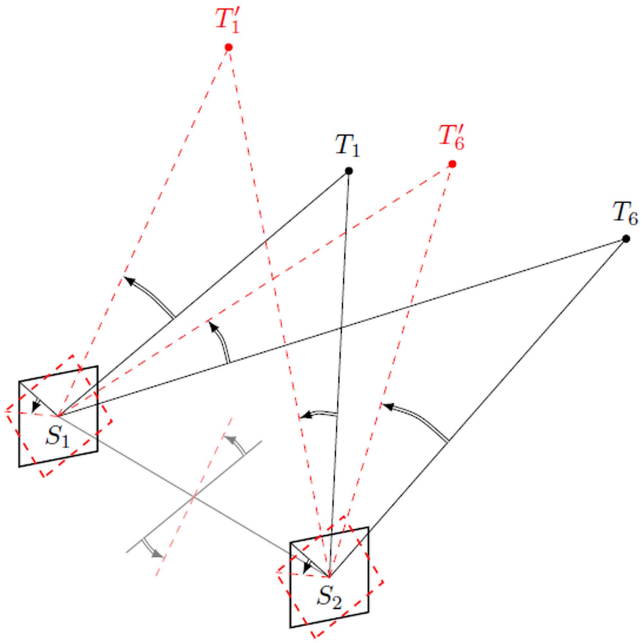


Fig. 10. Rotation around axis S_1S_2 of the sensors and all targets by the same angle leaves all the LOS angles from the sensors to the targets unchanged.

explained as follows: a rotation of the sensors and all the targets around the axis defined by the line S_1S_2 connecting the optical centers of the two sensors is not observable because this will yield the same measurements regardless of the magnitude of this rotation. Note that this rotation does not change the locations of the sensors, which are assumed known. Thus, with two sensors, one cannot estimate all 6 biases—we are limited to estimating 5 and this will be borne out by the FIM in the simulations. A similar observation was made in [5] for sensors that are facing each other. However the above discussion points out that the sensors do not have to face each other—there is an inherent lack of observability of any rotation around the above defined axis. This problem does not exist if there are three or more sensors[3] because there is no axis of rotation that does not change the location of at least one sensor.

Four scenarios of two sensors are examined for a set of target locations. They are shown in Figures 11–14. Each scenario is such that each target position can be observed by all sensors. As discussed in the previous section, the three sensor biases were roll, pitch, and yaw angle offsets. The second sensor roll bias is assumed to be known and null, this is in view of the above discussion about the inherent rank 1 deficiency of the FIM in the two sensors case which makes it impossible to estimate all the 6 sensor biases. Reducing the number of biases from 6 to 5 allows a full rank FIM. All the other biases for each sensor were set to $1^\circ = 17.45$ mrad.

We made 100 Monte Carlo runs for each scenario. In order to establish a baseline for evaluating the performance of our algorithm, we also ran the simulations without bias, and with bias but without bias estimation. The measurement noise standard deviation σ_s (identical across sensors for both azimuth and elevation measurements) was assumed to be 0.34 mrad. As a fifth scenario we simulated two optical sensors observing two targets (two trajectories) at three points in time for each target, as shown in Figure 15.

[3]Provided that the three sensors (or any number of) are not located in a straight line.

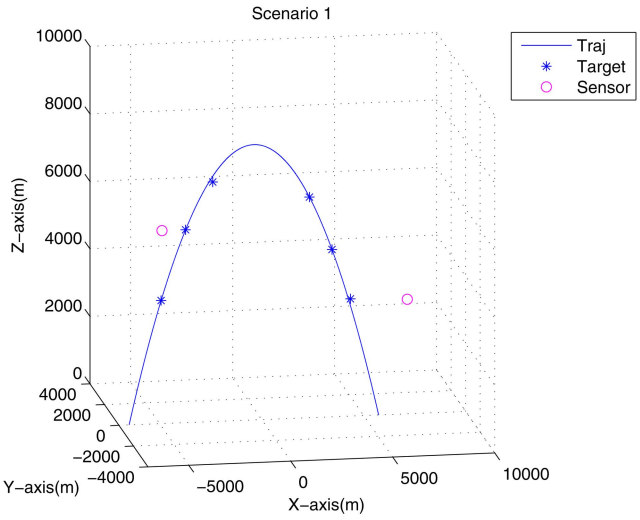


Fig. 11. Scenario 1 for the two-sensor case.

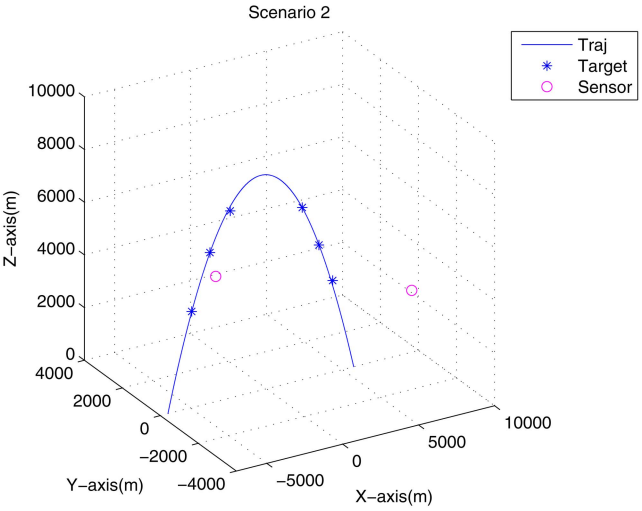


Fig. 12. Scenario 2 for the two-sensor case.

1) Description of the Scenarios. The sensors are assumed to provide LOS angle measurements. We denote by $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2$ the 3D Cartesian sensor positions, and $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6$ the 3D Cartesian target positions (all in CCS). The six target positions are the same for the first four scenarios, and they were chosen from a trajectory of a ballistic target as follows (in m)

$$\mathbf{x}_1 = [-4931, 0, 3649]' \tag{33}$$

$$\mathbf{x}_2 = [-3731, 0, 5714]' \tag{34}$$

$$\mathbf{x}_3 = [-2400, 0, 7100]' \tag{35}$$

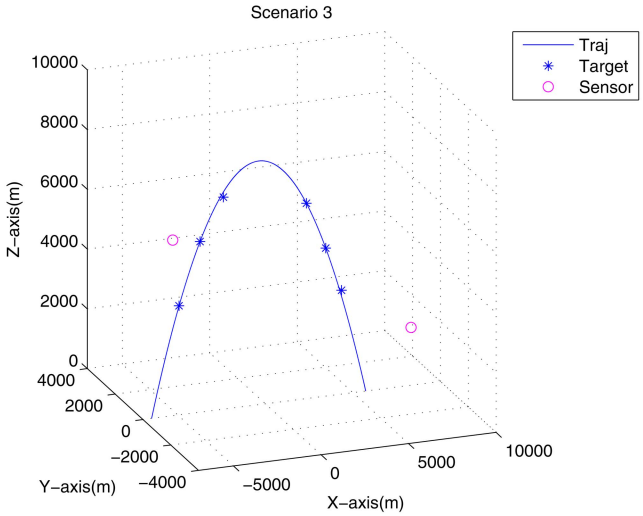


Fig. 13. Scenario 3 for the two-sensor case.

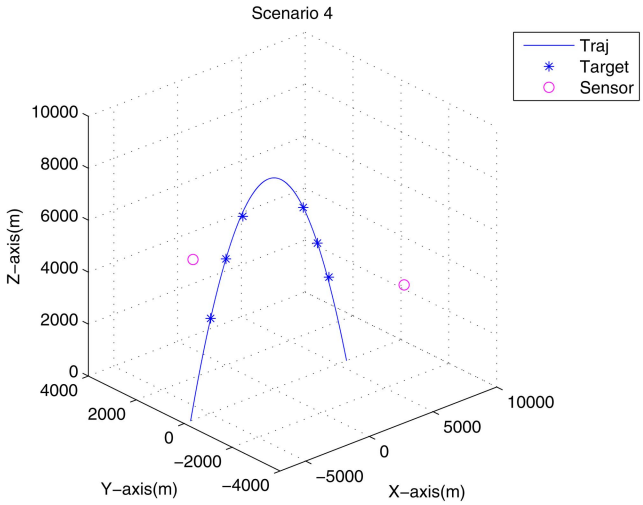


Fig. 14. Scenario 4 for the two-sensor case.

$$\mathbf{x}_4 = [2341, 0, 6538]' \tag{36}$$

$$\mathbf{x}_5 = [3448, 0, 4956]' \tag{37}$$

$$\mathbf{x}_6 = [4351, 0, 3475]' \tag{38}$$

For the fifth scenario, the six target positions were chosen from two trajectories of two ballistic targets as follows (in m)

$$\mathbf{x}_1 = [-4931, 0, 3649]' \tag{39}$$

$$\mathbf{x}_2 = [2994, 0, 5670]' \tag{40}$$

$$\mathbf{x}_3 = [-2400, 0, 7100]' \tag{41}$$

$$\mathbf{x}_4 = [-1400, 0, 7932]' \tag{42}$$

$$\mathbf{x}_5 = [2376, 0, 6497]' \tag{43}$$

$$\mathbf{x}_6 = [4075, 0, 3823]' \tag{44}$$

Table V summarizes the sensor positions (in m) for the five scenarios considered.

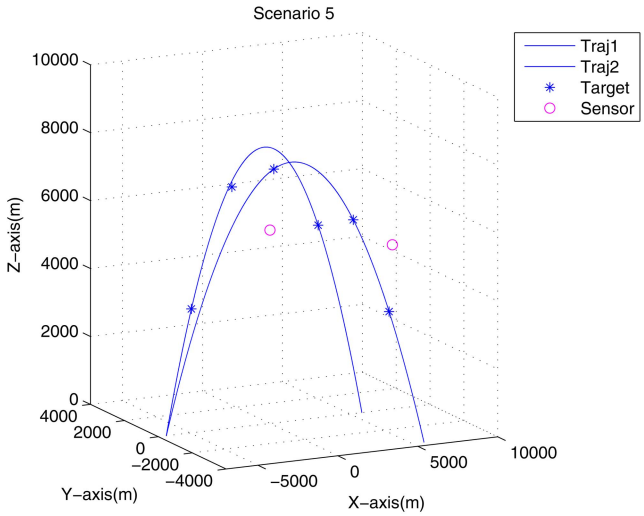


Fig. 15. Scenario 5 for the two-sensor case.

TABLE V
Sensor positions (in m) for the scenarios considered.

Scenario	First Sensor			Second Sensor		
	ξ	η	ζ	ξ	η	ζ
1	−4550	5420	−945	6170	4250	−2700
2	−4550	5420	950	6170	4250	−2700
3	−4550	5420	950	6170	3250	−2700
4	−4550	5420	950	5170	4250	−2700
5	−1550	6120	−1445	6170	5250	−1400

2) Statistical Efficiency of the Estimates. In order to test for the statistical efficiency of the estimate (of the 23 dimensional vector), the NEES is used, with the CRLB as the covariance matrix. The sample average NEES over 100 Monte Carlo runs is shown in Figure 16 for all scenarios. The NEES is calculated using the FIM evaluated at both the true bias values and target positions, as well as at the estimated biases and target positions. The results are practically identical regardless of which values are chosen for evaluation of the FIM. The 95% probability region for the 100 sample average NEES of the 23 dimensional parameter vector is $[21.68, 24.34]$. For all five scenarios these NEES are found to be within this interval and the MLE is therefore statistically efficient. Figure 17 shows the individual bias component NEES for all scenarios, The 95% probability region for the 100 sample average single component NEES is $[0.74, 1.29]$. For all five scenarios these NEES are found to be within this interval.

The RMS position errors for the six targets are summarized in Table VI. In this table, the first estimation scheme was established as a baseline using bias-free LOS measurements to estimate the target positions. For the second scheme, we used biased LOS measurements but we only estimated target positions. In the last scheme, we used biased LOS measurements and we simultaneously estimated the target positions and sensor biases. For the second scheme, the estimation algorithm

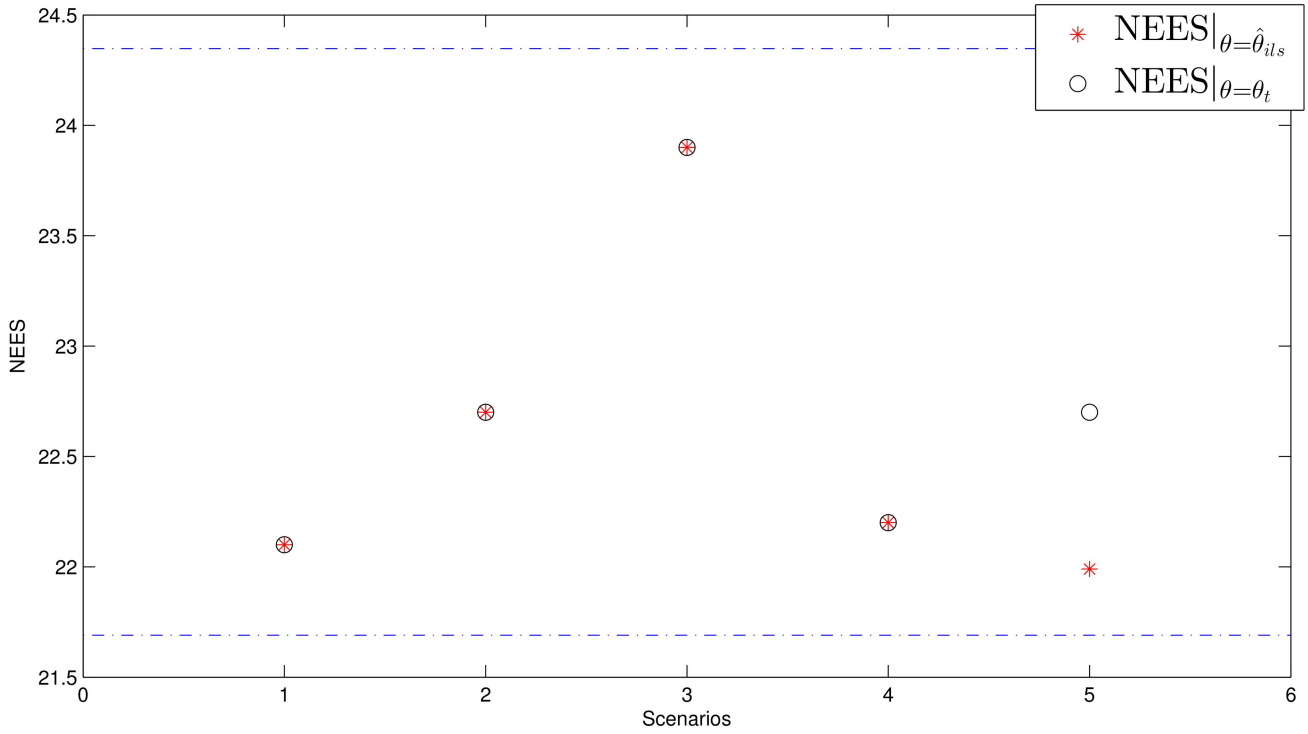


Fig. 16. Sample average NEES over 100 Monte Carlo runs for all five scenarios (two-sensor case).

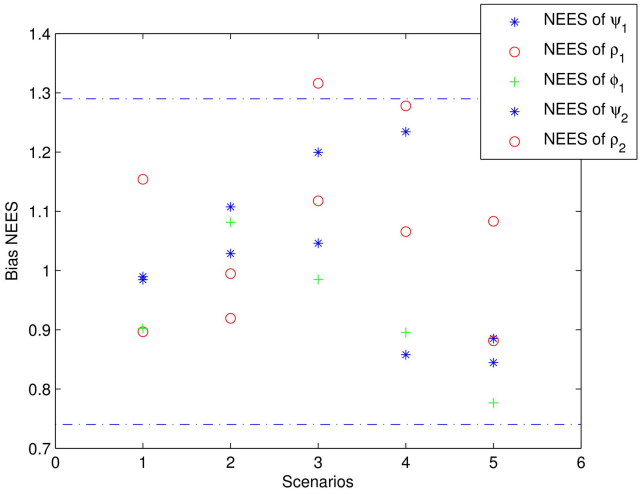


Fig. 17. Sample average bias NEES (CRLB evaluated at the estimate), for each of the five biases, over 100 Monte Carlo runs for all five scenarios (two-sensor case).

does not converge, while the third scheme shows satisfactory target RMS position errors in the presence of biases. The target position RMSE when the biases are also estimated, are close to the RMSE with no biases.

Each component of $\boldsymbol{\theta}$ should also be individually consistent with its corresponding σ_{CRLB} (the square root of the corresponding diagonal element of the inverse of FIM). In this case, the sample average bias RMSE over 100 Monte Carlo runs should be within 15% of its corresponding bias standard deviation from the CRLB (σ_{CRLB}) with 95% probability. Table VII demonstrates the efficiency of the individual bias estimates.

TABLE VI
Sample average position RMSE (in m) for the six targets, over 100 Monte Carlo runs, for the three estimation schemes (two-sensor case).

	First Target	Second Target	Third Target	Fourth Target	Fifth Target	Sixth Target
Scheme	RMSE	RMSE	RMSE	RMSE	RMSE	RMSE
1	3.68	4.84	3.42	4.06	4.64	3.63
3	7.08	7.65	6.49	7.91	7.70	7.76

TABLE VII
Sample average bias RMSE over 100 Monte Carlo runs and the corresponding bias standard deviation from the CRLB, for all configurations (in mrad) (two-sensor case).

		First Sensor			Second Sensor	
Scenario		ψ	ρ	ϕ	ψ	ρ
1	RMSE	0.195	0.271	0.254	0.186	0.314
	σ_{CRLB}	0.252	0.307	0.331	0.238	0.430
2	RMSE	0.437	0.442	0.500	0.428	0.348
	σ_{CRLB}	0.394	0.494	0.441	0.410	0.410
3	RMSE	1.675	1.668	1.634	1.646	0.4615
	σ_{CRLB}	1.279	1.572	1.305	1.207	0.536
4	RMSE	0.475	0.392	0.440	0.465	0.287
	σ_{CRLB}	0.467	0.440	0.510	0.483	0.384
5	RMSE	0.258	0.251	0.237	0.245	0.195
	σ_{CRLB}	0.345	0.246	0.357	0.347	0.168

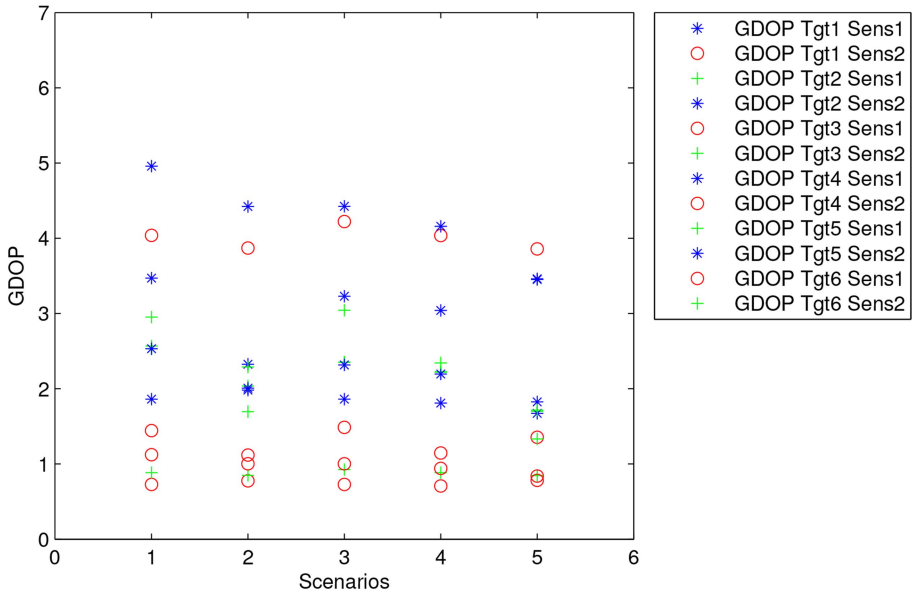


Fig. 18. GDOPs for the five scenarios considered (two-sensor case).

TABLE VIII
Sample average bias error $\bar{\tilde{b}}$ over $N = 100$ Monte Carlo runs for all configurations (in mrad) (to confirm that the bias estimates are unbiased) (two-sensor case).

Scenario		First Sensor			Second Sensor	
		ψ	ρ	ϕ	ψ	ρ
1	$\bar{\tilde{b}}$	0.000	0.007	0.000	0.003	−0.045
	$\frac{\sigma_{\text{CRLB}}}{\sqrt{N}}$	0.025	0.030	0.033	0.023	0.043
2	$\bar{\tilde{b}}$	−0.055	−0.058	−0.007	−0.016	−0.001
	$\frac{\sigma_{\text{CRLB}}}{\sqrt{N}}$	0.039	0.049	0.044	0.041	0.041
3	$\bar{\tilde{b}}$	−0.351	−0.098	−0.254	0.275	0.056
	$\frac{\sigma_{\text{CRLB}}}{\sqrt{N}}$	0.128	0.157	0.130	0.120	0.053
4	$\bar{\tilde{b}}$	−0.001	−0.069	0.042	−0.026	−0.013
	$\frac{\sigma_{\text{CRLB}}}{\sqrt{N}}$	0.046	0.044	0.051	0.048	0.038
5	$\bar{\tilde{b}}$	0.037	0.028	0.006	0.040	−0.005
	$\frac{\sigma_{\text{CRLB}}}{\sqrt{N}}$	0.034	0.024	0.0358	0.034	0.016

To confirm that the bias estimates are unbiased, the average bias error $\bar{\tilde{b}}$, from Table VIII, over 100 Monte Carlo runs confirms that $|\bar{\tilde{b}}|$ is less then $2\sigma_{\text{CRLB}}/\sqrt{N}$ (which it should be with 95% probability), i.e., these estimates are unbiased.

Figure 18 shows the various GDOP levels in the 12 target-sensor combinations for each of the five scenarios for which statistical efficiency was confirmed, in the case of the two sensors.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we presented an algorithm that uses targets of opportunity for estimation of measurement biases. The first step was formulating a general bias model for synchronized optical sensors at fixed known locations. The association of measurements is assumed to be perfect. Based on this, we used a ML approach that led to a nonlinear least-squares estimation problem for simultaneous estimation of the 3D Cartesian positions of the targets of opportunity and the angle measurement biases of the sensors. The bias estimates, obtained via ILS, were shown to be unbiased and statistically efficient. In the three-sensor case it was shown that one has complete observability of the sensor biases. In the two-sensor case a rank deficiency of 1 in the FIM was observed, i.e., this allows estimation of only 5 out of 6 biases. A suitable geometric explanation was provided for this. For moving sensors this problem is expected to go away if the sensors move sufficiently.

APPENDIX

The appropriate partial derivatives of (20) are

$$\frac{\partial g_{1_{is}}}{\partial x_k} = \frac{\partial g_{1_{is}}}{\partial x_{is}}\frac{\partial x_{is}}{\partial x_k} + \frac{\partial g_{1_{is}}}{\partial y_{is}}\frac{\partial y_{is}}{\partial x_k} + \frac{\partial g_{1_{is}}}{\partial z_{is}}\frac{\partial z_{is}}{\partial x_k} \quad (45)$$

$$\frac{\partial g_{1_{is}}}{\partial y_k} = \frac{\partial g_{1_{is}}}{\partial x_{is}}\frac{\partial x_{is}}{\partial y_k} + \frac{\partial g_{1_{is}}}{\partial y_{is}}\frac{\partial y_{is}}{\partial y_k} + \frac{\partial g_{1_{is}}}{\partial z_{is}}\frac{\partial z_{is}}{\partial y_k} \quad (46)$$

$$\frac{\partial g_{1_{is}}}{\partial z_k} = \frac{\partial g_{1_{is}}}{\partial x_{is}}\frac{\partial x_{is}}{\partial z_k} + \frac{\partial g_{1_{is}}}{\partial y_{is}}\frac{\partial y_{is}}{\partial z_k} + \frac{\partial g_{1_{is}}}{\partial z_{is}}\frac{\partial z_{is}}{\partial z_k} \quad (47)$$

$$\frac{\partial g_{1_{is}}}{\partial \psi_k} = \frac{\partial g_{1_{is}}}{\partial x_{is}}\frac{\partial x_{is}}{\partial \psi_k} + \frac{\partial g_{1_{is}}}{\partial y_{is}}\frac{\partial y_{is}}{\partial \psi_k} + \frac{\partial g_{1_{is}}}{\partial z_{is}}\frac{\partial z_{is}}{\partial \psi_k} \quad (48)$$

$$\frac{\partial g_{1_{is}}}{\partial \rho_k} = \frac{\partial g_{1_{is}}}{\partial x_{is}}\frac{\partial x_{is}}{\partial \rho_k} + \frac{\partial g_{1_{is}}}{\partial y_{is}}\frac{\partial y_{is}}{\partial \rho_k} + \frac{\partial g_{1_{is}}}{\partial z_{is}}\frac{\partial z_{is}}{\partial \rho_k} \quad (49)$$

$$\frac{\partial g_{1_{is}}}{\partial \phi_k} = \frac{\partial g_{1_{is}}}{\partial x_{is}}\frac{\partial x_{is}}{\partial \phi_k} + \frac{\partial g_{1_{is}}}{\partial y_{is}}\frac{\partial y_{is}}{\partial \phi_k} + \frac{\partial g_{1_{is}}}{\partial z_{is}}\frac{\partial z_{is}}{\partial \phi_k} \quad (50)$$

$$\frac{\partial g_{2_{is}}}{\partial x_k} = \frac{\partial g_{2_{is}}}{\partial x_{is}}\frac{\partial x_{is}}{\partial x_k} + \frac{\partial g_{2_{is}}}{\partial y_{is}}\frac{\partial y_{is}}{\partial x_k} + \frac{\partial g_{2_{is}}}{\partial z_{is}}\frac{\partial z_{is}}{\partial x_k} \quad (51)$$

$$\frac{\partial g_{2_{is}}}{\partial y_k} = \frac{\partial g_{2_{is}}}{\partial x_{is}}\frac{\partial x_{is}}{\partial y_k} + \frac{\partial g_{2_{is}}}{\partial y_{is}}\frac{\partial y_{is}}{\partial y_k} + \frac{\partial g_{2_{is}}}{\partial z_{is}}\frac{\partial z_{is}}{\partial y_k} \quad (52)$$

$$\frac{\partial g_{2_{is}}}{\partial z_k} = \frac{\partial g_{2_{is}}}{\partial x_{is}}\frac{\partial x_{is}}{\partial z_k} + \frac{\partial g_{2_{is}}}{\partial y_{is}}\frac{\partial y_{is}}{\partial z_k} + \frac{\partial g_{2_{is}}}{\partial z_{is}}\frac{\partial z_{is}}{\partial z_k} \quad (53)$$

$$\frac{\partial g_{2_{is}}}{\partial \psi_k} = \frac{\partial g_{2_{is}}}{\partial x_{is}}\frac{\partial x_{is}}{\partial \psi_k} + \frac{\partial g_{2_{is}}}{\partial y_{is}}\frac{\partial y_{is}}{\partial \psi_k} + \frac{\partial g_{2_{is}}}{\partial z_{is}}\frac{\partial z_{is}}{\partial \psi_k} \quad (54)$$

$$\frac{\partial g_{2_{is}}}{\partial \rho_k} = \frac{\partial g_{2_{is}}}{\partial x_{is}}\frac{\partial x_{is}}{\partial \rho_k} + \frac{\partial g_{2_{is}}}{\partial y_{is}}\frac{\partial y_{is}}{\partial \rho_k} + \frac{\partial g_{2_{is}}}{\partial z_{is}}\frac{\partial z_{is}}{\partial \rho_k} \quad (55)$$

$$\frac{\partial g_{2_{is}}}{\partial \phi_k} = \frac{\partial g_{2_{is}}}{\partial x_{is}}\frac{\partial x_{is}}{\partial \phi_k} + \frac{\partial g_{2_{is}}}{\partial y_{is}}\frac{\partial y_{is}}{\partial \phi_k} + \frac{\partial g_{2_{is}}}{\partial z_{is}}\frac{\partial z_{is}}{\partial \phi_k} \quad (56)$$

Given that (2) can be written as

$$\mathbf{x}_{is} = \begin{bmatrix} x_{is} \\ y_{is} \\ z_{is} \end{bmatrix} = T_s(\mathbf{x}_i - \boldsymbol{\xi}_s) = \begin{bmatrix} T_{s_{11}} & T_{s_{12}} & T_{s_{13}} \\ T_{s_{21}} & T_{s_{22}} & T_{s_{23}} \\ T_{s_{31}} & T_{s_{32}} & T_{s_{33}} \end{bmatrix} \begin{bmatrix} x_i - \xi_s \\ y_i - \eta_s \\ z_i - \zeta_s \end{bmatrix} \quad (57)$$

therefore

$$x_{is} = T_{s_{11}}(x_i - \xi_s) + T_{s_{12}}(y_i - \eta_s) + T_{s_{13}}(z_i - \zeta_s) \quad (58)$$

$$y_{is} = T_{s_{21}}(x_i - \xi_s) + T_{s_{22}}(y_i - \eta_s) + T_{s_{23}}(z_i - \zeta_s) \quad (59)$$

$$z_{is} = T_{s_{31}}(x_i - \xi_s) + T_{s_{32}}(y_i - \eta_s) + T_{s_{33}}(z_i - \zeta_s) \quad (60)$$

and

$$\begin{aligned} \frac{\partial x_{is}}{\partial x_k} &= T_{s_{11}}, \quad \frac{\partial x_{is}}{\partial y_k} = T_{s_{12}}, \quad \frac{\partial x_{is}}{\partial y_k} = T_{s_{13}} \\ \frac{\partial y_{is}}{\partial x_k} &= T_{s_{21}}, \quad \frac{\partial y_{is}}{\partial y_k} = T_{s_{22}}, \quad \frac{\partial y_{is}}{\partial y_k} = T_{s_{23}} \\ \frac{\partial z_{is}}{\partial x_k} &= T_{s_{31}}, \quad \frac{\partial z_{is}}{\partial y_k} = T_{s_{32}}, \quad \frac{\partial z_{is}}{\partial y_k} = T_{s_{33}} \end{aligned} \quad (61)$$

$$\frac{\partial x_{is}}{\partial \psi_k} = \frac{\partial T_{s_{11}}}{\partial \psi_k}(x_i - \xi_s) + \frac{\partial T_{s_{12}}}{\partial \psi_k}(y_i - \eta_s) + \frac{\partial T_{s_{13}}}{\partial \psi_k}(z_i - \zeta_s) \quad (62)$$

$$\frac{\partial x_{is}}{\partial \rho_k} = \frac{\partial T_{s_{11}}}{\partial \rho_k}(x_i - \xi_s) + \frac{\partial T_{s_{12}}}{\partial \rho_k}(y_i - \eta_s) + \frac{\partial T_{s_{13}}}{\partial \rho_k}(z_i - \zeta_s) \quad (63)$$

$$\frac{\partial x_{is}}{\partial \phi_k} = \frac{\partial T_{s_{11}}}{\partial \phi_k}(x_i - \xi_s) + \frac{\partial T_{s_{12}}}{\partial \phi_k}(y_i - \eta_s) + \frac{\partial T_{s_{13}}}{\partial \phi_k}(z_i - \zeta_s) \quad (64)$$

$$\frac{\partial y_{is}}{\partial \psi_k} = \frac{\partial T_{s_{21}}}{\partial \psi_k}(x_i - \xi_s) + \frac{\partial T_{s_{22}}}{\partial \psi_k}(y_i - \eta_s) + \frac{\partial T_{s_{23}}}{\partial \psi_k}(z_i - \zeta_s) \quad (65)$$

$$\frac{\partial y_{is}}{\partial \rho_k} = \frac{\partial T_{s_{21}}}{\partial \rho_k}(x_i - \xi_s) + \frac{\partial T_{s_{22}}}{\partial \rho_k}(y_i - \eta_s) + \frac{\partial T_{s_{23}}}{\partial \rho_k}(z_i - \zeta_s) \quad (66)$$

$$\frac{\partial y_{is}}{\partial \phi_k} = \frac{\partial T_{s_{11}}}{\partial \phi_k}(x_i - \xi_s) + \frac{\partial T_{s_{22}}}{\partial \phi_k}(y_i - \eta_s) + \frac{\partial T_{s_{23}}}{\partial \phi_k}(z_i - \zeta_s) \quad (67)$$

$$\frac{\partial z_{is}}{\partial \psi_k} = \frac{\partial T_{s_{31}}}{\partial \psi_k}(x_i - \xi_s) + \frac{\partial T_{s_{32}}}{\partial \psi_k}(y_i - \eta_s) + \frac{\partial T_{s_{33}}}{\partial \psi_k}(z_i - \zeta_s) \quad (68)$$

$$\frac{\partial z_{is}}{\partial \rho_k} = \frac{\partial T_{s_{31}}}{\partial \rho_k}(x_i - \xi_s) + \frac{\partial T_{s_{32}}}{\partial \rho_k}(y_i - \eta_s) + \frac{\partial T_{s_{33}}}{\partial \rho_k}(z_i - \zeta_s) \quad (69)$$

$$\frac{\partial z_{is}}{\partial \phi_k} = \frac{\partial T_{s_{31}}}{\partial \phi_k}(x_i - \xi_s) + \frac{\partial T_{s_{32}}}{\partial \phi_k}(y_i - \eta_s) + \frac{\partial T_{s_{33}}}{\partial \phi_k}(z_i - \zeta_s) \quad (70)$$

$$\frac{\partial g_{1_{is}}}{\partial x_{is}} = \frac{z_{is}}{z_{is}^2 + x_{is}^2} \quad (71)$$

$$\frac{\partial g_{1_{is}}}{\partial y_{is}} = 0 \quad (72)$$

$$\frac{\partial g_{1_{is}}}{\partial z_{is}} = -\frac{x_{is}}{x_{is}^2 + z_{is}^2} \quad (73)$$

$$\frac{\partial g_{2_{is}}}{\partial x_{is}} = -\frac{x_{is}y_{is}}{\sqrt{(x_{is}^2 + z_{is}^2)}(x_{is}^2 + y_{is}^2 + z_{is}^2)} \quad (74)$$

$$\frac{\partial g_{2_{is}}}{\partial y_{is}} = \frac{\sqrt{x_{is}^2 + z_{is}^2}}{x_{is}^2 + y_{is}^2 + z_{is}^2} \quad (75)$$

$$\frac{\partial g_{2_{is}}}{\partial z_{is}} = -\frac{z_{is}y_{is}}{(x_{is}^2 + y_{is}^2 + z_{is}^2)\left(\sqrt{x_{is}^2 + z_{is}^2}\right)} \quad (76)$$

$$\frac{\partial T_{s_{11}}}{\partial \psi_k} = -\sin\psi_k \cos\rho_k \quad (77)$$

$$\frac{\partial T_{s_{12}}}{\partial \psi_k} = -\sin\psi_k \sin\rho_k \sin\phi_k - \cos\psi_k \cos\phi_k \quad (78)$$

$$\frac{\partial T_{s_{13}}}{\partial \psi_k} = -\sin\psi_k \sin\rho_k \cos\phi_k + \cos\psi_k \sin\phi_k \quad (79)$$

$$\frac{\partial T_{s_{21}}}{\partial \psi_k} = \cos\psi_k \cos\rho_k \quad (80)$$

$$\frac{\partial T_{s_{22}}}{\partial \psi_k} = \cos\psi_k \sin\rho_k \sin\phi_k - \sin\psi_k \cos\phi_k \quad (81)$$

$$\frac{\partial T_{s_{23}}}{\partial \psi_k} = \cos\psi_k \sin\rho_k \cos\phi_k + \sin\psi_k \sin\phi_k \quad (82)$$

$$\frac{\partial T_{s_{31}}}{\partial \psi_k} = 0 \quad (83)$$

$$\frac{\partial T_{s_{32}}}{\partial \psi_k} = 0 \quad (84)$$

$$\frac{\partial T_{s_{33}}}{\partial \psi_k} = 0 \quad (85)$$

$$\frac{\partial T_{s_{11}}}{\partial \rho_k} = -\cos\psi_k \sin\rho_k \tag{86}$$

$$\frac{\partial T_{s_{12}}}{\partial \rho_k} = \cos\psi_k \cos\rho_k \sin\phi_k \tag{87}$$

$$\frac{\partial T_{s_{13}}}{\partial \rho_k} = \cos\psi_k \cos\rho_k \cos\phi_k \tag{88}$$

$$\frac{\partial T_{s_{21}}}{\partial \rho_k} = -\sin\psi_k \sin\phi_k \tag{89}$$

$$\frac{\partial T_{s_{22}}}{\partial \rho_k} = \sin\psi_k \cos\rho_k \sin\phi_k \tag{90}$$

$$\frac{\partial T_{s_{23}}}{\partial \rho_k} = \sin\psi_k \cos\rho_k \cos\phi_k \tag{91}$$

$$\frac{\partial T_{s_{31}}}{\partial \rho_k} = -\cos\phi_k \tag{92}$$

$$\frac{\partial T_{s_{32}}}{\partial \rho_k} = -\sin\rho_k \sin\phi_k \tag{93}$$

$$\frac{\partial T_{s_{33}}}{\partial \rho_k} = -\sin\rho_k \cos\phi_k \tag{94}$$

$$\frac{\partial T_{s_{11}}}{\partial \phi_k} = 0 \tag{95}$$

$$\frac{\partial T_{s_{12}}}{\partial \phi_k} = \cos\psi_k \sin\rho_k \cos\phi_k + \sin\psi_k \sin\phi_k \tag{96}$$

$$\frac{\partial T_{s_{13}}}{\partial \phi_k} = -\cos\psi_k \sin\rho_k \sin\phi_k + \sin\psi_k \cos\phi_k \tag{97}$$

$$\frac{\partial T_{s_{21}}}{\partial \phi_k} = 0 \tag{98}$$

$$\frac{\partial T_{s_{22}}}{\partial \phi_k} = \sin\psi_k \sin\rho_k \cos\phi_k - \cos\psi_k \sin\phi_k \tag{99}$$

$$\frac{\partial T_{s_{23}}}{\partial \phi_k} = -\sin\psi_k \sin\rho_k \sin\phi_k - \cos\psi_k \cos\phi_k \tag{100}$$

$$\frac{\partial T_{s_{31}}}{\partial \phi_k} = 0 \tag{101}$$

$$\frac{\partial T_{s_{32}}}{\partial \phi_k} = \cos\psi_k \cos\phi_k \tag{102}$$

$$\frac{\partial T_{s_{33}}}{\partial \phi_k} = -\cos\rho_k \sin\phi_k \tag{103}$$

REFERENCES

[1] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software.
J. Wiley and Sons, 2001.

[2] Y. Bar-Shalom, P. K. Willett, and X. Tian
Tracking and Data Fusion.
YBS Publishing, 2011.

[3] D. Belfadel, R. W. Osborne, and Y. Bar-Shalom
"A Minimalist Approach to Bias Estimation for Passive Sensor Measurements with Targets of Opportunity,"
in *Proc. SPIE Conf. Signal and Data Processing of Small Targets*, #8857-13, San Diego, California, Aug. 2013.

[4] D. Belfadel, R. W. Osborne, and Y. Bar-Shalom
"Bias Estimation for Optical Sensor Measurements with Targets of Opportunity,"
in *Proc. 16th International Conference on Information Fusion*, Istanbul, Turkey, Jul. 2013.

[5] X. Benlian and W. Zhiquan
"Biased Bearings-Only Parameter Estimation for Bistatic System,"
Journal of Electronics (China), vol. 24, no. 3, May 2007.

[6] S. V. Bordonaro, P. Willett, and Y. Bar-Shalom
"Tracking with Converted Position and Doppler Measurements,"
in *Proc. SPIE Conf. Signal and Data Processing of Small Targets*, #8137-12, San Diego, CA, Aug. 2011.

[7] S. V. Bordonaro, P. Willett, and Y. Bar-Shalom
"Consistent Linear Tracker with Position and Range Rate Measurements,"
in *Proc. Asilomar Conf.*, Asilomar, CA, Nov. 2012.

[8] S. V. Bordonaro, P. Willett, and Y. Bar-Shalom
"Unbiased Tracking with Converted Measurements,"
in *Proc. 2012 IEEE Radar Conf.*, Atlanta, GA, May 2012.

[9] S. V. Bordonaro, P. Willett, and Y. Bar-Shalom
"Unbiased Tracking with Converted Measurements,"
in *Proc. 51st IEEE Conf. on Decision and Control*, Maui, HI, Dec. 2012.

[10] D. F. Crouse, R. Osborne, III, K. Pattipati, P. Willett, and Y. Bar-Shalom
"2D Location Estimation of Angle-Only Sensor Arrays Using Targets of Opportunity,"
in *Proc. 13th International Conference on Information Fusion*, Edinburgh, Scotland, Jul. 2010.

[11] B. D. Kragel, S. Danford, S. M. Herman, and A. B. Poore
"Bias Estimation Using Targets of Opportunity,"
Proc. SPIE Conf. on Signal and Data Processing of Small Targets, #6699, Aug. 2007.

[12] B. D. Kragel, S. Danford, S. M. Herman, and A. B. Poore
"Joint MAP Bias Estimation and Data Association: Algorithms,"
Proc. SPIE Conf. on Signal and Data Processing of Small Targets, #6699-1E, Aug. 2007.

[13] B. D. Kragel, S. Danford, and A. B. Poore
"Concurrent MAP Data Association and Absolute Bias Estimation with an Arbitrary Number of Sensors,"
Proc. SPIE Conf. on Signal and Data Processing of Small Targets, #6969-50, May 2008.

[14] X. Lin, F. Y. Bar-Shalom, and T. Kirubarajan
"Exact Multisensor Dynamic Bias Estimation with Local Tracks,"
IEEE Trans. on Aerospace and Electronic Systems, vol. 40, no. 2, pp. 576–590, Apr. 2004.

[15] R. W. Osborne, III, and Y. Bar-Shalom
"Statistical Efficiency of Composite Position Measurements from Passive Sensors,"
in *Proc. SPIE Conf. on Signal Proc., Sensor Fusion, and Target Recognition*, #8050-07, Orlando, FL, Apr. 2011.

[16] R. W. Osborne, III, and Y. Bar-Shalom
"Statistical Efficiency of Composite Position Measurements from Passive Sensors,"
IEEE Trans. on Aerospace and Electronic Systems, vol. 49, no. 4, Oct. 2013.

Djedjiga Belfadel is a Ph.D. candidate in the Electrical Engineering department at the University of Connecticut, Storrs, CT. Her research interests include target tracking, data association, sensor fusion, sensor biases, machine vision, and other aspects of estimation. She obtained her B.S., degrees from the University of Mouloud Mammeri in 2003, and her M.S., degrees from the University of New Haven in 2008, both in electrical engineering. Before joining the Estimation and Signal Processing (ESP) Laboratory, she worked, as an Electrical Engineer, from 2009 to 2011, at Evax Systems Inc. in Branford, Connecticut.

Richard W. Osborne, III obtained his B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Connecticut in 2004, 2007, and 2012, respectively. He is currently an Assistant Research Professor in the Electrical Engineering department at the University of Connecticut, Storrs, CT. His academic interests include adaptive target tracking, information/sensor fusion, machine vision, and other aspects of estimation.

Yaakov Bar-Shalom was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion, Israel Institute of Technology, in 1963 and 1967 and the Ph.D. degree from Princeton University in 1970, all in electrical engineering. From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, California. Currently he is Board of Trustees Distinguished Professor in the Dept. of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering at the University of Connecticut. He is also Director of the ESP (Estimation and Signal Processing) Lab. His current research interests are in estimation theory, target tracking and data fusion. He has published over 500 papers and book chapters in these areas and in stochastic adaptive control. He coauthored the monograph Tracking and Data Association (Academic Press, 1988), the graduate texts Estimation and Tracking: Principles, Techniques and Software (Artech House, 1993), Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction (Wiley, 2001), the advanced graduate texts Multitarget-Multisensor Tracking: Principles and Techniques (YBS Publishing, 1995), Tracking and Data Fusion (YBS Publishing, 2011), and edited the books Multitarget-Multisensor Tracking: Applications and Advances (Artech House, Vol. I, 1990; Vol. II, 1992; Vol. III, 2000). He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multi-target tracking." He has been consulting to numerous companies and government agencies, and originated the series of Multitarget-Multisensor Tracking short courses offered via UCLA Extension, at Government Laboratories, private companies and overseas. During 1976 and 1977 he served as Associate Editor of the IEEE Transactions on Automatic Control and from 1978 to 1981 as Associate Editor of Automatica. He was Program Chairman of the 1982 American Control Conference, General Chairman of the 1985 ACC, and Co-Chairman of the 1989 IEEE International Conference on Control and Applications. During 1983–87 he served as Chairman of the Conference Activities Board of the IEEE Control Systems Society and during 1987–89 was a member of the Board of Governors of the IEEE CSS. He was a member of the Board of Directors of the International Society of Information Fusion (1999–2004) and served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002 and Vice President for Publications in 2004–13. In 1987 he received the IEEE CSS Distinguished Member Award. Since 1995 he is a Distinguished Lecturer of the IEEE AESS and has given numerous keynote addresses at major national and international conferences. He is co-recipient of the M. Barry Carlton Award for the best paper in the IEEE Transactions on Aerospace and Electronic Systems in 1995 and 2000 and recipient of the 1998 University of Connecticut AAUP Excellence Award for Research. In 2002 he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. In 2008 he was awarded the IEEE Dennis J. Picard Medal for Radar Technologies and Applications, and in 2012 the Connecticut Medal of Technology. He has been listed by academic.research.microsoft (top authors in engineering) as #1 among the researchers in Aerospace Engineering based on the citations of his work.

Statistical Efficiency of Target Localization from Angle and Shockwave Measurements

RICHARD W. OSBORNE, III
YAAKOV BAR-SHALOM
JEMIN GEORGE
LANCE KAPLAN

This work derives the Cramer-Rao lower bound (CRLB) for an acoustic target and sensor localization system in which the noise characteristics may depend on the location of the source. The system itself has been previously examined, but without deriving the CRLB and showing the statistical efficiency of the estimator used. Three different versions of the CRLB are derived, one in which direction of arrival (DOA) and (shockwave based) range measurements are available ("local estimate" based CRLB), one in which two types of DOA measurements and the time difference between them is available ("native measurement" based CRLB), and one in which only DOA measurements (bearing) are available ("bearings-only" CRLB). In each case, the estimator is found to be statistically efficient; but, depending on the sensor-target geometry, the range measurements may or may not significantly contribute to the accuracy of target localization. Additionally, the native measurements are found to result in superior localization when compared to the use of the range estimates.

Manuscript received May 30, 2013; revised January 14, 2014; released for publication October 14, 2014.

Refereeing of this contribution was handled by Ramona Georgescu.

This work was supported by the Army Research Office under contract W911NF-06-1-0467.

There are no conflict-of-interest or financial disclosure statements to be made at this time.

Authors' addresses: R. Osborne and Y. Bar-Shalom, Department of Electrical and Computer Engineering, University of Connecticut, 371 Fairfield Way, U-2157, Storrs, CT 06269 (e-mail: {rosborne, ybs}@engr.uconn.edu); J. George and L. Kaplan, Sensors and Electron Devices Directorate, U.S. Army Research Laboratory, 2800 Powder Mill Rd., Adelphi, MD 20783, (e-mail: {jemin.george, lance.m.kaplan}@us.army.mil)

1. INTRODUCTION

In any estimation system the ultimate goal is to extract the maximum information from the available data. The Fisher information matrix (FIM) provides a measure of the total information available from the observations of the system, and its inverse provides the Cramer-Rao lower bound (CRLB) [2]. A statistically efficient estimator is one in which the (co)variance of the estimation error meets the CRLB, and, therefore, extracts all of the available information from the observations.

The CRLB and statistical efficiency of an acoustic localization system will be examined here, based on the system described in [9], [10], which is meant to estimate the location of the source of a detected gunshot. Each sensor node of the system is assumed to provide an estimated bearing (direction of arrival—DOA) to the target, and, if the sensor node lies within a certain field of view (FOV) around the path of the bullet, a range estimate and bullet trajectory estimate as well. The range and bullet trajectory estimates are provided via a nonlinear transformation of the "native" measurements consisting of the bearing, a shockwave DOA measurement, and a time difference of arrival (TDOA) between the two DOA measurements. For those sensors that provide estimated range, the noise variance will be highly dependent on the position of the source. Each sensor node's local estimates (or, alternatively, its native measurements) are passed to a fusion center to perform the overall estimation of the target position. The sensor locations can also be simultaneously estimated with the target (source) location, but the improvement is negligible. It is also possible to remove the sensor locations from the estimation performed at the fusion center, but the inaccuracy of the sensor locations must then be explicitly accounted for in the CRLB derivations.

A number of papers have examined the problem of target localization in passive sensor environments, including [3], [7], [8], [18], [22], [24], [25]. The work of [24] generalizes the results of [7] to include sensor position uncertainty; however, neither paper examines the CRLB to see whether the estimator is statistically efficient. In [3], [8], [18], [25], different applications of localization with passive sensors are studied that also consider the CRLB. However, in [25], no estimation scheme is shown to meet the CRLB. In [8] the maximum likelihood (ML) estimation scheme examined is shown to be statistically efficient only when a significant number of measurements are used. In none of the above-mentioned papers were cases of position-dependent measurement noise considered.

Specific research pertaining to the acoustic localization of small-arms fire is examined in [1], [5], [9]–[17], [20], [21], [23]. In most of these works, CRLB/efficiency analysis is not performed. References

[11], [23] use the "local estimates," while the remainder use either "native measurements" or only time-of-arrival (TOA) or TDOA measurements. Additionally, [13], [14], [16], [17] employ a bullet deceleration model. The work of [1] examined the effect of assuming an (incorrect) constant velocity bullet model and demonstrated modest localization errors for realistic scenarios. Of the previously mentioned work, only [5], [12], [16] perform any analysis of the CRLB. In [12], the CRLB is examined, but only in the case of native measurements. Reference [5] examines a method of localization using only TDOA measurements and derives the CRLB; however, the CRLB is shown to provide only a loose bound, with inaccurate cross-range performance prediction. Reference [16] also used TDOA measurements without DOA and derived the CRLB, but did not present results statistically demonstrating efficiency, though it was mentioned that simulations indicated their estimator met the bound for TDOA accuracy below a given threshold.

In this work, the CRLB of the central estimator (fuser) is derived for three cases: a "bearings-only" case, which assumes that only bearing measurements are available from the sensor nodes; a "local estimate" case, which assumes that range and bullet trajectory estimates are available in addition to bearing; and a "native measurement" case, which assumes that the two previously mentioned types of DOA measurements (to the shooter and the shockwave) and the TDOA are available.

Section 2 provides an overview of the system in question, and examines the probability distribution and density of the range estimate errors from the individual sensors. Section 3 provides the expressions necessary to evaluate the CRLB for the problem in question, both with and without the position-dependent noise terms. Section 4 describes the simulation scenarios and provides the results. Finally, Section 5 concludes the paper.

2. LOCALIZATION SYSTEM OVERVIEW

The system to be examined here is the same as the one described in [9], [10] except that we also consider the use of native measurements. A brief overview of the system is provided here, however, to introduce the concepts and notations.

A number of acoustic sensors are placed throughout a surveillance region with the intent of detecting gunfire and estimating the position of the source. The target (source) location will be denoted as

$$T = [T_x \quad T_y]' \tag{1}$$

and the ith sensor location is denoted as

$$S_i = [S_{i_x} \quad S_{i_y}]' \tag{2}$$

The problem is assumed constrained to a two-dimensional plane for simplicity.

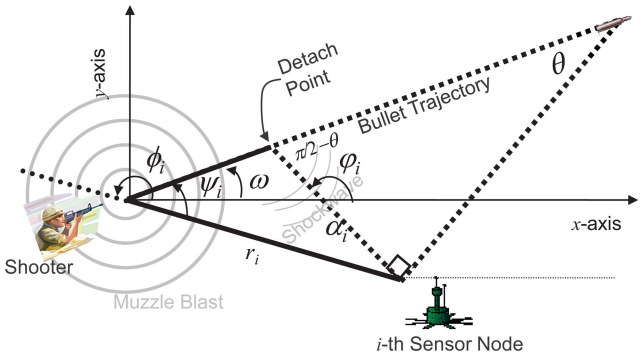


Fig. 1. Geometry of the bullet trajectory and the DOA angles of the muzzle blast and shockwave as seen by the ith sensor node.

2.1. Sensor Nodes

Each sensor will be assumed to provide at most five *native measurements*

$$\zeta_i = [\hat{\phi}_i \quad \hat{\varphi}_i \quad \hat{\tau}_i]' \tag{3}$$

and

$$\hat{S}_i = [\hat{S}_{i_x} \quad \hat{S}_{i_y}]' \tag{4}$$

where $\hat{\phi}_i$ is the DOA angle to the shooter, based on the detection of the muzzle blast; $\hat{\varphi}_i$ is the DOA angle of the shockwave from the bullet; $\hat{\tau}_i$ is the TDOA between the muzzle blast and the shockwave; and $\hat{S}_i$ is the noisy sensor location (obtained via a GPS sensor at each node).[1] The DOA measurements are assumed to be measured counter-clockwise (CCW) from the x-axis of a global reference coordinate system, to which each sensor is assumed to be aligned. Each measurement is assumed to be corrupted by zero-mean Gaussian noise, with standard deviations of σ_ϕ, σ_φ, σ_τ, σ_{i_x} and σ_{i_y}, respectively. The overall geometry of the various angular measurements involved are depicted in Figure 1.

The shockwave (and therefore the TDOA measurement) is only visible to sensor nodes that are within a limited FOV around the path of the bullet, depicted in Figure 2. The FOV is $\pi - 2\theta$ [6], where

$$\theta = \sin^{-1}\left(\frac{1}{m}\right) \tag{5}$$

and m is the Mach number of the bullet, assumed here to be $m = 2$ [12]. Note that in this work, a constant velocity bullet model is considered.

The target bearing from the ith sensor node is

$$\phi_i = \tan^{-1}\left(\frac{T_y - S_{i_y}}{T_x - S_{i_x}}\right) \tag{6}$$

and the DOA angle of the shockwave is

$$\varphi_i = \begin{cases} -\dfrac{\pi}{2} - \theta + \omega & \text{if } \pi + \omega < \phi_i < \dfrac{3\pi}{2} - \theta + \omega \\ \dfrac{\pi}{2} + \theta + \omega & \text{if } \dfrac{\pi}{2} + \theta + \omega < \phi_i < \pi + \omega \end{cases} \tag{7}$$

[1] We use similar notation and terminology as in [9], [10].

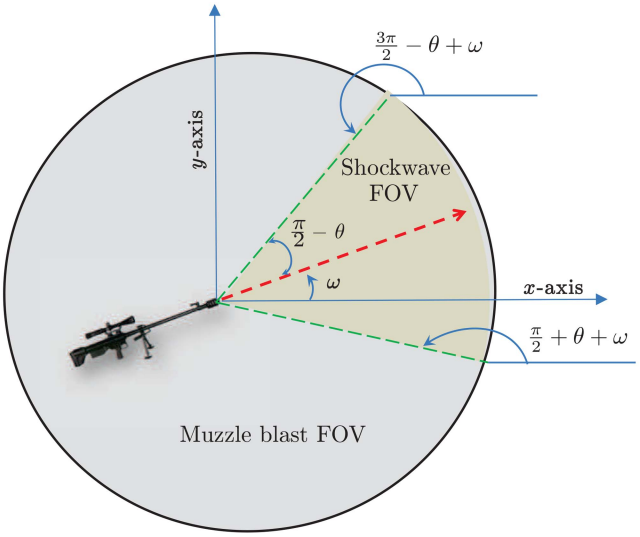


Fig. 2. Field of view (FOV) of the muzzle blast and shockwave DOA angles.

where ω is the angle of the trajectory (counter-clockwise) with respect to the x-axis.

Using the *native measurements*, the sensors can compute the *local estimates*[2]

$$z_i = [\hat{\phi}_i \quad \hat{r}_i \quad \hat{\omega}_i]' \tag{8}$$

where [9], [10]

$$\hat{r}_i = \frac{c\hat{\tau}_i}{1 - \cos(\hat{\phi}_i - \hat{\varphi}_i)} \tag{9}$$

is the estimated target range, c is the speed of sound (assumed to be known perfectly), and $\hat{\omega}_i$ is the estimated bullet trajectory angle. In view of (7), the estimated bullet trajectory can be obtained directly from $\hat{\varphi}_i$ and the standard deviation of $\hat{\omega}_i$ is σ_φ.

Sensor i sends its measured location (4), and either the native measurements (3) or the local estimates (8) to a fusion center, to estimate the source location T.

The variance of the range estimate (which is location dependent) can be approximated as (similar to [9], [10])

$$\sigma^2_{r_i}(T, \hat{S}_i, \omega) \approx \nabla r_i \begin{bmatrix} \sigma^2_\tau & 0 \\ 0 & \sigma^2_\phi + \sigma^2_\varphi \end{bmatrix} \nabla r_i' \tag{10}$$

where

$$\nabla r_i = \begin{bmatrix} \frac{\partial r_i}{\partial \tau} & \frac{\partial r_i}{\partial(\phi - \varphi)} \end{bmatrix} \tag{11}$$

and

$$\frac{\partial r_i}{\partial \tau} = \frac{c}{1 - \cos(\phi_i - \varphi_i)} \tag{12}$$

$$\frac{\partial r_i}{\partial(\phi - \varphi)} = -\frac{r_i \sin(\phi_i - \varphi_i)}{1 - \cos(\phi_i - \varphi_i)} \tag{13}$$

[2]The angular measurements of (8) are also assumed to be measured CCW from the x-axis of the global reference frame.

The likelihood function of T, S_i, and ω given the estimate z_i is

$$\Lambda_{z_i}(T, S_i, \omega) \triangleq p(z_i \mid T, S_i, \omega) \approx \mathcal{N}(z_i; \boldsymbol{\mu}_{z_i}, \Sigma_{z_i}) \tag{14}$$

where

$$\boldsymbol{\mu}_{z_i} = [\phi_i \quad r_i \quad \omega]' \tag{15}$$

and, since the noise on ω is the same as the noise on φ in view of (7), and is the same for all i,

$$\Sigma_{z_i}(T, \hat{S}_i, \omega) = \begin{bmatrix} \sigma^2_\phi & \mathrm{cov}(r_i, \phi_i) & 0 \\ \mathrm{cov}(r_i, \phi_i) & \sigma^2_{r_i}(T, \hat{S}_i, \omega) & \mathrm{cov}(r_i, \omega) \\ 0 & \mathrm{cov}(r_i, \omega) & \sigma^2_\varphi \end{bmatrix} \tag{16}$$

with (see Appendix A)

$$\mathrm{cov}(r_i, \phi_i) = -\frac{r_i \sin(\phi_i - \varphi_i)}{1 - \cos(\phi_i - \varphi_i)} \sigma^2_{\phi_i} \tag{17}$$

$$\mathrm{cov}(r_i, \omega_i) = \frac{r_i \sin(\phi_i - \varphi_i)}{1 - \cos(\phi_i - \varphi_i)} \sigma^2_{\varphi_i} \tag{18}$$

The errors in (14) are assumed to be uncorrelated across the sensors.

The likelihood function of T, S_i, and ω given the native measurements ζ_i is

$$\Lambda_{\zeta_i}(T, S_i, \omega) \triangleq p(\zeta_i \mid T, S_i, \omega) = \mathcal{N}(\zeta_i; \boldsymbol{\mu}_{\zeta_i}, \Sigma_{\zeta_i}) \tag{19}$$

where

$$\boldsymbol{\mu}_{\zeta_i} = [\phi_i \quad \varphi_i \quad \tau_i]' \tag{20}$$

and

$$\Sigma_{\zeta_i} = \begin{bmatrix} \sigma^2_\phi & 0 & 0 \\ 0 & \sigma^2_\varphi & 0 \\ 0 & 0 & \sigma^2_\tau \end{bmatrix} \tag{21}$$

The likelihood function of S_i is

$$\Lambda_{S_i}(S_i) \triangleq p(\hat{S}_i \mid S_i) = \mathcal{N}(\hat{S}_i; S_i, \Sigma_{S_i}) \tag{22}$$

where

$$\Sigma_{S_i} = \begin{bmatrix} \sigma^2_{i_x} & 0 \\ 0 & \sigma^2_{i_y} \end{bmatrix} \tag{23}$$

The sensors are assumed to obtain their locations, albeit imperfectly, from GPS.[3] Additionally, the sensors are assumed to be aligned to a common reference frame (e.g., via compass readings, where any error/bias present is assumed to be identical across sensors due to the small area involved). For the sensors, the GPS localization serves as a prior and guarantees complete observability for the target-sensor complex. The final estimates of the sensor locations can be only slightly improved over their initial GPS estimates, but the improvement this makes to the final target localization is

[3]If the sensor position estimates contain a common (slowly varying) bias across sensors (a reasonable assumption since these sensors are not too far from each other), the relative sensor registration will be unaffected and the target estimate will exhibit the same bias.

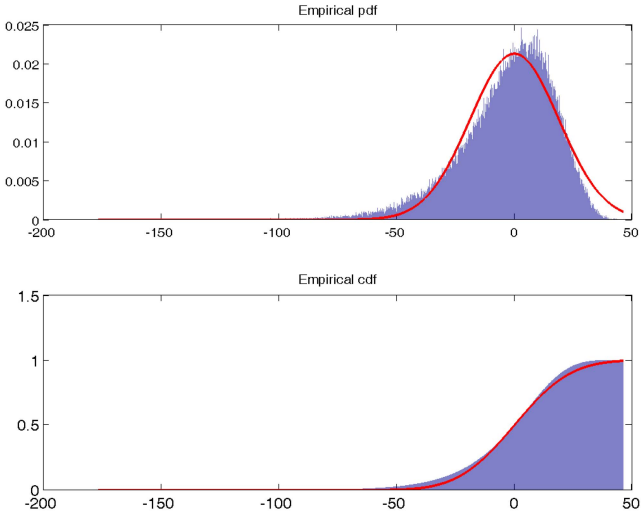


Fig. 3. Empirical pdf and cdf of $\tilde{r}_i$ for Sensor 1 of Scenario 3.

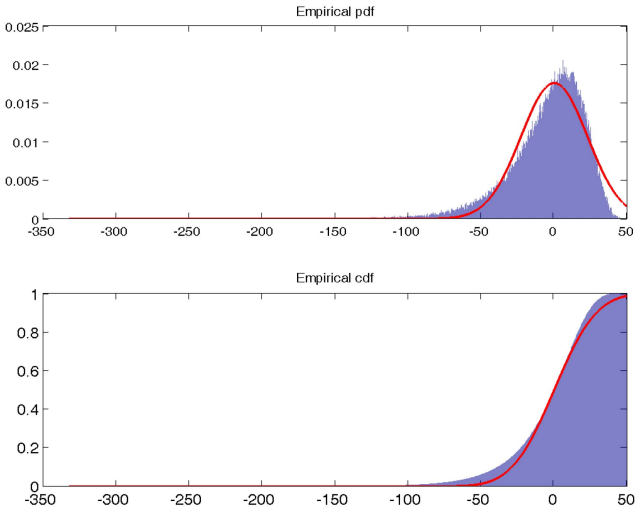


Fig. 4. Empirical pdf and cdf of $\tilde{r}_i$ for Sensor 2 of Scenario 3.

negligible. For that reason, the simultaneous estimation of the sensor positions can reasonably be omitted from the overall estimation, but the effect of the sensor localization errors needs to be accounted for in the variances of ϕ_i and r_i, particularly when calculating the CRLB for use in determining the overall efficiency of the estimation scheme.

2.2. Range Estimate Error Distribution

The preceding section followed the analysis of [9], [10] with regards to the range estimation error

$$\tilde{r}_i = \hat{r}_i - r_i \tag{24}$$

where the range estimation error was assumed to be Gaussian distributed, i.e.,

$$\tilde{r}_i \sim \mathcal{N}(r_i, \sigma^2_{r_i}(T, S_i, \omega)) \tag{25}$$

where the variance is given by (10), or equivalently as

$$\sigma^2_{r_i}(T, S_i, \omega) = \frac{c^2\sigma^2_\tau + r_i^2(\sigma^2_\phi + \sigma^2_\varphi)\sin^2(\phi_i - \varphi_i)}{(1 - \cos(\phi_i - \varphi_i))^2} \tag{26}$$

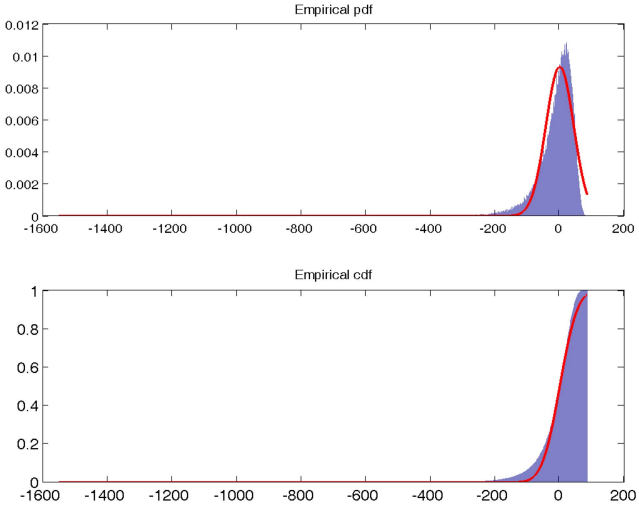


Fig. 5. Empirical pdf and cdf of $\tilde{r}_i$ for Sensor 3 of Scenario 3.

In order to examine the Gaussian assumption on the range estimate errors, an empirical pdf and cdf (i.e., histograms) of the range estimate error was generated from 10^5 Monte Carlo simulations. The range estimates used to generate the empirical pdfs and cdfs were generated by (9) using the native measurements corrupted by Gaussian noise. The nonlinearity of (9) is what causes the range errors to be non-Gaussian.

The simulation parameters are set identically to Scenario 3 of Section 4. The empirical pdfs and cdfs are shown in Figures 3–5. Each figure, in addition to the histograms, is overlaid with the Gaussian pdf of (25). It is clear from these figures that the actual density of the range errors is not symmetric, and there is a heavier left tail than if the errors were indeed Gaussian.

2.3. Centralized Fusion

The estimates z_i and $\hat{S}_i$ (or ζ_i and $\hat{S}_i$) from each sensor are passed on to a fusion center in order to determine the estimate $\hat{\mathbf{x}}$ by means of the Iterated Least Squares (ILS) estimator[4] [2]. The parameter vector to be estimated is

$$\mathbf{x} = [T_x \quad T_y \quad \omega \quad S_{1_x} \quad S_{1_y} \cdots S_{n_x} \quad S_{n_y}]' \tag{27}$$

with observations

$$\mathbf{y} = [\mathbf{y}_1' \cdots \mathbf{y}_n']' \tag{28}$$

where

$$\mathbf{y}_i = [z_i' \quad \hat{S}_i']' \tag{29}$$

or with observations

$$\boldsymbol{\eta} = [\boldsymbol{\eta}_1' \cdots \boldsymbol{\eta}_n']' \tag{30}$$

where

$$\boldsymbol{\eta}_i = [\zeta_i' \quad \hat{S}_i']' \tag{31}$$

depending on whether the native measurements (31) or local estimates (29) are sent to the fusion center.

[4] Alternatively, Levenberg-Marquardt, or any other suitable nonlinear least squares solver may be used.

In order to use the ILS estimation algorithm, an initial estimate of $\mathbf{x}$ is needed. It has been noted [9], [10] that the ILS estimator is sensitive to the initial estimate and may diverge if the initial estimate is too far from the truth.

While the initialization of the target position could be performed by using the bearing and range measurements (from the nodes with range measurements), the large variance of the range measurements was found to occasionally cause divergence in the ILS algorithm. A more robust initialization was found to follow a similar method to that used in [4]. This method of initialization utilizes only the available bearing measurements from each sensor (6), which can be rewritten as

$$\underbrace{\begin{bmatrix} \tan\phi_1 & -1 \\ \tan\phi_2 & -1 \\ \vdots & \vdots \\ \tan\phi_n & -1 \end{bmatrix}}_{A} T = \underbrace{\begin{bmatrix} S_{1_x}\tan\phi_1 - S_{1_y} \\ S_{2_x}\tan\phi_2 - S_{2_y} \\ \vdots \\ S_{n_x}\tan\phi_n - S_{n_y} \end{bmatrix}}_{b} \tag{32}$$

and T is obtained as

$$T = A^{\dagger} b \tag{33}$$

where $A^{\dagger}$ is the (right) pseudo-inverse of A.

Also, note that (32) can be rewritten using the expression

$$\phi_i = \cot^{-1}\left(\frac{T_x - S_{i_x}}{T_y - S_{i_y}}\right) \tag{34}$$

which is simply (6) rewritten using the cotangent function. As suggested in [4], use of the cotangent function has been made when the measured bearing is between 45° and 135° or between -45° and -135°, in order to avoid the singularity of the tangent function around $\pm 90^\circ$.

To complete the initialization of $\mathbf{x}$, ω can be taken as the average of $\hat{\omega}_i$. If the native measurements are sent to the fusion center, the DOA shockwave estimates $\hat{\varphi}_i$ can be used to solve for the equivalent $\hat{\omega}_i$ in order to initialize in the same manner.

Due to the nature of the DOA shockwave estimates (7) (which is one of two angles depending on which side of the bullet trajectory the sensor is located on), the predicted values of φ_i that must be calculated for the ILS algorithm may occasionally exhibit very large errors. This will occur if the errors on $\hat{\phi}_i$ and $\hat{\omega}_i$ are such that the ith sensor is predicted to appear on the incorrect side of the bullet trajectory. For this reason, when the native measurements are sent to the fusion center, the predicted value of φ_i will be given as whichever of the two possibilities is closest to the value of $\hat{\varphi}_i$ sent to the fusion center. This is a reasonable solution to resolving the ambiguity since the errors of $\hat{\varphi}_i$ will be assumed to be much smaller than the difference between the two values of φ_i.

3. CRAMER-RAO LOWER BOUND

The CRLB provides a lower bound on the covariance matrix of the estimate $\hat{\mathbf{x}}$ as

$$E[(\hat{\mathbf{x}} - \mathbf{x})(\hat{\mathbf{x}} - \mathbf{x})'] \geq J^{-1} \tag{35}$$

where J is the FIM

$$J = E\{[\nabla_{\mathbf{x}}\lambda(\mathbf{x})][\nabla_{\mathbf{x}}\lambda(\mathbf{x})]'\} \tag{36}$$

and $\lambda(\mathbf{x})$ is the negative log-likelihood function (NLLF).

3.1. Native Measurement Based FIM

The likelihood function of $\mathbf{x}$ based on $\boldsymbol{\eta}$, assuming the sensor location estimate errors are independent of the native measurement errors, is

$$\Lambda_{\boldsymbol{\eta}}(\mathbf{x}) = \prod_{i=1}^{n} \Lambda_{\zeta_i}(T, S_i, \omega)\Lambda_{S_i}(S_i) \tag{37}$$

The NLLF corresponding to (37) is

$$\lambda_{\boldsymbol{\eta}}(\mathbf{x}) = \tfrac{1}{2}\sum_{i=1}^{n} (\boldsymbol{\eta}_i - \mu_{\boldsymbol{\eta}_i}(\mathbf{x}))'\Sigma_{\boldsymbol{\eta}_i}^{-1}(\boldsymbol{\eta}_i - \mu_{\boldsymbol{\eta}_i}(\mathbf{x})) \tag{38}$$

where

$$\mu_{\boldsymbol{\eta}_i} = [\mu_{\zeta_i}' \quad S_i']' \tag{39}$$

$$\Sigma_{\boldsymbol{\eta}_i} = \begin{bmatrix} \Sigma_{\zeta_i} & 0 \\ 0 & \Sigma_{S_i} \end{bmatrix} \tag{40}$$

and the unnecessary constant terms have been omitted. In this case, the FIM can be shown to be

$$J_{\boldsymbol{\eta}} = H_{\boldsymbol{\eta}}'\Sigma_{\boldsymbol{\eta}}^{-1}H_{\boldsymbol{\eta}} \tag{41}$$

where

$$H_{\boldsymbol{\eta}} \triangleq \frac{\partial \mu_{\boldsymbol{\eta}}(\mathbf{x})}{\partial \mathbf{x}} \tag{42}$$

is the Jacobian matrix of the native measurements,

$$\mu_{\boldsymbol{\eta}}(\mathbf{x}) = [\mu_{\boldsymbol{\eta}_1}(\mathbf{x})', \ldots, \mu_{\boldsymbol{\eta}_n}(\mathbf{x})']' \tag{43}$$

and

$$\Sigma_{\boldsymbol{\eta}} = \begin{bmatrix} \Sigma_{\boldsymbol{\eta}_1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Sigma_{\boldsymbol{\eta}_n} \end{bmatrix} \tag{44}$$

3.2. Local Estimate Based FIM

The likelihood function of $\mathbf{x}$ based on $\mathbf{y}$ follows similarly to (37). The corresponding NLLF is

$$\lambda_{\mathbf{y}}(\mathbf{x}) = \sum_{i=1}^{n} [\tfrac{1}{2}(\mathbf{y}_i - \mu_{\mathbf{y}_i}(\mathbf{x}))'\Sigma_{\mathbf{y}_i}(\mathbf{x})^{-1}(\mathbf{y}_i - \mu_{\mathbf{y}_i}(\mathbf{x})) + \tfrac{1}{2}\ln|\Sigma_{z_i}(\mathbf{x})|] \tag{45}$$

where

$$\mu_{\mathbf{y}_i} = [\mu_{z_i}' \quad S_i']' \tag{46}$$

$$\Sigma_{\mathbf{y}_i}(\mathbf{x}) = \begin{bmatrix} \Sigma_{z_i}(\mathbf{x}) & 0 \\ 0 & \Sigma_{S_i} \end{bmatrix} \tag{47}$$

and the unnecessary constant terms have been omitted. Note that some entries of Σ_{z_i} are dependent on the target-sensor geometry. In this case, the FIM will be more complicated.

The FIM for the case of a multivariate Gaussian likelihood with parameter-dependent covariance is as follows [19]. The gradient terms of the FIM are

$$\begin{aligned}\frac{\partial \lambda_{\mathbf{y}}(\mathbf{x})}{\partial x_j} &= \frac{1}{2}\mathrm{tr}\left(\Sigma_{\mathbf{y}}^{-1}(\mathbf{x})\frac{\partial \Sigma_{\mathbf{y}}(\mathbf{x})}{\partial x_j}\right) \\ &\quad - \frac{1}{2}(\mathbf{y}-\mu_{\mathbf{y}}(\mathbf{x}))'\Sigma_{\mathbf{y}}^{-1}(\mathbf{x})\frac{\partial \Sigma_{\mathbf{y}}(\mathbf{x})}{\partial x_j}\Sigma_{\mathbf{y}}^{-1}(\mathbf{x})(\mathbf{y}-\mu_{\mathbf{y}}(\mathbf{x}))' \\ &\quad - \left[\frac{\partial \mu_{\mathbf{y}}(\mathbf{x})}{\partial x_j}\right]'\Sigma_{\mathbf{y}}^{-1}(\mathbf{x})(\mathbf{y}-\mu_{\mathbf{y}}(\mathbf{x}))\end{aligned}$$

where x_j is the jth entry of $\mathbf{x}$. The (i,j)th entry in the FIM $J_{\mathbf{y}}$ is then

$$\begin{aligned}J_{i,j} &= \frac{1}{2}\mathrm{tr}\left(\Sigma_{\mathbf{y}}^{-1}(\mathbf{x})\frac{\partial \Sigma_{\mathbf{y}}(\mathbf{x})}{\partial x_j}\Sigma_{\mathbf{y}}^{-1}(\mathbf{x})\frac{\partial \Sigma_{\mathbf{y}}(\mathbf{x})}{\partial x_\ell}\right) \\ &\quad + \left[\frac{\partial \mu_{\mathbf{y}}(\mathbf{x})}{\partial x_j}\right]'\Sigma_{\mathbf{y}}^{-1}(\mathbf{x})\left[\frac{\partial \mu_{\mathbf{y}}(\mathbf{x})}{\partial x_\ell}\right] \qquad (48)\end{aligned}$$

In order to compare bearings-only localization to the localization schemes presented here, the "bearings-only" FIM, J_b, must also be derived. It can be shown that J_b follows identically to (41), but with (42) and (44) modified to remove the portions dealing with φ_i and τ_i.

In all, three versions of the CRLB are evaluated in the sequel: the bearings-only CRLB, J_b^{-1}, the local estimate based CRLB, $J_{\mathbf{y}}^{-1}$, and the native measurement CRLB, $J_{\boldsymbol{\eta}}^{-1}$.

For the case of the local estimate based FIM $J_{\mathbf{y}}$, appropriate care should be taken to adjust $\mu_{\mathbf{y}}(\mathbf{x})$ and $\Sigma_{\mathbf{y}}(\mathbf{x})$ for sensors that do not provide $\hat{r}_i$ and $\hat{\omega}_i$. Additionally, in order to calculate the FIM from (48), the partial derivatives of $\sigma_{r_i}^2$, $\mathrm{cov}(r_i,\phi_i)$, and $\mathrm{cov}(r_i,\omega_i)$ are needed, as well as the partial derivatives of ϕ_i, r_i and ω. The expressions for the necessary partial derivatives can be found in Appendix B.

4. SIMULATION RESULTS

The simulation scenarios examined here include the scenarios of [9], [10] and an additional modified scenario with fewer sensors. For each scenario, the Mach number of the bullet is assumed to be $m = 2$, and the speed of sound is assumed to be $c = 342$ m/s. The measurement noise standard deviations are $\sigma_\phi = \sigma_\varphi = 4°$, $\sigma_\tau = 1$ ms, and $\sigma_{i_x} = \sigma_{i_y} = 2$ m. The simulations were performed for 100 Monte Carlo runs for each scenario.

For each scenario, the fusion center estimates the vector $\mathbf{x}$ of (27) via the ILS algorithm, using each of the following sets of measurements:

(i) the bearings-only case, with $\hat{\phi}_i$ and $\hat{S}_i$
(ii) and the local estimate case, with z_i and $\hat{S}_i$
(iii) the native measurement case, with ζ_i and $\hat{S}_i$

When the ILS algorithm is performed at the fusion center with the local estimates $\hat{z}_i$, the measurement noise covariance was modified from that of (16). Namely, the crosscovariance between ϕ_i and r_i, and the crosscovariance between ω_i and r_i are assumed to be zero. Even though these terms were found to be reasonably good approximations to the true crosscovariance between the range and angular errors, the ILS algorithm performed poorly when provided a measurement noise covariance containing these terms (results demonstrating this can be found in Figure 21).

In Scenarios 1 and 2, there are five sensor nodes located at (all positions are in m)

$$S = \begin{bmatrix} 127 & 20 & 90 & 136 & 182 \\ 107 & 22 & 0 & 68 & 59 \end{bmatrix} \qquad (49)$$

In Scenario 1, the target is located at $T = [50,50]'$ and the bullet is fired at a trajectory of $\omega = 30°$ (counterclockwise from the x-axis). Due to the location of the sensors and the trajectory of the bullet, only sensors 1, 4 and 5 receive the shockwave and are able to send range and bullet trajectory estimates (or, equivalently, φ_i and τ_i in the native measurement case) to the fusion center.

The results of Scenario 1 are shown in Figures 6–8. Each figure shows the true locations of the target and sensors, along with the corresponding 95% error ellipses. Figure 6 shows the error ellipses corresponding to the sample covariance matrix (dashed line) calculated from the estimation errors over the 100 Monte Carlo runs when only bearing measurements are sent to the fusion center, and the covariance matrix from the bearings-only CRLB (solid line, denoted as $\mathrm{CRLB}_{\mathrm{bo}}$). Figure 7 shows the covariance matrix calculated from the estimation errors when the native measurements of ϕ, φ, and τ are available at the fusion center, and the covariance matrix from the native measurement CRLB (denoted as $\mathrm{CRLB}_{\mathrm{nm}}$). Figure 8 shows the covariance matrix calculated from the estimation errors when local estimates of range and bullet trajectory are available at the fusion center (in addition to bearing), and the covariance matrix from the local estimate based CRLB (denoted $\mathrm{CRLB}_{\mathrm{le}}$).

The covariance matrices from the CRLBs closely match the covariances of the estimation errors calculated from the simulation. This first indicates that the ILS estimation carried out by the fusion center is statistically efficient. Additionally, the fact that the bearings-only $\mathrm{CRLB}_{\mathrm{bo}}$ and local estimate based $\mathrm{CRLB}_{\mathrm{le}}$ matrices closely match suggests that very little information is gained from the range estimates sent from sensors 1, 4 and 5. The native measurement $\mathrm{CRLB}_{\mathrm{nm}}$ matrix,

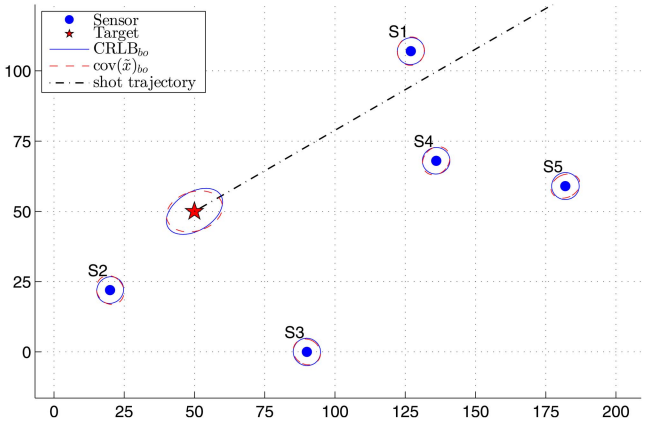


Fig. 6. Scenario 1, bearings-only CRLB$_{bo}$, ellipses and error ellipses of estimated target and sensor locations (all 95%).

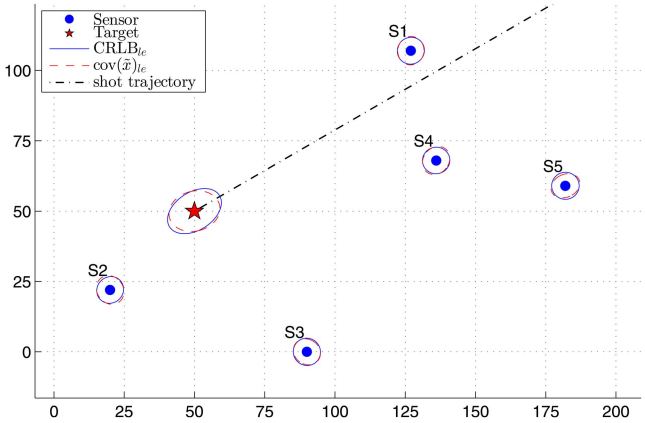


Fig. 7. Scenario 1, native measurement CRLB$_{nm}$ ellipses and error ellipses of estimated target and sensor locations (all 95%).

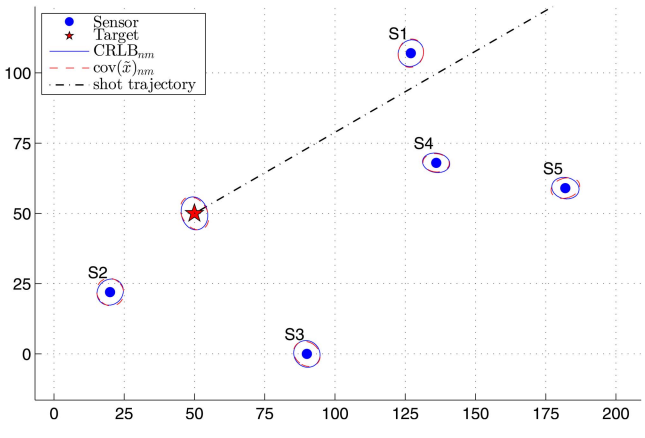


Fig. 8. Scenario 1, local estimate based CRLB$_{le}$ ellipses and error ellipses of estimated target and sensor locations (all 95%).

however, shows there is room for improvement of the target localization accuracy, which can be achieved by sending the native measurements to the fusion center.

In Scenario 2, the target is located at $T = [150, -50]'$ and the bullet is fired at a trajectory of $\omega = 170°$. Due to the location of the sensors and the trajectory of the bullet, only sensors 2 and 3 receive the shockwave and are able to send range and bullet trajectory estimates to the fusion center.

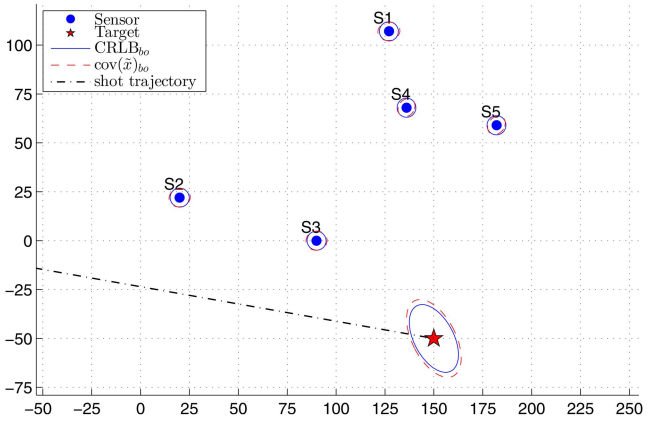


Fig. 9. Scenario 2, bearings-only CRLB$_{bo}$ ellipses and error ellipses of estimated target and sensor locations (all 95%).

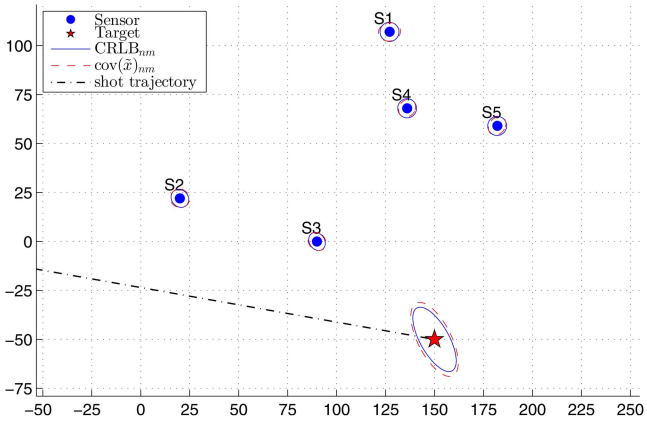


Fig. 10. Scenario 2, native measurement CRLB$_{nm}$ ellipses and error ellipses of estimated target and sensor locations (all 95%).

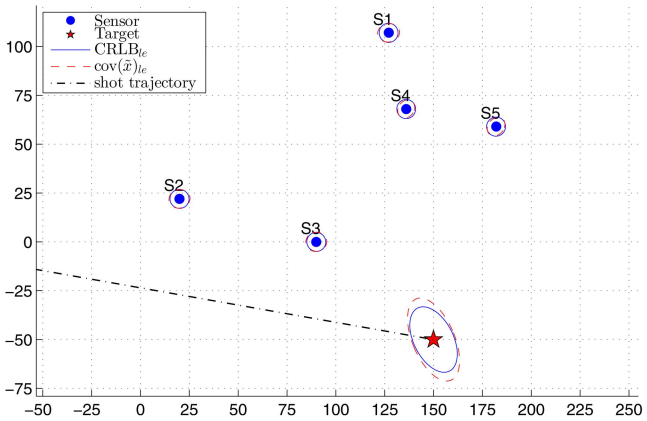


Fig. 11. Scenario 2, local estimate based CRLB$_{le}$ ellipses and error ellipses of estimated target and sensor locations (all 95%).

The results of Scenario 2 are shown in Figures 9–11. Each figure once again shows the various 95% error ellipses. Figure 9 shows the error ellipses of the bearings-only CRLB$_{bo}$ and the estimation errors, Figure 10 shows the error ellipses of the native measurement CRLB$_{nm}$ and estimation errors, and Figure 11 shows the error ellipses of the local estimate CRLB$_{le}$ and estimation errors.

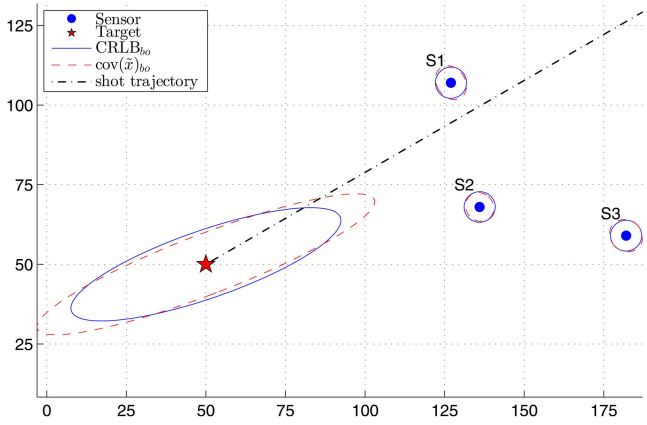


Fig. 12. Scenario 3, bearings-only CRLB$_{bo}$ ellipses and error ellipses of estimated target and sensor locations (all 95%).

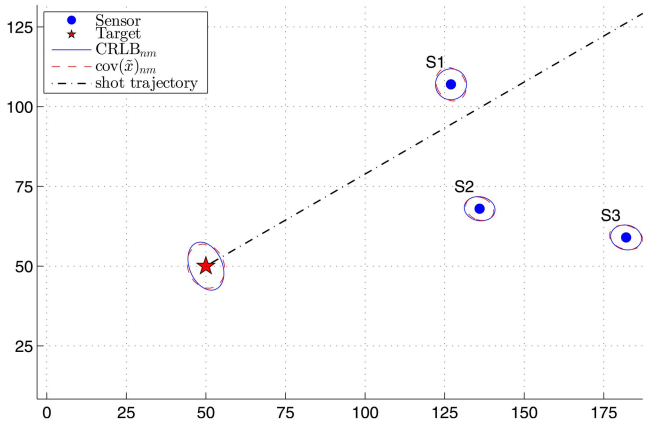


Fig. 13. Scenario 3, native measurement CRLB$_{nm}$ ellipses and error ellipses of estimated target and sensor locations (all 95%).

The covariance matrices for all versions of the CRLB again closely match those obtained from the estimation errors, indicating that the estimator is once again efficient and the range estimates carry very little information. In this scenario, the advantage of sending the native measurements is minimal in comparison to Scenario 1.

Scenario 3 consists of an identical situation to Scenario 1, but with sensors 2 and 3 removed. In this case, the geometry of the sensors and target is poor, with each sensor having very similar line-of-sight (LOS) angles to the target.

The results of Scenario 3 are shown in Figures 12–14. In this case, the local estimate CRLB$_{le}$ and estimation errors are improved over the bearings-only case, and the native measurement case is considerably better than either of the alternatives. The covariance of the estimation errors of the bearings-only case does not match well to the corresponding CRLB, which suggests that the estimator may not be efficient in this case. A statistical hypothesis test for efficiency via the normalized estimation error squared (NEES) is carried out to more rigorously examine the statistical efficiency of each case.

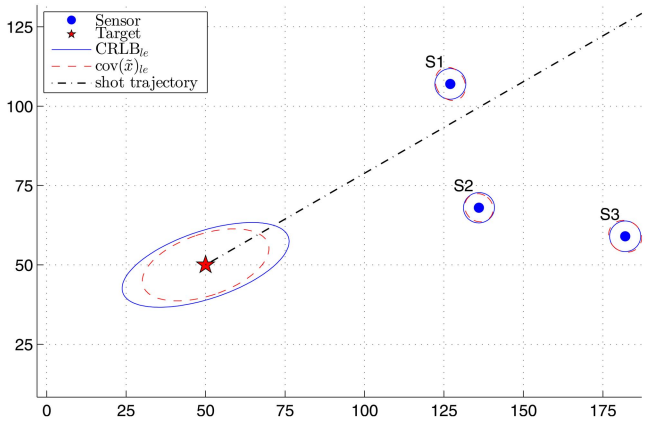


Fig. 14. Scenario 3, local estimate based CRLB$_{le}$ ellipses and error ellipses of estimated target and sensor locations (all 95%).

The NEES for the source localization was examined for each scenario, using the bearings-only CRLB$_{bo}$, the native measurement CRLB$_{nm}$, and the local estimate based CRLB$_{le}$, to provide a statistical confirmation of the efficiency of the estimator. In each case, the CRLB was evaluated at the true $\mathbf{x}$. The NEES was calculated for the following:

(i) the fused position estimation errors using bearings-only measurements with the bearings-only CRLB,
(ii) the fused position estimation errors using local estimates with the local estimate based CRLB, and
(iii) the fused position estimation errors using native measurements with the native measurement CRLB.

The NEES results (with the 95% probability region based on the chi-square distribution with two degrees of freedom and 100 Monte Carlo runs [2], being [1.63, 2.41]) are shown in Figures 15–17. Each scenario was simulated with multiple levels of angular measurement noise, namely, the standard deviations σ_ϕ and σ_φ were varied from 10% to 150% of their original value of 4°. The remaining measurement noise standard deviations remained the same as in the previous simulations.

The NEES results show that each estimation scheme is statistically efficient, with the exception of the bearings-only and local estimate based case of Scenario 3, and is "marginally" efficient for the local estimate based case of Scenario 2. The confidence region for the NEES assumes that the estimate errors are Gaussian distributed, which is an approximation in the local estimate case (see Section 2.2). For the more difficult geometry of Scenario 3, the bearings-only case loses efficiency for higher levels of angular measurement noise. Likewise, at very small levels of measurement noise, the local estimate scheme appears inefficient. This is likely due to the approximations involved in the range variance, and in the assumption that the range estimation errors are Gaussian distributed.

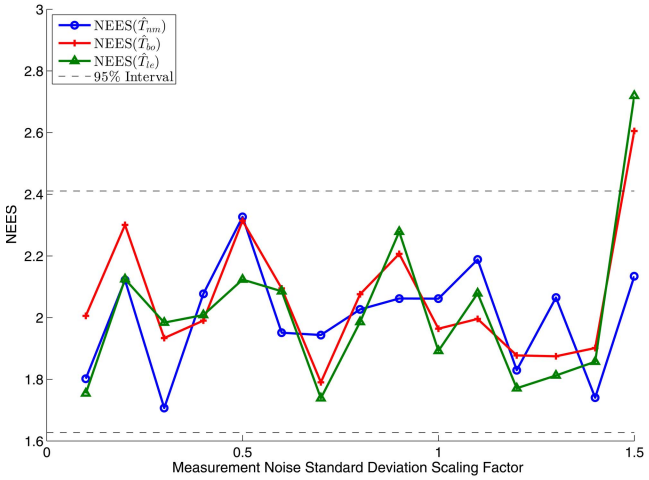


Fig. 15. Scenario 1, NEES for different levels of angular measurement noise.

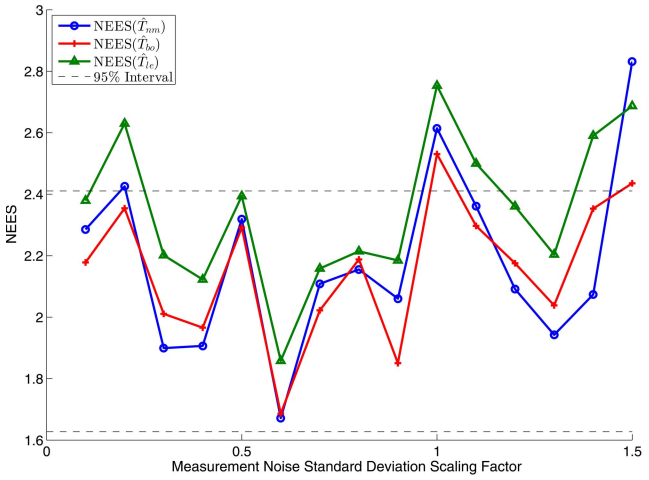


Fig. 16. Scenario 2, NEES for different levels of angular measurement noise.

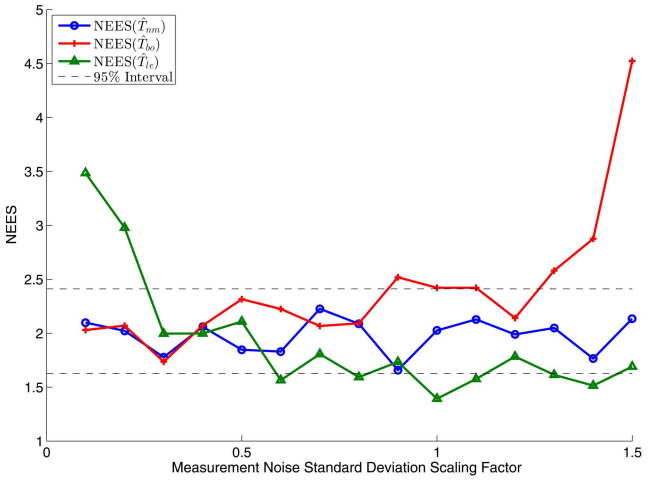


Fig. 17. Scenario 3, NEES for different levels of angular measurement noise.

Figures 18–20 show the target position root mean squared error (RMSE) (coordinate-combined) for Scenarios 1–3, along with the CRLB, for the bearings-only, native measurement, and local estimate cases.

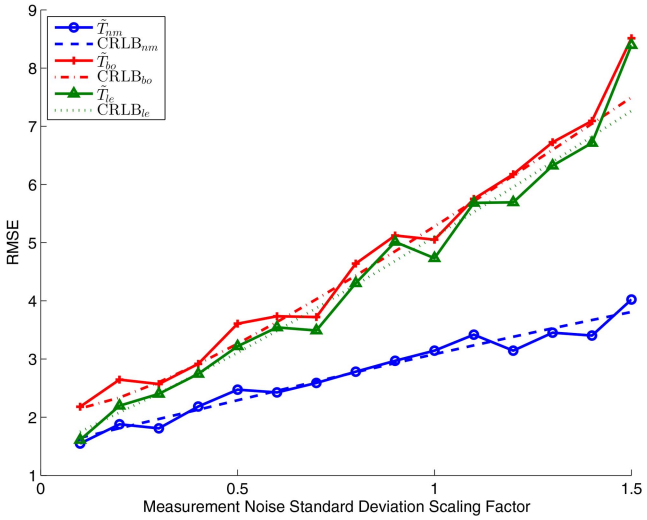


Fig. 18. Scenario 1, target position RMSE for different levels of angular measurement noise.

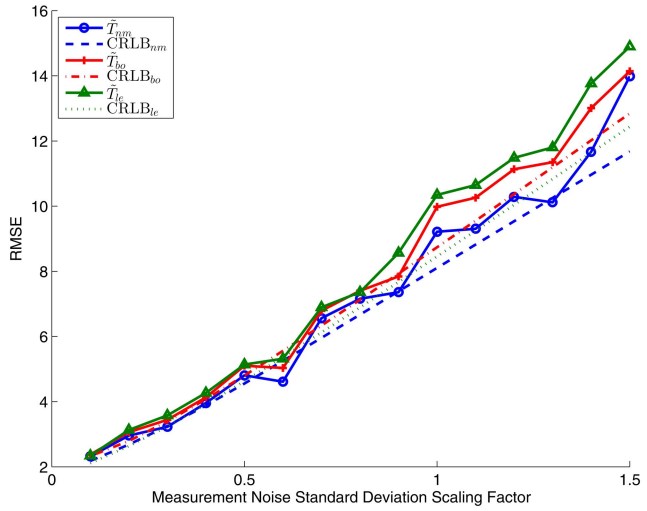


Fig. 19. Scenario 2, target position RMSE for different levels of angular measurement noise.

Figures 18 and 19 show that, over a range of angular measurement noise levels, the favorable geometry of Scenarios 1 and 2 provides for very little differentiation in the performance of target localization between the bearings-only and local estimate cases. Figure 20 shows that, for the less favorable geometry of Scenario 3, the inclusion of range estimates provides a significant increase in the accuracy of target localization. In both Scenarios 1 and 3, as seen in Figures 18 and 20, the native measurement case provides significantly improved target localization accuracy over the local estimate based case.

It should also be noted that Figure 20 appears to show the local estimate case outperforming its CRLB in Scenario 3. The CRLB in the local estimate case is necessarily approximate due to two factors in particular: namely, that the local estimates are assumed to have zero crosscovariances (because the use of approximate

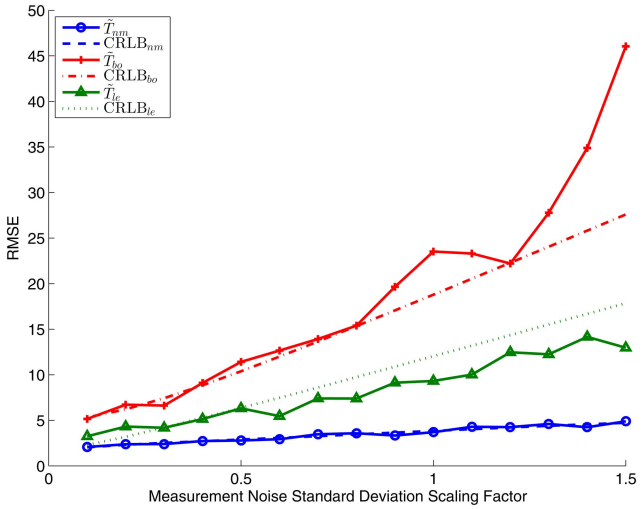


Fig. 20. Scenario 3, target position RMSE for different levels of angular measurement noise.

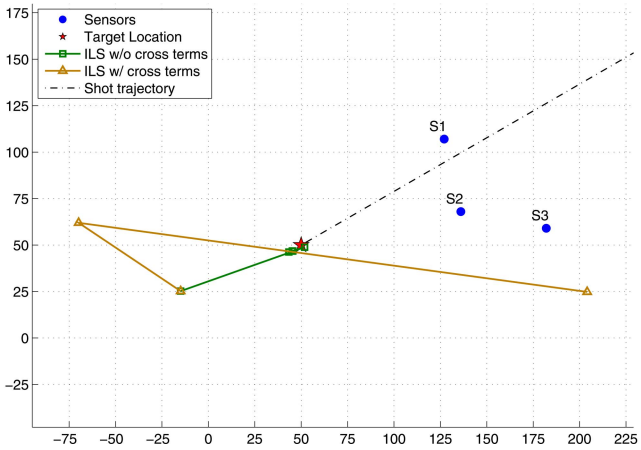


Fig. 21. Comparison of ILS iterations with and without approximate crosscovariance terms (the overlapping square and triangle represent the initial estimate).

crosscovariances can cause instability in the search), even though the simulations result in non-zero cross-covariances; and the range errors are assumed Gaussian, which was demonstrated in Section 2.2 to be an approximation. Additionally, the affect of this approximation is not obvious when there is a more favorable geometry (e.g., Scenarios 1 and 2, where there are more (bearings-only) sensors present).

Note that in every case, $\mathrm{CRLB}_{\mathrm{nm}}$ is always lower than $\mathrm{CRLB}_{\mathrm{le}}$, and indeed the performance of the native measurements is always better than the local estimates. This is unsurprising as the local estimates are derived from the native measurements, and as such, cannot add any extra information beyond what exists in the native measurements. A better model for the distribution of the local estimates is likely needed to approach the performance of the native measurements.

Figure 21 shows an example run which demonstrates the reason for setting the crosscovariance terms of (16) to zero, rather than the (approximate) expressions provided in (17) and (18). Occasionally, when using these crosscovariance terms, the ILS search will diverge, whereas, this behavior is not observed when assuming there is no correlation between the local estimate errors. This divergence is caused by the crosscovariance terms causing (16) to become ill-conditioned, which causes difficulty in converging to the global maximum of the likelihood function (LF) surface. Note that in both cases, the local estimates are formed from the noisy native measurements, so there is indeed a correlation between the local estimate errors. The ILS algorithm must use the latest estimate to calculate the necessary terms of the covariance matrix, however, and it appears that the algorithm is more likely to diverge for non-zero crosscovariance terms. In fact, no divergence was observed in any run (in 3 scenarios, for 15 levels of measurement noise, with 100 runs each) when assuming zero crosscovariance.

Additionally, the different versions of the CRLB can be compared to gain insights into a particular scenario. The ratio of the area of the bearings-only $\mathrm{CRLB}_{\mathrm{bo}}$ ellipse to the local estimate based $\mathrm{CRLB}_{\mathrm{le}}$ ellipse can be calculated as $|(J_b^{-1})_T(J_{\mathbf{y}}^{-1})_T^{-1}|^{1/2}$, where $(J^{-1})_T$ is the portion of the CRLB that deals with the target localization (as opposed to the entire vector $\mathbf{x}$). This is plotted over a two-dimensional (2-D) grid corresponding to various shooter locations in Figure 22, for sensor locations identical to Scenario 3. Each point in the 2-D grid corresponds to the ratio of the CRLB ellipse areas for a shooter at that location, shooting toward the marked aimpoint. Figure 22 clearly shows the shooter locations where the range measurements are most beneficial and the bearings-only localization will perform particularly poorly. Figure 23 shows the same results, only this time the native measurement $\mathrm{CRLB}_{\mathrm{nm}}$ is compared with the bearings-only $\mathrm{CRLB}_{\mathrm{bo}}$. Figure 23 shows a slightly different aimpoint in order to demonstrate the large difference in performance that is achieved when the bullet trajectory passes between different sensors. When the bullet trajectory passes on the same side of every sensor (which amounts to every sensor seeing the same shock-wave DOA φ) the performance is not much improved over either the bearings-only case or the local estimate based case (this is further demonstrated by the results of Scenario 2, which corresponds to such a sensor-target geometry). When the bullet trajectory passes between sensors (as in Scenario 1 and 3 above), the performance of the target localization is greatly improved by using the native measurements.

5. CONCLUSIONS

The CRLB and statistical efficiency were examined for multiple scenarios of a localization system using either native measurements or local estimates, where there are position-dependent noise terms. The system

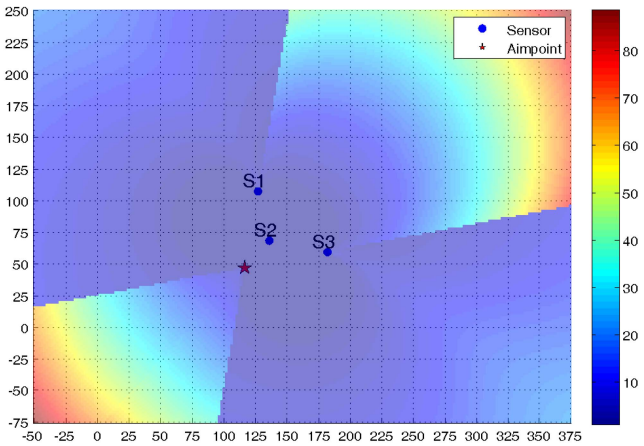


Fig. 22. Comparison of local estimate based CRLB_{le} and bearings-only CRLB_{bo} ($|(J_b^{-1})_T(J_{\mathbf{y}}^{-1})_T^{-1}|^{1/2}$) over a 2-D grid (E-N, in m) of shooter position (for a fixed aimpoint at $\star$).

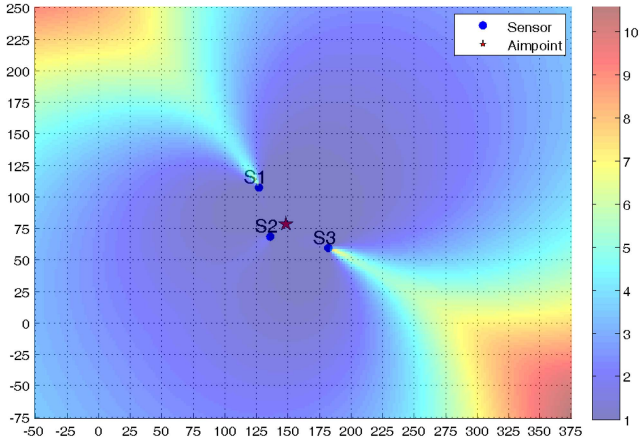


Fig. 23. Comparison of native measurement CRLB_{nm} and bearings-only CRLB_{bo} ($|(J_b^{-1})_T(J_{\eta}^{-1})_T^{-1}|^{1/2}$) over a 2-D grid (E-N, in m) of shooter position (for a fixed aimpoint at $\star$).

in question is a recently developed shooter localization scheme using acoustic gunfire detection sensors [9], [10]. The CRLB was derived for the cases of sensor nodes which send the following:

(i) bearings only (CRLB_{bo})
(ii) bearing, range, and bullet trajectory estimates ("local estimate" based CRLB_{le}), and
(iii) bearing, shockwave angle, and TDOA measurements ("native measurement" CRLB_{nm})

When range estimates are passed to the fusion center in cases where the sensor-target geometry is favorable for angle-only localization, the bearings-only results (i) closely match the results of (ii), suggesting that there is little, if any, information contained in the range measurements in those cases. If the geometry is poor, however, as in Scenario 3, there is a significant difference between the bearings-only results (i) and the local estimate based results (ii). For each of the scenarios tested, when the native measurements are passed to the fusion center (iii), the localization accuracy was improved, with a significant improvement in Scenarios 1 and 3.

The distribution of the range estimate errors was also examined in order to highlight the approximation which is made when assuming the range errors to be Gaussian distributed. This assumption is likely the cause for the discrepancy in localization performance between the local estimate based case (ii) and the native measurement case (iii).

In each scenario, the NEES shows that the estimation is statistically efficient (with the exception of the bearings-only case with high measurement noise and a poor sensor-target geometry). When native measurements are passed to the fusion center, the localization is performed particularly well for the poor geometry of Scenario 3 and very closely matches the corresponding CRLB. The results show both that the estimator used in this particular acoustic localization system is efficient, and that the CRLB can be used as an accurate means of performance prediction for such a system (particularly for the native measurement case).

APPENDIX A, COVARIANCE BETWEEN RANGE AND ANGULAR MEASUREMENTS

The ith sensor's range measurement error $\tilde{r}_i$ can be approximated (via Taylor series expansion) as

$$\tilde{r}_i \approx \frac{c\tilde{\tau}_i}{1-\cos(\phi_i-\varphi_i)} - \frac{r_i\sin(\phi_i-\varphi_i)(\tilde{\phi}_i-\tilde{\varphi}_i)}{1-\cos(\phi_i-\varphi_i)} \tag{50}$$

where $\tilde{\tau}$, $\tilde{\phi}_i$, and $\tilde{\varphi}_i$ are the errors of τ, ϕ_i, and φ_i, respectively.

In view of (50), the covariance between the ith sensor's range and bearing measurement is

$$\begin{aligned}\text{cov}(r_i,\phi_i) &= E[\tilde{r}_i\tilde{\phi}_i] \\ &= -\frac{r_i\sin(\phi_i-\varphi_i)}{1-\cos(\phi_i-\varphi_i)}\sigma_{\phi_i}^2\end{aligned} \tag{51}$$

Similarly, the covariance between the ith sensor's range and shockwave angle measurement is

$$\begin{aligned}\text{cov}(r_i,\omega) &= E[\tilde{r}_i\tilde{\varphi}_i] \\ &= \frac{r_i\sin(\phi_i-\varphi_i)}{1-\cos(\phi_i-\varphi_i)}\sigma_{\varphi}^2\end{aligned} \tag{52}$$

APPENDIX B, PARTIAL DERIVATIVE TERMS FOR LOCAL ESTIMATE FIM

In order to calculate the $\partial\mu(\mathbf{x})$ terms of (48), the partial derivatives of (6), (9) and ω are needed. The partial derivatives of (6) are

$$\frac{\partial\phi_i}{\partial T_x} = -\frac{T_y - S_{i_y}}{r_i^2} \tag{53}$$

$$\frac{\partial\phi_i}{\partial T_y} = \frac{T_x - S_{i_x}}{r_i^2} \tag{54}$$

$$\frac{\partial \phi_i}{\partial \omega} = 0 \tag{55}$$

$$\frac{\partial \phi_i}{\partial S_x} = \frac{T_y - S_{i_y}}{r_i^2} \tag{56}$$

$$\frac{\partial \phi_i}{\partial S_y} = -\frac{T_x - S_{i_x}}{r_i^2} \tag{57}$$

The partial derivatives of (9) are

$$\frac{\partial r_i}{\partial T_x} = \frac{T_x - S_{i_x}}{r_i} \tag{58}$$

$$\frac{\partial r_i}{\partial T_y} = \frac{T_y - S_{i_y}}{r_i} \tag{59}$$

$$\frac{\partial r_i}{\partial \omega} = 0 \tag{60}$$

$$\frac{\partial r_i}{\partial S_x} = -\frac{T_x - S_{i_x}}{r_i} \tag{61}$$

$$\frac{\partial r_i}{\partial S_y} = -\frac{T_y - S_{i_y}}{r_i} \tag{62}$$

The partial derivatives of ω are straightforward.

The $\partial\Sigma(\mathbf{x})$ terms of (48) require partial derivatives of (10), (17) and (18).

The partial derivatives of (10) are

$$\begin{aligned}\frac{\partial \sigma_{r_i}^2}{\partial \omega} &= \frac{2}{\sin(\phi_i - \varphi_i)}\left[\sigma_{r_i}^2 + \frac{c^2\sigma_\tau^2\cos(\phi_i - \varphi_i)}{(1-\cos(\phi_i - \varphi_i))^2}\right] \\ &= \frac{2\sin(\phi_i - \varphi_i)}{1-\cos(\phi_i - \varphi_i)}\left[\sigma_{r_i}^2 - \frac{(\sigma_\phi^2 + \sigma_\varphi^2)r_i^2\cos(\phi_i - \varphi_i)}{1-\cos(\phi_i - \varphi_i)}\right]\end{aligned} \tag{63}$$

$$\begin{aligned}\frac{\partial \sigma_{r_i}^2}{\partial T_x} &= -\frac{\partial \phi_i}{\partial T_x}\frac{\partial \sigma_{r_i}^2}{\partial \omega} \\ &\quad + \frac{2(T_x - S_{i_x})(\sigma_\phi^2 + \sigma_\varphi^2)\sin^2(\phi_i - \varphi_i)}{(1-\cos(\phi_i - \varphi_i))^2}\end{aligned} \tag{64}$$

$$\begin{aligned}\frac{\partial \sigma_{r_i}^2}{\partial T_y} &= -\frac{\partial \phi_i}{\partial T_y}\frac{\partial \sigma_{r_i}^2}{\partial \omega} \\ &\quad + \frac{2(T_y - S_{i_y})(\sigma_\phi^2 + \sigma_\varphi^2)\sin^2(\phi_i - \varphi_i)}{(1-\cos(\phi_i - \varphi_i))^2}\end{aligned} \tag{65}$$

$$\frac{\partial \sigma_{r_i}^2}{\partial S_{i_x}} = -\frac{\partial \sigma_{r_i}^2}{\partial T_x} \tag{66}$$

$$\frac{\partial \sigma_{r_i}^2}{\partial S_{i_y}} = -\frac{\partial \sigma_{r_i}^2}{\partial T_y} \tag{67}$$

The partial derivatives of (17) are

$$\frac{\partial \mathrm{cov}(r_i, \phi_i)}{\partial \omega} = -\frac{r_i\sigma_\phi^2}{1-\cos(\phi_i - \varphi_i)} \tag{68}$$

$$\frac{\partial \mathrm{cov}(r_i, \phi_i)}{\partial T_x} = -\frac{\partial \phi_i}{\partial T_x}\frac{\partial \mathrm{cov}(r_i, \phi_i)}{\partial \omega} + \frac{\partial \phi_i}{\partial T_y}\mathrm{cov}(r_i, \phi_i) \tag{69}$$

$$\frac{\partial \mathrm{cov}(r_i, \phi_i)}{\partial T_y} = -\frac{\partial \phi_i}{\partial T_y}\frac{\partial \mathrm{cov}(r_i, \phi_i)}{\partial \omega} - \frac{\partial \phi_i}{\partial T_x}\mathrm{cov}(r_i, \phi_i) \tag{70}$$

$$\frac{\partial \mathrm{cov}(r_i, \phi_i)}{\partial S_x} = -\frac{\partial \mathrm{cov}(r_i, \phi_i)}{\partial T_x} \tag{71}$$

$$\frac{\partial \mathrm{cov}(r_i, \phi_i)}{\partial S_y} = -\frac{\partial \mathrm{cov}(r_i, \phi_i)}{\partial T_y} \tag{72}$$

The partial derivatives of (18) are

$$\frac{\partial \mathrm{cov}(r_i, \omega_i)}{\partial \omega} = \frac{r_i\sigma_\varphi^2}{1-\cos(\phi_i - \varphi_i)} \tag{73}$$

$$\frac{\partial \mathrm{cov}(r_i, \omega_i)}{\partial T_x} = -\frac{\partial \phi_i}{\partial T_x}\frac{\partial \mathrm{cov}(r_i, \omega_i)}{\partial \omega} + \frac{\partial \phi_i}{\partial T_y}\mathrm{cov}(r_i, \omega_i) \tag{74}$$

$$\frac{\partial \mathrm{cov}(r_i, \omega_i)}{\partial T_y} = -\frac{\partial \phi_i}{\partial T_y}\frac{\partial \mathrm{cov}(r_i, \omega_i)}{\partial \omega} - \frac{\partial \phi_i}{\partial T_x}\mathrm{cov}(r_i, \omega_i) \tag{75}$$

$$\frac{\partial \mathrm{cov}(r_i, \omega_i)}{\partial S_x} = -\frac{\partial \mathrm{cov}(r_i, \omega_i)}{\partial T_x} \tag{76}$$

$$\frac{\partial \mathrm{cov}(r_i, \omega_i)}{\partial S_y} = -\frac{\partial \mathrm{cov}(r_i, \omega_i)}{\partial T_y} \tag{77}$$

The $\partial\Sigma(\mathbf{x})$ terms of (48) can now be constructed from (63)–(77).

REFERENCES

[1] J. Ash, G. Whipps, and R. Kozick
"Performance of shockwave-based shooter localization under model misspecification,"
in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 2010, pp. 2694–2697.

[2] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software.
J. Wiley and Sons, 2001.

[3] J. L. Crassidis, R. Alonso, and J. L. Junkins
"Optimal Attitude and Position Determination from Line-of-Sight Measurements,"
Journal of the Astronautical Sciences, vol. 48(2), pp. 391–408, 2000.

[4] D. F. Crouse, R. W. Osborne, III, K. Pattipati, P. Willett, and Y. Bar-Shalom
"2D Location Estimation of Angle-Only Sensor Arrays Using Targets of Opportunity,"
in *13th International Conference on Information Fusion*, Edinburgh, UK, Jul. 2010, pp. 1–8.

[5] T. Damarla, L. M. Kapplan, and G. T. Whipps
"Sniper localization using acoustic asynchronous sensors,"
IEEE Sensors Journal, vol. 10, no. 9, pp. 1469–1478, Sep. 2010.

[6] P. Fishbane, S. Gasiorowicz, and S. Thornton
Physics for Scientists and Engineers,
2nd ed. Upper Saddle River, NJ: Prentice Hall, 1996, vol. 1, ch. 14, pp. 398–399.

[7] W. H. Foy
"Position-Location Solutions by Taylor-Series Estimation,"
IEEE Trans. on Aerospace and Electronic Systems, vol. 12(2), pp. 187–194, Mar. 1976.

[8] M. Gavish and A. J. Weiss
"Performance Analysis of Bearing-Only Target Location Algorithms,"
IEEE Trans. on Aerospace and Electronic Systems, vol. 28(3), pp. 817–828, Jul. 1992.

[9] J. George and L. M. Kaplan
"Shooter Localization using Soldier-Worn Gunfire Detection Systems,"
in *14th International Conference on Information Fusion*, Chicago, IL, Jul. 2011, pp. 398–405.

[10] J. George and L. M. Kaplan
"Shooter Localization using a Wireless Sensor Network of Soldier-Worn Gunfire Detection Systems,"
Journal of Advances in Information Fusion, vol. 8, no. 1, Jun. 2013.

[11] D. Grasing and S. Desai
"Data fusion methods for small arms localization solutions,"
in *Information Fusion (FUSION), 2012 15th International Conference on*, 2012, pp. 713–718.

[12] L. M. Kaplan, T. Damarla, and T. Pham
"QoI for Passive Acoustic Gunfire Localization,"
in *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, Oct. 2008, pp. 754–759.

[13] R. J. Kozick, J. George, L. M. Kaplan, and S. Deligeorges
"Centralized fusion of acoustic sensor data for gunfire localization,"
in *2011 MSS Battlespace Acoustic and Seismic Sensing, Magnetic and Electric Field Sensors (BAMS)*, Oct. 2011.

[14] R. J. Kozick, G. T. Whipps, and J. N. Ash
"Supersonic projectile models for asynchronous shooter localization,"
in *Proc. SPIE Conf. Unattended Ground, Sea, and Air Sensor Technologies and Applications*, vol. 8046, May 2011.

[15] P. Kuckertz, J. Ansari, J. Riihijarvi, and P. Mahonen
"Sniper fire localization using wireless sensor networks and genetic algorithm based data fusion,"
in *IEEE Military Communications Conference*, 2007, pp. 1–8.

[16] D. Lindgren, O. Wilsson, F. Gustafsson, and H. Habberstad
"Shooter localization in wireless sensor networks,"
in *12th International Conference on Information Fusion*, Seattle, WA, 2009, pp. 404–411.

[17] K. W. Lo and B. G. Ferguson
"Localization of small arms fire using acoustic measurements of muzzle blast and/or ballistic shock wave arrivals,"
J. Acoust. Soc. Am., vol. 132, no. 5, pp. 2997–3017, Nov. 2012.

[18] R. W. Osborne, III and Y. Bar-Shalom
"Statistical Efficiency of Composite Position Measurements from Passive Sensors,"
in *Proc. SPIE Conf. Signal Processing, Sensor Fusion and Target Recognition*, vol. 8050-07, Orlando, FL, Apr. 2011.

[19] B. Porat and B. Friedlander
"Computation of the Exact Information Matrix of Gaussian Time Series with Stationary Random Components,"
IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. 34, no. 1, pp. 118–130, Feb. 1986.

[20] J. Sallai, A. Lédeczi, and P. Völgyesi
"Acoustic shooter localization with a minimal number of single-channel wireless sensor nodes,"
in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '11. New York, NY, USA: ACM, 2011, pp. 96–107. [Online]. Available: http://doi.acm.org/10.1145/2070942.2070953.

[21] J. Sallai, P. Völgyesi, A. Lédeczi, K. Pence, T. Bapty, S. Neema, and J. R. Davis
"Acoustic shockwave-based bearing estimation,"
in *Proceedings of the 12th international conference on Information processing in sensor networks*, ser. IPSN '13. New York, NY, USA: ACM, 2013, pp. 217–228. [Online]. Available: http://doi.acm.org/10.1145/2461381.2461409.

[22] D. J. Torrieri
"Statistical Theory of Passive Location Systems,"
IEEE Trans. on Aerospace and Electronic Systems, vol. 20(2), pp. 183–198, Mar. 1984.

[23] P. Volgyesi, G. Balogh, A. Nadas, C. B. Nash, and A. Ledeczi
"Shooter localization and weapon classification with soldier-wearable networked sensors,"
in *Proceedings of the 5th international conference on Mobile systems, applications and services*, ser. MobiSys '07. New York, NY, USA: ACM, 2007, pp. 113–126. [Online]. Available: http://doi.acm.org/10.1145/1247660.1247676.

[24] M. Wax
"Position Location from Sensors with Position Uncertainty,"
IEEE Trans. on Aerospace and Electronic Systems, vol. 19(5), pp. 658–662, Sep. 1983.

[25] Y. Zhao, D. Yi, Y. Li, and Q. Zhang
"CRLB of Initial State Estimation for Boost Phase Object Based on 8-State Gravity Turn Model Using Space-Based Observations,"
in *2010 2nd International Conference on Signal Processing Systems (ICSPS)*, vol. 1, Dalian, China, Jul. 2010, pp. 781–789.

Richard W. Osborne, III obtained his B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Connecticut in 2004, 2007, and 2012, respectively. He is currently an Assistant Research Professor in the Electrical Engineering department at the University of Connecticut, Storrs, CT. His academic interests include adaptive target tracking, information/sensor fusion, machine vision, and other aspects of estimation.

Yaakov Bar-Shalom was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion, Israel Institute of Technology, in 1963 and 1967 and the Ph.D. degree from Princeton University in 1970, all in electrical engineering. From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, California. Currently he is Board of Trustees Distinguished Professor in the Dept. of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering at the University of Connecticut. He is also Director of the ESP (Estimation and Signal Processing) Lab. His current research interests are in estimation theory, target tracking and data fusion. He has published over 500 papers and book chapters in these areas and in stochastic adaptive control. He coauthored the monograph Tracking and Data Association (Academic Press, 1988), the graduate texts Estimation and Tracking: Principles, Techniques and Software (Artech House, 1993), Estimation with Applications to Tracking and Navigation: Algorithms and Software for Information Extraction (Wiley, 2001), the advanced graduate texts Multitarget-Multisensor Tracking: Principles and Techniques (YBS Publishing, 1995), Tracking and Data Fusion (YBS Publishing, 2011), and edited the books Multitarget-Multisensor Tracking: Applications and Advances (Artech House, Vol. I, 1990; Vol. II, 1992; Vol. III, 2000). He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multi-target tracking." He has been consulting to numerous companies and government agencies, and originated the series of Multitarget-Multisensor Tracking short courses offered via UCLA Extension, at Government Laboratories, private companies and overseas. During 1976 and 1977 he served as Associate Editor of the IEEE Transactions on Automatic Control and from 1978 to 1981 as Associate Editor of Automatica. He was Program Chairman of the 1982 American Control Conference, General Chairman of the 1985 ACC, and Co-Chairman of the 1989 IEEE International Conference on Control and Applications. During 1983–87 he served as Chairman of the Conference Activities Board of the IEEE Control Systems Society and during 1987–89 was a member of the Board of Governors of the IEEE CSS. He was a member of the Board of Directors of the International Society of Information Fusion (1999–2004) and served as General Chairman of FUSION 2000, President of ISIF in 2000 and 2002 and Vice President for Publications in 2004–13. In 1987 he received the IEEE CSS Distinguished Member Award. Since 1995 he is a Distinguished Lecturer of the IEEE AESS and has given numerous keynote addresses at major national and international conferences. He is co-recipient of the M. Barry Carlton Award for the best paper in the IEEE Transactions on Aerospace and Electronic Systems in 1995 and 2000 and recipient of the 1998 University of Connecticut AAUP Excellence Award for Research. In 2002 he received the J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group. He is a member of the Connecticut Academy of Science and Engineering. In 2008 he was awarded the IEEE Dennis J. Picard Medal for Radar Technologies and Applications, and in 2012 the Connecticut Medal of Technology. He has been listed by academic.research.microsoft (top authors in engineering) as #1 among the researchers in Aerospace Engineering based on the citations of his work.

Jemin George received his M.S. ('07), and Ph.D. ('10) in Aerospace Engineering from the State University of New York at Buffalo. In 2008, he was a Summer Research Scholar with the U.S. Air Force Research Laboratory's Space Vehicles Directorate at Kirtland Air Force Base in Albuquerque, New Mexico. He was a National Aeronautics and Space Administration Langley Aerospace Research Summer Scholar with the Langley Research Center in 2009. From 2009–2010 he was a Research Fellow with the Stochastic Research Group, Department of Mathematics, Technische Universitat Darmstadt, Darmstadt, Germany. He joined ARL in 2010 as a civilian employee and is currently with the Networked Sensing & Fusion Branch of the Signal & Image Processing Division. He has published over 30 refereed technical articles in the areas of estimation, control and information fusion. He received the 2012 Army Research and Development Achievement Award for Outstanding Collaboration. He has chaired numerous technical conference sessions at the Institute of Electrical and Electronics Engineers (IEEE) American Control Conference (ACC), Society for Industrial and Applied Mathematics (SIAM) Conference on Uncertainty Quantification, and The American Institute of Aeronautics and Astronautics (AIAA) Guidance, Navigation and Control Conference. He currently serves on the Technical Program Committee for the International Conference on Information Fusion (FUSION) and as the Associate Editor for Contributed Papers for the ACC. His principal research interests include stochastic systems, control theory, nonlinear filtering, information fusion, adaptive networks, distributed sensing and estimation.

Lance Kaplan received the B.S. degree with distinction from Duke University, Durham, NC, in 1989 and the M.S. and Ph.D. degrees from the University of Southern California, Los Angeles, in 1991 and 1994, respectively, all in Electrical Engineering. From 1987–1990, Dr. Kaplan worked as a Technical Assistant at the Georgia Tech Research Institute. He held a National Science Foundation Graduate Fellowship and a USC Dean's Merit Fellowship from 1990–1993, and worked as a Research Assistant in the Signal and Image Processing Institute at the University of Southern California from 1993–1994. Then, he worked on staff in the Reconnaissance Systems Department of the Hughes Aircraft Company from 1994–1996. From 1996–2004, he was a member of the faculty in the Department of Engineering and a senior investigator in the Center of Theoretical Studies of Physical Systems (CTSPS) at Clark Atlanta University (CAU), Atlanta, GA. Currently, he is a researcher in the Networked Sensing and Fusion branch of the U.S. Army Research Laboratory. Dr. Kaplan serves as Editor-In-Chief for the IEEE Transactions on Aerospace and Electronic Systems (AES). In addition, he also serves on the Board of Governors of the IEEE AES Society and on the Board of Directors of the International Society of Information Fusion. He is a three time recipient of the Clark Atlanta University Electrical Engineering Instructional Excellence Award from 1999–2001. His current research interests include signal and image processing, automatic target recognition, information/data fusion, and resource management.

Recursive Bingham Filter for Directional Estimation Involving 180 Degree Symmetry

GERHARD KURZ
IGOR GILITSCHENSKI
SIMON JULIER
UWE D. HANEBECK

This work considers filtering of uncertain data defined on periodic domains, particularly the circle and the manifold of orientations in 3D space. Filters based on the Kalman filter perform poorly in this directional setting as they fail to take the structure of the underlying manifold into account. We present a recursive filter based on the Bingham distribution, which is defined on the considered domains. The proposed filter can be applied to circular filtering problems with 180 degree symmetry and to estimation of orientations in three dimensional space. It is easily implemented using standard numerical techniques and suitable for real-time applications. We evaluate our filter in a challenging scenario and compare it to a Kalman filtering approach adapted to the particular setting.

Manuscript received October 23, 2013; revised June 5, 2014 and September 23, 2014; released for publication September 23, 2014.

Refereeing of this contribution was handled by Peter Willett.

This is an extended version of the paper *Recursive Estimation of Orientation Based on the Bingham Distribution* published at the *16th International Conference on Information Fusion (Fusion 2013)*, which received the *Best Student Paper Award, First Runner-Up*.

Authors' addresses: G. Kurz, I. Gilitschenski, and U. Hanebeck, Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany (e-mail: {gerhard.kurz, gilitschenski}@kit.edu, uwe.hanebeck@ieee.org); S. Julier, Virtual Environments and Computer Graphics Group, Department of Computer Science, University College London (UCL), United Kingdom (e-mail: s.julier@cs.ucl.ac.uk).

1. INTRODUCTION

Tracking cars, ships, or airplanes may involve estimation of their current orientation or heading. Furthermore, many applications in the area of robotics or augmented reality depend on reliable estimation of the pose of certain objects. When estimating the orientation of two-way roads or relative angles of two unlabeled targets, the estimation task can be thought of as estimation of a directionless orientation. Thus, the estimation task reduces to estimating the alignment of an axis, i.e., estimation with 180° symmetry.

All these estimation problems share the need for processing angular or directional data, which differs in many ways from the linear setting. First, periodicity of the underlying manifold needs to be taken into account. Second, directional quantities do not lie in a vector space. Thus, there is no equivalent to a linear model, as there are no linear mappings. These problems become particularly significant for high uncertainties, e.g., as a result of poor initialization, inaccurate sensors such as magnetometers, or sparse measurements causing a large integration error.

In many applications, even simple estimation problems involving angular data are often considered as linear or nonlinear estimation problems on linear domains and handled with techniques such as the Kalman Filter [19], the Extended Kalman Filter (EKF), or the Unscented Kalman Filter (UKF) [17]. In a circular setting, most approaches to filtering suffer from assuming a Gaussian probability density at a certain point. They fail to take into account the periodic nature of the underlying domain and assume a (linear) vector space instead of a curved manifold. This shortcoming can cause poor results, in particular when the angular uncertainty is large. In certain cases, the filters may even diverge.

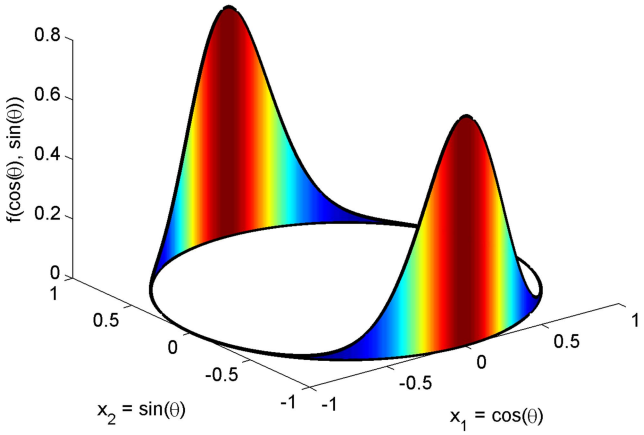


Fig. 1. Bingham probability density function with $\mathbf{M} = \mathbf{I}_{2\times 2}$ and $\mathbf{Z} = \mathrm{diag}(-8,0)$ as a 3D plot. This corresponds to a standard deviation of 16°.

Strategies to avoid these problems in an angular setting involve an "intelligent" repositioning of measurements (typically by multiples of π) or even discarding certain undesired measurements. Sometimes, nonlinear

equality constraints have to be fulfilled, for example, unit length of a vector, which makes it necessary to inflate the covariance [16]. There are also approaches that use operators on a manifold to provide a local approximation of a vector space [13]. While these approaches yield reasonable results in some circumstances, they still suffer from ignoring the true geometry of circular data within their probabilistic models, which are usually based on assuming normally distributed noise. This assumption is often motivated by the Central Limit Theorem, i.e., the limit distribution of a normalized sum of i.i.d. random variables with finite variance is normally distributed [42]. However, this motivation does not apply to uncertain data from a periodic domain. Thus, choosing a circular distribution for describing uncertainty can offer better results.

In this paper, we consider the use of the Bingham distribution [5] (see Fig. 1) for recursive estimation. The Bingham distribution is defined on the hypersphere of arbitrary dimension. Here, we focus on the cases of two- and four-dimensional Bingham distributed random vectors and apply our results to angular estimation with 180° symmetry and estimating orientation in 3D space.

Estimating orientation is achieved by using unit quaternions to represent the full 3D orientation of an object. It is well known that quaternions avoid the singularities present in other representations such as Euler angles [25]. Their only downsides are the fact that they must remain normalized and the property that the quaternions $\underline{q}$ and $-\underline{q}$ represent the same orientation. Both of these issues can elegantly be resolved by use of the Bingham distribution, since it is by definition restricted to the hypersphere and is 180° symmetric.

This work extends our results on Bingham filtering [32] and the first-order quaternion Bingham filter proposed in [11] in several ways. First of all, we present a relationship between the two-dimensional Bingham distribution and the von Mises distribution and we show how to exploit it to obtain a more efficient way of computing the normalization constant and its derivatives. Furthermore, we show a relation to the von Mises-Fisher distribution, which can be used to speed up parameter estimation and moment matching procedures in an important special case. In that situation, we avoid the need for precomputed lookup tables. This is of considerable interest because the computation of the normalization constant plays a crucial role for the performance of the Bingham filter. Finally, we perform a more thorough evaluation of both two- and four-dimensional scenarios using different types of noise distributions and different degrees of uncertainty.

This paper is structured as follows. First, we present an overview of previous work in the area of directional statistics and angular estimation (Sec. 2). Then, we introduce our key idea in Sec. 3. In Sec. 4, we give a detailed introduction to the Bingham distribution and in Sec. 5, we derive the necessary operations needed to create a recursive Bingham filter. Based on these prerequisites, we introduce our filter in Sec. 6. We have carried out an evaluation in simulations, which is presented in Sec. 7. Finally, we conclude this work in Sec. 8.

2. RELATED WORK

Directional statistics is a subdiscipline of statistics, which focuses on dealing with directional data. That is, it considers random variables which are constrained to lie on manifolds (for example the circle or the sphere) rather than random variables located in d-dimensional vector spaces (typically $\mathbb{R}^d$). Classical results in directional statistics are summed up in the books by Mardia and Jupp [37] and by Jammalamadaka and Sengupta [15]. Probability distributions on the unit sphere are described in more detail in [6].

There is a broad range of research investigating the two-dimensional orientation estimation. A recursive filter based on the von Mises distribution for estimating the orientation on the $SO(2)$ was presented in [3], [45]. It has been applied to GPS phase estimation problems [44]. Furthermore, a nonlinear filter based on von Mises and wrapped normal distributions was presented in [30], [31]. This filter takes advantage of the fact that wrapped normal distributions are closed under convolution and the fact that von Mises distributions are closed under Bayesian inference. This filter has also been applied to constrained object tracking [29].

In 1974, Bingham published the special case for three dimensions of his distribution in [5], which he originally developed in his PhD thesis [4]. Further work on the Bingham distribution has been done by Kent [21], [22] as well as Jupp and Mardia [18], [35]. So far, there have been a few applications of the Bingham distribution, for example in geology [36], [28], [33].

Antone published some results on a maximum likelihood approach for Bingham-based pose estimation in 2001 [2]. However, this method was limited to offline applications. In 2011, Glover used the Bingham distribution for a Monte Carlo based pose estimation [10], which he later generalized into a quaternion-based recursive filter [11] and applied it to tracking the spin of a ping pong ball [12]. Glover also released a library called `libbingham` [9] that includes C and MATLAB implementations of some of the methods discussed in Sec. 4. It should be noted that our implementation is not based on `libbingham`. Our implementation calculates the normalization constant online, whereas `libbingham` relies on values that have been precomputed offline. In the case of a two-dimensional Bingham-distributed random vector, the computation of the normalization constant of the corresponding probability density function reduces to the evaluation of Bessel functions. In higher dimensions, a saddlepoint approximation can be used [26].

In 2013, we proposed a recursive Bingham filter for 2D axis estimation [32], which serves as a foundation for this paper. We also published a nonlinear generalization to the quaternion case in [8].

3. KEY IDEA OF THE BINGHAM FILTER

In this paper, we derive a recursive filter based on the Bingham distribution for two- and four-dimensional random vectors of unit length, because they can be used to represent orientations on the plane and in three-dimensional space. Rather than relying on approximations involving the Gaussian distribution, we chose to represent all occurring probability densities as Bingham distributions. The Bingham distribution is defined on the hypersphere and is antipodally symmetric. Our use of the Bingham distribution is motivated by its convenient representation of hyperspherical random vectors, its relationship to the Gaussian distribution, and a maximum entropy property [35]. Although we restrict ourselves to the two- and four-dimensional cases in this paper, we would like to emphasize that some of the presented methods can easily be generalized to higher dimensions.

In order to derive a recursive filter, we need to be able to perform two operations. First, we need to calculate the predicted state at the next time step from the current state and the system noise affecting the state. In a recursive estimation problem in $\mathbb{R}^d$ with additive noise, this involves a convolution with the noise density. We provide a suitable analogue on the hypersphere in order to account for the *composition* of uncertain rotations. Since Bingham distributions are not closed under this operation, we present an approximate solution to this problem based on matching covariance matrices.

Second, we need to perform a Bayes update. As usual, this requires the *multiplication* of the prior density with the likelihood density. We prove that Bingham distributions are closed under multiplication and show how to obtain the posterior density.

4. BINGHAM DISTRIBUTION

In this section, we lay out the Bingham distribution and the fundamental operations that we use to develop the filter and discuss its relation to several other distributions. The Bingham distribution on the hypersphere naturally appears when a d-dimensional normal random vector $\underline{x}$ with $\mathrm{E}(\underline{x}) = \underline{0}$ is conditioned on $\|\underline{x}\| = 1$ [26]. One of the main challenges when dealing with the Bingham distribution is the calculation of its normalization constant, so we discuss this issue in some detail.

4.1. Probability Density Function

As a consequence of the motivation above, it can be seen that the Bingham probability density function (pdf) looks exactly like its Gaussian counterpart except for the normalization constant. Furthermore, the parameter matrix of the Bingham distribution appearing in the exponential (which is the inverse covariance matrix in the Gaussian case) is usually decomposed into an orthogonal and a diagonal matrix, which yields an intuitive interpretation of the matrices. This results in the following definition.

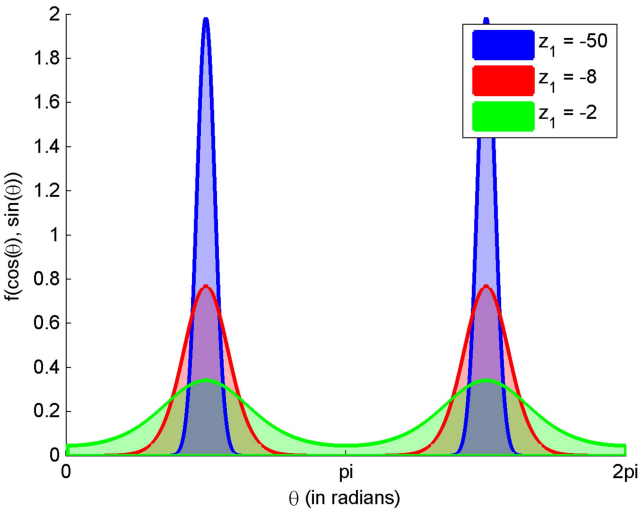


Fig. 2. Bingham probability density function with $\mathbf{M} = \mathbf{I}_{2\times 2}$ for different values of $\mathbf{Z} = \mathrm{diag}(z_1, 0)$ and $\underline{x} = (\cos(\theta), \sin(\theta))^T$. These values for z_1 correspond to standard deviations of approximately 6°, 16°, and 36°, respectively.

DEFINITION 1 Let $S_{d-1} = \{\underline{x} \in \mathbb{R}^d : \|\underline{x}\| = 1\} \subset \mathbb{R}^d$ be the unit hypersphere in $\mathbb{R}^d$. The probability density function (pdf)

$$f : S_{d-1} \to \mathbb{R} \tag{1}$$

of a Bingham distribution [5] is given by

$$f(\underline{x}) = \frac{1}{F} \cdot \exp(\underline{x}^T \mathbf{M}\mathbf{Z}\mathbf{M}^T \underline{x}), \tag{2}$$

where $\mathbf{M} \in \mathbb{R}^{d\times d}$ is an orthogonal matrix[1] describing the orientation, $\mathbf{Z} = \mathrm{diag}(z_1, \ldots z_{d-1}, 0) \in \mathbb{R}^{d\times d}$ with $z_1 \leq \cdots \leq z_{d-1} \leq 0$ is the concentration matrix, and F is a normalization constant.

As Bingham showed [5], adding a multiple of the identity matrix $\mathbf{I}_{d\times d}$ to $\mathbf{Z}$ does not change the distribution. Thus, we conveniently force the last entry of $\mathbf{Z}$ to be zero. Because it is possible to swap columns of $\mathbf{M}$ and the according diagonal entries in $\mathbf{Z}$ without changing the distribution, we can enforce $z_1 \leq \cdots \leq z_{d-1}$.

The probability density function is antipodally symmetric, i.e., $f(\underline{x}) = f(-\underline{x})$ holds for all $\underline{x} \in S_{d-1}$. Consequently, the Bingham distribution is invariant to rotations by 180°. Examples for two dimensions ($d = 2$) are shown in Fig. 1 and Fig. 2. Examples for three dimensions ($d = 3$) are shown in Fig. 3. The relation of the Bingham distribution to certain other distributions is discussed the appendix.

It deserves to mention that some authors use slightly different parameterizations of the Bingham distribution. In particular, the rightmost column of $\mathbf{M}$ is sometimes omitted [11], because it is, up to sign, uniquely determined by being a unit vector that is orthogonal to the

[1] An orthogonal matrix $\mathbf{M}$ fulfills the equation $\mathbf{M}\mathbf{M}^T = \mathbf{M}^T\mathbf{M} = \mathbf{I}_{d\times d}$.

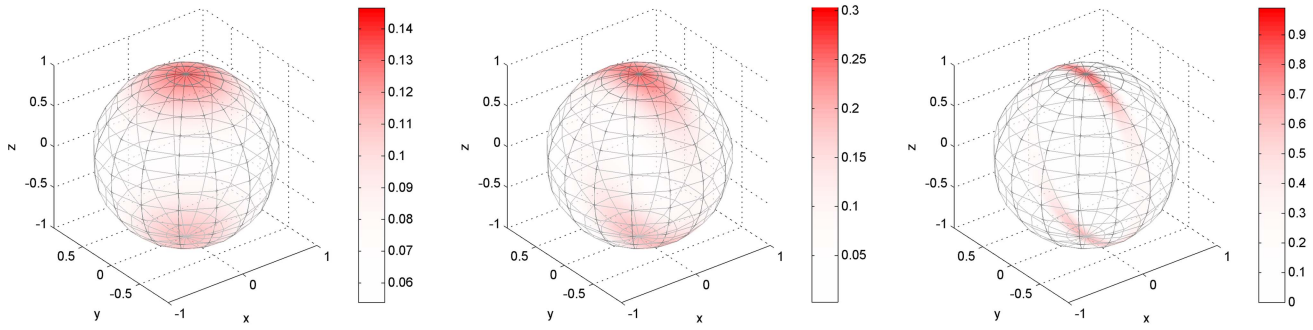


Fig. 3. Bingham pdf with $\mathbf{M} = \mathbf{I}_{3\times 3}$ for values of $\mathbf{Z} = \mathrm{diag}(-1,-1,0)$, $\mathbf{Z} = \mathrm{diag}(-5,-1,0)$, and $\mathbf{Z} = \mathrm{diag}(-50,-1,0)$.

other columns of $\mathbf{M}$. As a result of antipodal symmetry, the sign can safely be ignored. Still, we prefer to include the entire matrix $\mathbf{M}$ because this representation allows us to obtain the mode of the distribution very easily by taking the last column of $\mathbf{M}$.

4.2. Normalization Constant

The normalization constant of the Bingham distribution is difficult to calculate, which constitutes one of the most significant challenges when dealing with the Bingham distribution. Because

$$F = \int_{S_{d-1}} \exp(\underline{x}^T \mathbf{M}\mathbf{Z}\mathbf{M}^T \underline{x}) d\underline{x} \tag{3}$$

$$= \int_{S_{d-1}} \exp(\underline{x}^T \mathbf{Z}\underline{x}) d\underline{x}, \tag{4}$$

the normalization constant does not depend on $\mathbf{M}$. It can be calculated with the help of the hypergeometric function of a matrix argument [14], [23], [40] according to

$$F := |S_{d-1}| \cdot {}_1F_1\left(\frac{1}{2}, \frac{d}{2}, \mathbf{Z}\right), \tag{5}$$

where

$$|S_{d-1}| = \frac{2 \cdot \pi^{n/2}}{\Gamma(N/2)} \tag{6}$$

is the surface area of the d-sphere and ${}_1F_1(\cdot,\cdot,\cdot)$ is the hypergeometric function of matrix argument. In the d-dimensional case, this reduces to

$$F = |S_{d-1}| \cdot {}_1F_1\left(\frac{1}{2}, \frac{d}{2}, \begin{bmatrix} z_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & z_{d-1} & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix}\right) \tag{7}$$

$$= |S_{d-1}| \cdot {}_1F_1\left(\frac{1}{2}, \frac{d}{2}, \begin{bmatrix} z_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & z_{d-1} \end{bmatrix}\right), \tag{8}$$

so it is sufficient to compute the hypergeometric function a diagonal matrix of size $(d-1)\times(d-1)$. If $d = 2$, this is a hypergeometric function of a scalar argument, which is described in [1]. We will later show how to further reduce this to a Bessel function for $d = 2$.

A number of algorithms for computing the hypergeometric function have been proposed, for example saddle point-approximations [26], a series of Jack functions [23], and holonomic gradient descent [24]. Glover has suggested the formula [11, (8)]

$$F = 2\sqrt{\pi} \sum_{\alpha_1=0}^{\infty} \cdots \sum_{\alpha_{d-1}=0}^{\infty} \frac{\prod_{i=1}^{d-1} \Gamma\left(\alpha_i + \frac{1}{2}\right) \frac{z_i^{\alpha_i}}{\alpha_i!}}{\Gamma\left(\frac{d}{2} + \sum_{i=1}^{d-1} \alpha_i\right)} \tag{9}$$

which should only be evaluated for positive $z_1, \ldots, z_{d-1}$ to avoid a numerically unstable alternating series.[2] Because of the computational complexity involved, `libbingham` [9] provides a precomputed lookup table and linear interpolation is used at runtime to obtain an approximate value. The technique of precomputed tables has previously been used by Mardia et al. for the maximum likelihood estimate, which involves the normalization constant [38].

To allow for online calculation of the normalization constant, we use Bessel functions for $d = 2$ and the saddle-point approximation by Kume et al. [26] for $d > 2$. The derivatives of the normalization constant, which are required for the maximum likelihood estimation procedure, can be calculated according to [27].

5. OPERATIONS ON THE BINGHAM DISTRIBUTION

In this section, we derive the formulas for multiplication of two Bingham probability density functions. Furthermore, we will present a method for computing the composition of two Bingham-distributed random variables, which is analogous to the addition of real random variables.

5.1. Multiplication

For two given Bingham densities, we want to obtain their product. This product is used for Bayesian inference involving Bingham densities. The result presented below yields a convenient way to calculate the product of Bingham densities.

[2] This can easily be achieved by adding a multiple of the identity matrix to the concentration matrix $\mathbf{Z}$.

LEMMA 1 *Bingham densities are closed under multiplication with renormalization.*

PROOF Consider two Bingham densities

$$f_1(\underline{x}) = F_1 \cdot \exp(\underline{x}^T \mathbf{M}_1 \mathbf{Z}_1 \mathbf{M}_1^T \underline{x}) \tag{10}$$

and

$$f_2(\underline{x}) = F_2 \cdot \exp(\underline{x}^T \mathbf{M}_2 \mathbf{Z}_2 \mathbf{M}_2^T \underline{x}). \tag{11}$$

Then

$$f_1(\underline{x}) \cdot f_2(\underline{x}) = F_1 F_2 \cdot \exp(\underline{x}^T (\underbrace{\mathbf{M}_1 \mathbf{Z}_1 \mathbf{M}_1^T + \mathbf{M}_2 \mathbf{Z}_2 \mathbf{M}_2^T}_{=:\mathbf{C}}) \underline{x})$$

$$\propto F \cdot \exp(\underline{x}^T \mathbf{M} \mathbf{Z} \mathbf{M}^T \underline{x}) \tag{12}$$

with F as the new normalization constant after renormalization, $\mathbf{M}$ are the unit eigenvectors of $\mathbf{C}$, $\mathbf{D}$ has the eigenvalues of $\mathbf{C}$ on the diagonal (sorted in ascending order) and $\mathbf{Z} = \mathbf{D} - \mathbf{D}_{dd}\mathbf{I}_{d\times d}$ where $\mathbf{D}_{dd}$ refers to the bottom right entry of $\mathbf{D}$, i.e., the largest eigenvalue.

5.2. Estimation of Bingham Distribution Parameters

Estimating parameters for the Bingham distribution is not only motivated by the need to estimate distribution parameters of the process noise. It also plays a crucial role in the prediction process when computing the composition of two Bingham random vectors and reapproximating a Bingham distribution. This procedure is based on matching covariance matrices. Be aware that although the Bingham distribution is only defined on S_{d-1}, we can still compute the covariance matrix of a Bingham-distributed random vector $\underline{x} \in S_{d-1}$ according to $\mathbf{S} = \mathrm{E}(\underline{x} \cdot \underline{x}^T)$ in $\mathbb{R}^d$. Thus, we will present both the computation of the covariance matrix of a Bingham distributed random vector and the computation of parameters for a Bingham distribution with a given covariance (which could correspond to an arbitrary distribution on the hypersphere).

The maximum-likelihood estimate for the parameters $(\mathbf{M}, \mathbf{Z})$ of a Bingham distribution can be obtained from given or empirical moments (in the case of given samples) as described in [5]. $\mathbf{M}$ can be obtained as the matrix of eigenvectors of the covariance $\mathbf{S}$ with eigenvalues $\omega_1 \leq \cdots \leq \omega_d$. In other words, $\mathbf{M}$ can be found as the eigendecomposition of

$$\mathbf{S} = \mathbf{M} \cdot \mathrm{diag}(\omega_1, \ldots, \omega_d) \cdot \mathbf{M}^T. \tag{13}$$

To calculate $\mathbf{Z}$, the equations

$$\frac{\frac{\partial}{\partial z_i} {}_1F_1\left(\frac{1}{2}, 1, \mathrm{diag}(z_1, \ldots, z_d)\right)}{{}_1F_1\left(\frac{1}{2}, 1, \mathrm{diag}(z_1, \ldots, z_d)\right)} = \omega_i, \quad i = 1, \ldots, d \tag{14}$$

have to be solved under the constraint $z_d = 0$, which is justified by the argumentation above and used to simplify the computation. The actual computation is performed numerically. In our case, the `fsolve` routine from Matlab was used, which utilizes a trust region method for solving nonlinear equations.

Conversely, for a given Bingham$(\mathbf{M}, \mathbf{Z})$-distributed random vector $\underline{x} \in S_{d-1}$, the covariance matrix can be calculated according to

$$\mathrm{E}(\underline{x} \cdot \underline{x}^T) = \mathbf{M} \cdot \mathrm{diag}(\omega_1, \ldots, \omega_d) \cdot \mathbf{M}^T \tag{15}$$

$$= \mathbf{M} \cdot \mathrm{diag}\left(\frac{1}{F}\frac{\partial F}{\partial z_1}, \ldots, \frac{1}{F}\frac{\partial F}{\partial z_d}\right) \cdot \mathbf{M}^T. \tag{16}$$

Thus, the underlying distribution parameters of a Bingham distributed random vector are uniquely defined by its covariance matrix and vice versa. However, it is important to note that this covariance matrix is usually not the same as the covariance matrix of a Gaussian random vector which was conditioned to one in order to obtain the Bingham distribution.

REMARK 1 For $d = 2$, there is an interesting relation of the covariance matrix to the circular (or trigonometric) moments

$$m_n = \int_0^{2\pi} \exp(inx) f(x) dx \in \mathbb{C}, \quad i^2 = -1 \tag{17}$$

that are commonly used for circular distributions. A Bingham distribution with $\mathbf{M} = \mathbf{I}_{2\times 2}$ and $\underline{x} = [\cos(\theta), \sin(\theta)]^T$ has covariance

$$\mathbf{S} = \begin{bmatrix} \omega_1 & 0 \\ 0 & \omega_2 \end{bmatrix} = \begin{bmatrix} \mathrm{E}(x_1^2) & 0 \\ 0 & \mathrm{E}(x_2^2) \end{bmatrix} \tag{18}$$

$$= \begin{bmatrix} \mathrm{E}(\cos(\theta)^2) & 0 \\ 0 & \mathrm{E}(\sin(\theta)^2) \end{bmatrix}, \tag{19}$$

i.e., $\omega_1 = \mathrm{Re}\, m_2$ and $\omega_2 = \mathrm{Im}\, m_2$.

5.3. Composition

Now, we want to derive the composition of Bingham distributed random vectors, which is the directional analogue to addition of random vectors in a linear space. Thus, the density of the random vector resulting from this operation is the directional analogue to the convolution in linear space. First, we define a composition of individual points on the hypersphere S_{d-1}, which we then use to derive the composition of Bingham distributed random vectors. We consider a composition function

$$\oplus : S_{d-1} \times S_{d-1} \to S_{d-1}, \tag{20}$$

where $\oplus$ has to be compatible with 180° degree symmetry, i.e.,

$$x \oplus y = \pm((-x) \oplus y) \tag{21}$$

$$= \pm(x \oplus (-y)) \tag{22}$$

$$= \pm((-x) \oplus (-y)) \tag{23}$$

for all $x, y \in S_{d-1}$. Furthermore, we require the quotient $(S_{d-1}/\{\pm 1\}, \oplus)$ to have an algebraic group structure. This guarantees associativity, the existence of an identity element, and the existence of inverse elements.

REMARK 2 It has been shown that the only hyperspheres admitting a topological group structure are S_0, S_1, and S_3 [39]. Because S_0 only consists of two elements, S_1 and S_3 (i.e., $d = 2$ and $d = 4$) are the only two relevant hyperspheres. This structure is necessary to obtain a suitable composition operation.

For this reason, we only consider the cases $d = 2$ and $d = 4$ from now on. These two cases are of practical interest as they conveniently allow the representation of two-dimensional axes and three-dimensional orientations via quaternions.

5.3.1. Two-dimensional case:

For $d = 2$, we interpret $S_1 \subset \mathbb{R}^2$ as elements in $\mathbb{C}$ of unit length, where the first dimension is the real part and the second dimension the imaginary part. In this interpretation, the Bingham distributions can be understood as a distribution on a subset of the complex plane, namely the unit circle.

DEFINITION 2 For $d = 2$, the composition function $\oplus$ is defined to be complex multiplication, i.e.,

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \oplus \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 y_1 - x_2 y_2 \\ x_1 y_2 + x_2 y_1 \end{bmatrix} \tag{24}$$

analogous to

$$\begin{aligned}(x_1 + ix_2) \cdot (y_1 + iy_2) = (x_1 y_1 - x_2 y_2) \\ + i(x_1 y_2 + x_2 y_1).\end{aligned} \tag{25}$$

Since we only consider unit vectors, the composition $\oplus$ is equivalent to adding the angles of both complex numbers when they are represented in polar form. The identity element is ± 1 and the inverse element for $(x_1, x_2)^T$ is the complex conjugate $\pm(x_1, -x_2)^T$.

Unfortunately, the Bingham distribution is not closed under this kind of composition. That is, the resulting random vector is no longer Bingham distributed (see Lemma 3). Thus, we propose a technique to approximate the composed random vector with a Bingham distribution. The composition of two Bingham distributions $f_{\mathbf{A}}$ and $f_{\mathbf{B}}$ is calculated by considering the composition of their covariance matrices $\mathbf{A}, \mathbf{B}$ and estimating the parameters of $f_{\mathbf{C}}$ based on the resulting covariance matrix. Composition of covariance matrices can be derived from the composition of random vectors. Note that since covariance matrices are always symmetric, we can ignore the bottom left entry in our notation and mark it with an asterisk.

LEMMA 2 *Let $f_{\mathbf{A}}$ and $f_{\mathbf{B}}$ be Bingham distributions with covariance matrices*

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ * & a_{22} \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ * & b_{22} \end{bmatrix}, \tag{26}$$

respectively. Let $\underline{x}, \underline{y} \in S_1 \subset \mathbb{R}^2$ be independent random vectors distributed according to $f_{\mathbf{A}}$ and $f_{\mathbf{B}}$. Then the covariance

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} \\ * & c_{22} \end{bmatrix} := \mathrm{Cov}(\underline{x} \oplus \underline{y}) \tag{27}$$

of the composition is given by

$$c_{11} = a_{11} b_{11} - 2a_{12} b_{12} + a_{22} b_{22}, \tag{28}$$

$$c_{12} = a_{11} b_{12} - a_{12} b_{22} + a_{12} b_{11} - a_{22} b_{12}, \tag{29}$$

$$c_{22} = a_{11} b_{22} + 2a_{12} b_{12} + a_{22} b_{11}. \tag{30}$$

PROOF See Appendix E.

Based on $\mathbf{C}$, the maximum likelihood estimate is used to obtain the parameters $\mathbf{M}$ and $\mathbf{Z}$ of the uniquely defined Bingham distribution with covariance $\mathbf{C}$ as described above. This computation can be done in an efficient way, even though the solution of the equation involving the hypergeometric function is not given in closed form. This does not present a limitation to the proposed algorithm, because there are many efficient ways for the computation of the confluent hypergeometric function of a scalar argument [34], [41].

5.3.2. Four-dimensional case:

In the four-dimensional case ($d = 4$), we interpret $S_3 \subset \mathbb{R}^4$ as unit quaternions in $\mathbb{H}$ [25]. A quaternion $\underline{q} = [q_1, q_2, q_3, q_4]^T$ consists of the real part q_1 and imaginary parts q_2, q_3, q_4. It is written as

$$\mathbf{q} = q_1 + q_2 i + q_3 j + q_4 k, \tag{31}$$

where $i^2 = j^2 = k^2 = ijk = -1$ are the imaginary units. A rotation in $SO(3)$ with rotation axis $[v_1, v_2, v_3]^T \in S_2$ and rotation angle $\phi \in [0, 2\pi)$ can be represented as the quaternion

$$\mathbf{q} = \cos(\phi/2) + \sin(\phi/2)(v_1 i + v_2 j + v_3 k) \tag{32}$$

and applied to a vector $\underline{w} = [w_1, w_2, w_3] \in \mathbb{R}^3$ according to

$$\mathbf{w}^{rot} = \mathbf{q}(0 + w_1 i + w_2 j + w_3 k)\bar{\mathbf{q}}. \tag{33}$$

Here, $\bar{\mathbf{q}} = q_1 - q_2 i - q_3 j - q_4 k$ denotes the conjugate of $\mathbf{q}$ and $\mathbf{w}^{rot}$ quaternion containing the rotated vector encoded as the factors of the quaternion basis elements i, j, and k.

DEFINITION 3 For $d = 4$, the composition function $\oplus$ is defined to be quaternion multiplication, i.e.,

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \oplus \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} x_1 y_1 - x_2 y_2 - x_3 y_3 - x_4 y_4 \\ x_1 y_2 + x_2 y_1 + x_3 y_4 - x_4 y_3 \\ x_1 y_3 - x_2 y_4 + x_3 y_1 + x_4 y_2 \\ x_1 y_4 + x_2 y_3 - x_3 y_2 + x_4 y_1 \end{bmatrix}, \tag{34}$$

sometimes also referred to as Hamilton product.

This definition corresponds to the composition of rotations. The identity element is $\pm[1,0,0,0]^T$ and the inverse element is given by the quaternion conjugate as given above.

LEMMA 3 *For all nontrivial hyperspheres that allow a topological group structure ($d = 2$ and $d = 4$, see Remark 2), the Bingham distribution is not closed under composition of random variables.*

PROOF We prove this Lemma by computing the true distribution of the (Hamilton-) product $\oplus$ of two Bingham distributed random vectors $\underline{x} \sim f_x(\cdot)$ and $\underline{y} \sim f_y(\cdot)$ with respective parameter matrices $\mathbf{M}_x$, $\mathbf{M}_y$, $\mathbf{Z}_x$, and $\mathbf{Z}_y$. The true density $f(\cdot)$ of $\underline{x} \oplus \underline{y}$ can be expressed in terms of the densities of $\underline{x}$ and $\underline{y}$ by

$$f(\underline{z}) = \int_{S_{d-1}} f_x(\underline{z} \oplus \underline{a}^{-1}) f_y(\underline{a}) d\underline{a}. \tag{35}$$

Inversion of unit quaternions and complex numbers of unit length can both be obtained by conjugation. Furthermore, complex numbers and quaternions can both be represented by matrices. This can be used to construct a matrix $\mathbf{Q}_z$ such that $\underline{z} \oplus \underline{a}^{-1} = \mathbf{Q}_z \underline{a}$. Thus, we obtain

$$f(\underline{z}) = \int_{S_{d-1}} f_x(\mathbf{Q}_z \underline{a}) f_y(\underline{a}) da \tag{36}$$

$$\propto \int_{S_{d-1}} \exp(\underline{a}^T \mathbf{Q}_z^T \mathbf{M}_x \mathbf{Z}_x \mathbf{M}_x^T \mathbf{Q}_z \underline{a} + \underline{a}^T \mathbf{M}_y \mathbf{Z}_y \mathbf{M}_y^T \underline{a}) d\underline{a} \tag{37}$$

$$\propto \int_{S_{d-1}} \exp(\underline{a}^T (\mathbf{Q}_z^T \mathbf{M}_x \mathbf{Z}_x \mathbf{M}_x^T \mathbf{Q}_z + \mathbf{M}_y \mathbf{Z}_y \mathbf{M}_y^T) \underline{a}) d\underline{a}.$$

Computation of the integral yields a rescaled hypergeometric function of matrix argument. Therefore, the random variable $\underline{x} \oplus \underline{y}$ does not follow a Bingham distribution.

LEMMA 4 *Let $f_{\mathbf{A}}$ and $f_{\mathbf{B}}$ be Bingham distributions with covariance matrices*

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ * & a_{22} & a_{23} & a_{24} \\ * & * & a_{33} & a_{34} \\ * & * & * & a_{44} \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ * & b_{22} & b_{23} & b_{24} \\ * & * & b_{33} & b_{34} \\ * & * & * & b_{44} \end{bmatrix},$$

respectively. Let $\underline{x}, \underline{y} \in S_3 \subset \mathbb{R}^4$ be independent random vectors distributed according to $f_{\mathbf{A}}$ and $f_{\mathbf{B}}$. Then the covariance matrix

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ * & c_{22} & c_{23} & c_{24} \\ * & * & c_{33} & c_{34} \\ * & * & * & c_{44} \end{bmatrix} = \mathrm{Cov}(\underline{x} \oplus \underline{y}) \tag{38}$$

of the composition is given by

$$c_{ij} = \mathrm{E}((\underline{x} \oplus \underline{y})_i \cdot (\underline{x} \oplus \underline{y})_j), \quad i, j = 1, \ldots, 4. \tag{39}$$

PROOF Analogous to Lemma 2. The complete formula for c_{ij} is given in [11, A.9.2].

6. FILTER IMPLEMENTATION

The techniques presented in the preceding section can be applied to derive a recursive filter based on the Bingham distribution. The system model is given by

$$\underline{x}_{k+1} = \underline{x}_k \oplus \underline{w}_k, \tag{40}$$

where $\underline{w}_k$ is Bingham-distributed noise. The measurement model is given by

$$\underline{z}_k = \underline{x}_k \oplus \underline{v}_k, \tag{41}$$

where $\underline{v}_k$ is Bingham-distributed noise and $\underline{x}_k$ is an uncertain Bingham-distributed system state. Intuitively, this means that both system and measurement model are the identity disturbed by Bingham-distributed noise. Note that the modes of the distributions of $\underline{w}_k$ and $\underline{v}_k$ can be chosen to include a constant offset. This can be thought of as a directional equivalent to nonzero noise in the linear setting. For example, the mode of $\underline{w}_k$ can be chosen such that it represents a known angular velocity or a given control input. Alternatively, to avoid dealing with nonzero-mean noise distributions, a rotation may be applied to $\underline{x}_k$ first and zero-mean noise added subsequently.

The predicted and estimated distributions at time k are described by their parameter matrices $(\mathbf{M}_k^p, \mathbf{Z}_k^p)$ and $(\mathbf{M}_k^e, \mathbf{Z}_k^e)$, respectively. The noise distributions at time k are described by $(\mathbf{M}_k^w, \mathbf{Z}_k^w)$ and $(\mathbf{M}_k^v, \mathbf{Z}_k^v)$.

6.1. Prediction Step

The prediction can be calculated with the Chapman-Kolmogorov-equation

$$f_p(\underline{x}_{k+1}) \tag{42}$$

$$= \int_{S_{d-1}} f(\underline{x}_{k+1} \mid \underline{x}_k) f_e(\underline{x}_k) d\underline{x}_k \tag{43}$$

$$= \int_{S_{d-1}} \int_{S_{d-1}} f(\underline{x}_{k+1}, \underline{w}_k \mid \underline{x}_k) d\underline{w}_k f_e(\underline{x}_k) d\underline{x}_k \tag{44}$$

$$= \int_{S_{d-1}} \int_{S_{d-1}} f(\underline{x}_{k+1} \mid \underline{w}_k, \underline{x}_k) f_w(\underline{w}_k) d\underline{w}_k f_e(\underline{x}_k) d\underline{x}_k \tag{45}$$

$$= \int_{S_{d-1}} \int_{S_{d-1}} \delta(\underline{w}_k - (\underline{x}_k^{-1} \oplus \underline{x}_{k+1})) f_w(\underline{w}_k) d\underline{w}_k f_e(\underline{x}_k) d\underline{x}_k$$

$$= \int_{S_{d-1}} f_w(\underline{x}_k^{-1} \oplus \underline{x}_{k+1}) f_e(\underline{x}_k) d\underline{x}_k. \tag{46}$$

This yields

$$(\mathbf{M}_{k+1}^p, \mathbf{Z}_{k+1}^p) = \mathrm{composition}((\mathbf{M}_k^e, \mathbf{Z}_k^e), (\mathbf{M}_k^w, \mathbf{Z}_k^w)), \tag{47}$$

which uses the previously introduced composition operation to disturb the estimate with the system noise.

ALGORITHM 1 *Algorithm for prediction step.*

Input: estimate $\mathbf{M}_k^e, \mathbf{Z}_k^e$, noise $\mathbf{M}_k^w, \mathbf{Z}_k^w$
Output: prediction $\mathbf{M}_{k+1}^p, \mathbf{Z}_{k+1}^p$
/* obtain covariance matrices $\mathbf{A}, \mathbf{B}$ */
$\mathbf{A} \leftarrow \mathbf{M}_k^e \cdot \operatorname{diag}\left(\frac{1}{F}\frac{\partial F}{\partial z_1}, \ldots, \frac{1}{F}\frac{\partial F}{\partial z_d}\right) \cdot (\mathbf{M}_k^e)^T$;
$\mathbf{B} \leftarrow \mathbf{M}_k^w \cdot \operatorname{diag}\left(\frac{1}{F}\frac{\partial F}{\partial z_1}, \ldots, \frac{1}{F}\frac{\partial F}{\partial z_d}\right) \cdot (\mathbf{M}_k^w)^T$;
/* obtain $\mathbf{C}$ with to Lemma 2 or 4 */
$c_{ij} \leftarrow \mathrm{E}((\underline{x} \oplus \underline{y})_i \cdot (\underline{x} \oplus \underline{y})_j)$, $i, j = 1, \ldots, d$;
$\mathbf{C} = (c_{ij})_{ij}$;
/* obtain $\mathbf{M}_{k+1}^p, \mathbf{Z}_{k+1}^p$ based on $\mathbf{C}$ */
$\mathbf{M}_{k+1}^p, \mathbf{Z}_{k+1}^p \leftarrow \mathrm{MLE}(\mathbf{C})$;

6.2. Measurement Update

Given a measurement $\hat{\underline{z}}_k$, we can calculate the updated density $\hat{f}$ of $\underline{x}_k$ given $\underline{z}_k$ from the density f_v of $\underline{v}_k$ and the prior density f_x of $\underline{x}_k$. This is performed using the transformation theorem for densities and Bayes' rule

$$\hat{f}(\underline{a}) \propto f_v(\underline{a}^{-1} \oplus \hat{\underline{z}}) \cdot f_x(\underline{a}). \tag{48}$$

First, we make use of the fact that negation corresponds to conjugation for quaternions and complex numbers of unit length. Thus, we have $\underline{a}^{-1} \oplus \hat{\underline{z}} = \mathbf{D}(\hat{\underline{z}}^{-1} \oplus \underline{a})$ with $\mathbf{D} = \operatorname{diag}(1, -1)$ for $d = 2$ and $\mathbf{D} = \operatorname{diag}(1, -1, -1, -1)$. As in our proof of Lemma 7, we can use a matrix representation $\mathbf{Q}_{\hat{\underline{z}}^{-1}}$ of $\hat{\underline{z}}^{-1}$ such that $\hat{\underline{z}}^{-1} \oplus \underline{a} = \mathbf{Q}_{\hat{\underline{z}}^{-1}} \underline{a}$. Thus, we obtain

$$f_v(\underline{a}^{-1} \oplus \hat{\underline{z}}) = f_v(\mathbf{D} \cdot \mathbf{Q}_{\hat{z}^{-1}} \underline{a}). \tag{49}$$

This yields

$$f_v(\mathbf{D} \cdot \mathbf{Q}_{\hat{\underline{z}}^{-1}} \cdot \underline{a}) \tag{50}$$

$$\propto \exp(\underline{a}^T \mathbf{Q}_{\hat{\underline{z}}^{-1}}^T \mathbf{D}^T \mathbf{M}_k^v \mathbf{Z}_k^v (\mathbf{M}_k^v)^T \mathbf{D} \mathbf{Q}_{\hat{\underline{z}}^{-1}} \underline{a}) \tag{51}$$

$$= \exp(\underline{a}^T \mathbf{Q}_{\hat{\underline{z}}} \mathbf{D} \mathbf{M}_k^v \mathbf{Z}_k^v (\mathbf{M}_k^v)^T \mathbf{D} \mathbf{Q}_{\hat{\underline{z}}^{-1}} \underline{a}). \tag{52}$$

The last identity is due to $\mathbf{D}^T = \mathbf{D}$ and the fact that the transpose of the usual matrix representations of complex numbers and quaternions corresponds to the representation of their conjugates.

Finally, the parameters of the resulting Bingham distribution are obtained by

$$(\mathbf{M}_k^e, \mathbf{Z}_k^e) = \text{multiply}((\mathbf{M}, \mathbf{Z}_k^v), (\mathbf{M}_k^p, \mathbf{Z}_k^p)) \tag{53}$$

with $\mathbf{M} = (\hat{\underline{z}} \oplus (\mathbf{D}\mathbf{M}_k^v))$, where $\oplus$ is evaluated for each column of $\mathbf{D}\mathbf{M}_k^v$ and "multiply" denotes the procedure outlined in Sec. 5.1. This operation can be performed solely on the Bingham parameters and does not involve the calculation of normalization constants (see Algorithm 2).

ALGORITHM 2 *Algorithm for update step.*

Input: prediction $\mathbf{M}_k^p, \mathbf{Z}_k^p$, noise $\mathbf{M}_k^v, \mathbf{Z}_k^v$, measurement $\hat{\underline{z}}_k$
Output: estimate $\mathbf{M}_k^e, \mathbf{Z}_k^e$
/* rotate noise according to measurement */
$\mathbf{M} \leftarrow \hat{\underline{z}} \oplus (\mathbf{D}\mathbf{M}_k^v)$;
/* multiply with prior distribution */
$(\mathbf{M}_k^e, \mathbf{Z}_k^e) \leftarrow \text{multiply}((\mathbf{M}, \mathbf{Z}_k^v), (\mathbf{M}_k^p, \mathbf{Z}_k^p))$;

7. EVALUATION

The proposed filter was evaluated in simulations for both the 2D and 4D cases. In this section, all angles are given in radians unless specified differently.

For comparison, we implemented modified Kalman filters with two- and four-dimensional state vectors [19]. In order to deal with axial estimates, we introduce two modifications:

1) We mirror the estimate $\hat{\underline{z}} \leftarrow -\hat{\underline{z}}$ if the angle between prediction and measurement $\angle(\underline{x}_k^p, \hat{\underline{z}}) > \pi/2$.
2) We normalize the estimate $\underline{x}_k^e$ after each update step $\underline{x}_k^e \leftarrow (\underline{x}_k^e / \|\underline{x}_k^e\|)$.

It should be noted that in two-dimensional scenarios, a comparison to a Kalman filter with a scalar state is also possible. We previously performed this simulation in [32] and showed that the Bingham filter is superior to Kalman filter with scalar state in the considered scenario.

7.1. Two-Dimensional Case

In our example, we consider the estimation of an axis in robotics. This could be the axis of a symmetric rotor blade or any robotic joint with 180° symmetry. We use the initial estimate with mode $(0,1)^T$

$$\mathbf{M}_0^e = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{Z}_0^e = \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix}, \tag{54}$$

the system noise with mode $(1,0)^T$

$$\mathbf{M}_k^w = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \mathbf{Z}_k^w = \begin{pmatrix} -200 & 0 \\ 0 & 0 \end{pmatrix}, \tag{55}$$

and the measurement noise with mode $(1,0)^T$

$$\mathbf{M}_k^v = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \mathbf{Z}_k^v = \begin{pmatrix} -2 & 0 \\ 0 & 0 \end{pmatrix}. \tag{56}$$

The true initial state is given by $(1,0)^T$, i.e., the initial estimate with mode $(0,1)^T$ is very poor.

To calculate the covariance matrices for the Kalman filter we fit a Gaussian to one of the two Bingham modes by means of numerical integration, i.e.,

$$C = \int_{\alpha_m - \pi/2}^{\alpha_m + \pi/2} f([\cos(\phi), \sin(\phi)]^T) \tag{57}$$

$$\cdot \begin{bmatrix} (\cos(\phi) - m_1)^2 & (\cos(\phi) - m_1)(\sin(\phi) - m_2) \\ * & (\sin(\phi) - m_2)^2 \end{bmatrix} d\phi,$$

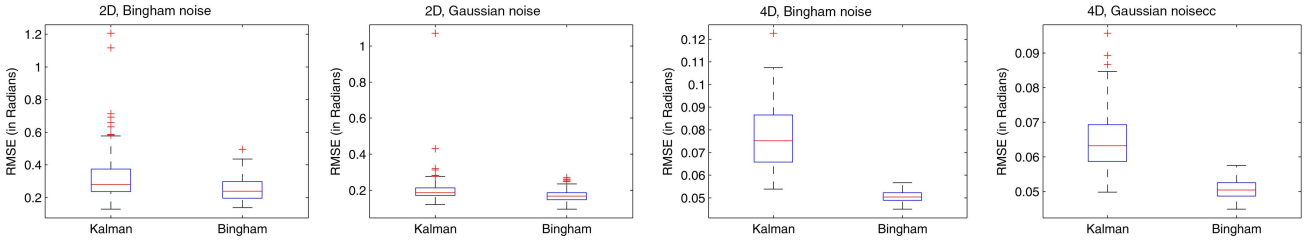


Fig. 4. RMSE from 100 Monte Carlo runs.

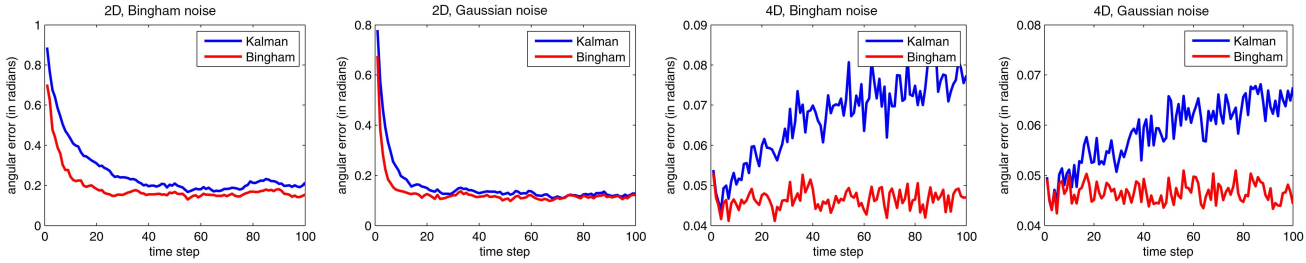


Fig. 5. Average error over time from 100 Monte Carlo runs.

where $(m_1, m_2)^T$ is a mode of the Bingham distribution and $\alpha_m = \text{atan2}(m_2, m_1)$. The original Bingham distribution and the resulting Gaussian are illustrated in Fig. 6. We obtain the parameters

$$C_0^e = \begin{bmatrix} 3.8 \times 10^{-1} & 0 \\ 0 & 1.5 \times 10{-}1 \end{bmatrix}, \tag{58}$$

$$C_k^w = \begin{bmatrix} 4.7 \times 10{-}6 & 0 \\ 0 & 2.5 \times 10^{-3} \end{bmatrix}, \tag{59}$$

$$C_k^v = \begin{bmatrix} 8.8 \times 10^{-2} & 0 \\ 0 & 2.8 \times 10^{-1} \end{bmatrix}, \tag{60}$$

which is equivalent to angular standard deviations of 43.9° for the first time step, 2.9° for the system noise and 36.3° for the measurement noise.

7.2. Four-Dimensional Case

For the quaternion case, we use the initial estimate with mode $(0,0,0,1)^T$

$$\mathbf{M}_0^e = \mathbf{I}_{4\times 4}, \quad \mathbf{Z}_0^e = \text{diag}(-1,-1,-1,0), \tag{61}$$

the system noise with mode $(1,0,0,0)^T$

$$\mathbf{M}_k^w = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \tag{62}$$

$$\mathbf{Z}_k^w = \text{diag}(-200,-200,-2,0), \tag{63}$$

and the measurement noise with mode $(1,0,0,0)^T$

$$\mathbf{M}_k^v = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \tag{64}$$

$$\mathbf{Z}_k^v = \text{diag}(-500,-500,-500,0). \tag{65}$$

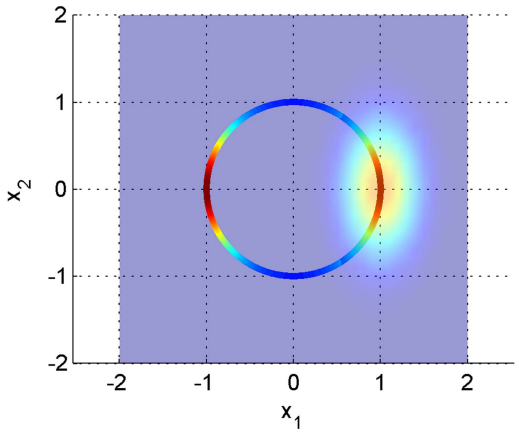


Fig. 6. The Bingham density with parameters $\mathbf{M}_k^v, \mathbf{Z}_k^v$ (on the circle) and a Gaussian (in the plane) fitted to one of the modes with the mean located at the mode and covariance computed according to (57).

The true initial state is $(0,1,0,0)^T$, i.e., the initial estimate with mode $(0,0,0,1)^T$ is very poor. It should be noted that the system noise is not isotropic, because the uncertainty is significantly higher in the third dimension than in the first two.

We converted the Bingham noise parameters to Gaussians analogous to the two-dimensional case.

7.3. Results

We simulate the system for a duration of $k_{\max} = 100$ time steps. For evaluation, we consider the angular RMSE given by

$$\sqrt{\frac{1}{k_{\max}} \sum_{k=1}^{k_{\max}} (e_k)^2} \tag{66}$$

with angular error

$$e_k = \min(\angle(\underline{x}_k^{\text{true}}, \text{mode}(\mathbf{M}_k^e)), \tag{67}$$

$$\pi - \angle(\underline{x}_k^{\text{true}}, \text{mode}(\mathbf{M}_k^e)) \tag{68}$$

at time step k. Obviously, $0 \leq e_k \leq \pi/2$ holds, which is consistent with our assumption of 180° symmetry. This error measure can be used in the two- and the four-dimensional setting. As we have shown in [8], the angle between two quaternions in four-dimensional space is proportional to the angle of the corresponding rotation between the two orientations in three dimensions, so e_k is a reasonable measure for quaternions.

The presented results are based on 100 Monte Carlo runs. Even though our filter is computationally more demanding than a Kalman filter, it is still fast enough for real-time applications. On a standard laptop with an Intel Core i7-2640M CPU, our non-optimized implementation in MATLAB needs approximately 8 ms for one time step (prediction and update) in the two-dimensional case. In the four-dimensional case, we implemented the hypergeometric function in C, but the maximum likelihood estimation is written in MATLAB. The calculations fore one time step require 13 ms on our laptop.

We consider two different types of noise, Bingham and Gaussian. Even though Bingham distributed noise may be a more realistic assumption in a circular setting, we do not want to give the proposed filter an unfair advantage by comparing it to a filter with an incorrect noise assumption. In the cases of Gaussian noise, we obtain the parameters of the Gaussian as described in (57) and convert the resulting Gaussians back to Bingham distributions to account for any information that was lost in the conversion from Bingham to Gaussian.

The results for all considered scenarios are depicted in Fig. 4 and Fig. 5. It can be seen that the proposed filter outperforms the Kalman filter in all considered scenarios. Particularly, it outperforms the Kalman filter even if Gaussian noise is used. This is due to the fact that projecting the Gaussian noise to the unit sphere does not yield a Gaussian distribution, which makes the Kalman filter suboptimal. Furthermore, the Kalman filter does not consider the nonlinearity of the underlying domain. As expected, the advantage of using the Bingham filter is larger if the noise is following a Bingham distribution.

8. CONCLUSION

We have presented a recursive filter based on the Bingham distribution. It can be applied to angular estimation in the plane with 180° symmetry and to quaternion-based estimation of orientation of objects in three-dimensional space. Thus, it is relevant for a wide area of applications, particularly when uncertainties occur, for example as a result of cheap sensors or very limited prior knowledge.

We have evaluated the proposed approaches in very challenging settings involving large non-isotropic noise. Our simulations have shown the superiority of the presented approach compared to the solution based on an adapted Kalman filter for both the circular and the quaternion case. This is true no matter if the noise is distributed according to a Bingham or a Gaussian distribution. Furthermore, we have shown that the proposed algorithms are fast enough on a typical laptop to be used in real-time applications.

Open challenges include an efficient estimator of the Bingham parameters based on available data. This makes an efficient evaluation of the confluent hypergeometric function necessary. Furthermore, extensions to nonlinear measurement equations and the group of rigid body motions $SE(3)$ may be of interest.

ACKNOWLEDGMENT

This work was partially supported by grants from the German Research Foundation (DFG) within the Research Training Groups RTG 1194 “Self-organizing Sensor-Actuator-Networks” and RTG 1126 “Soft-tissue Surgery: New Computer-based Methods for the Future Workplace.”

APPENDIX A. RELATION TO GAUSSIAN DISTRIBUTION

The Bingham distribution is closely related to the widely used Gaussian distribution.

DEFINITION 4 The pdf of a multivariate Gaussian distribution in $\mathbb{R}^d$ is given by

$$f^G(\underline{x}) = \frac{1}{\sqrt{(2\pi)^d \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\underline{x}-\underline{\mu})^T \boldsymbol{\Sigma}^{-1}(\underline{x}-\underline{\mu})\right)$$

with mean $\underline{\mu} \in \mathbb{R}^d$ and positive definite covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$.

If we require $\underline{\mu} = 0$ and restrict $\underline{x}$ to the unit hypersphere, i.e., $\|\underline{x}\| = 1$, we have

$$f^G(\underline{x}) = \frac{1}{\sqrt{(2\pi)^d \det \boldsymbol{\Sigma}}} \exp\left(\underline{x}^T \left(-\frac{1}{2}\boldsymbol{\Sigma}^{-1}\right)\underline{x}\right), \quad (69)$$

which is an unnormalized Bingham distribution with $\mathbf{M}\mathbf{Z}\mathbf{M}^T = -\frac{1}{2}\boldsymbol{\Sigma}^{-1}$. Conversely, any Bingham distribution is a restricted Gaussian distribution with $\boldsymbol{\Sigma} = (-2\mathbf{M}\mathbf{Z}\mathbf{M}^T)^{-1}$ if $\mathbf{M}\mathbf{Z}\mathbf{M}^T$ is negative definite. This condition can always be fulfilled by adding a multiple of the identity matrix $\mathbf{I}_{d\times d}$ to $\mathbf{Z}$. Modifying $\mathbf{Z}$ in this way yields a different Gaussian distribution, but the values on the unit hypersphere stay the same, i.e., the Bingham distribution does not change. A graphical illustration of the relation between a Gaussian density and the corresponding Bingham resulting from conditioning the original Gaussian random vector to unit length is given in Fig. 7.

Due to local linear structure of the underlying manifold, each mode of the Bingham distribution defined on this manifold is very similar to a Gaussian of dimension $d-1$ if and only if the uncertainty is small. This can be seen in Fig. 8, which shows the Kullback-Leibler

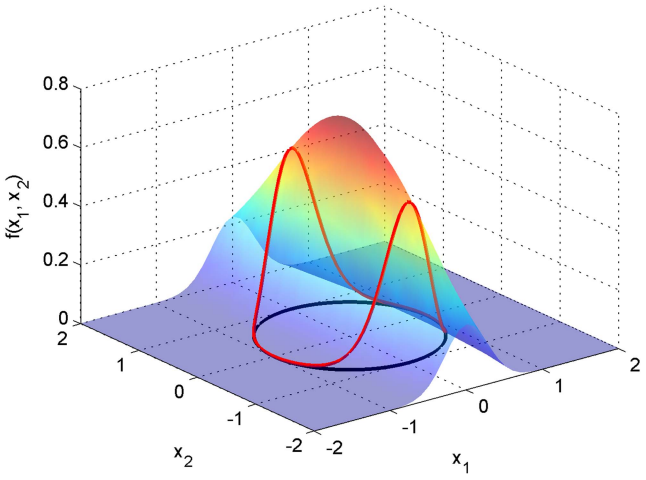


Fig. 7. A two-dimensional Gaussian distribution, which is restricted to the unit circle to obtain a two-dimensional Bingham distribution.

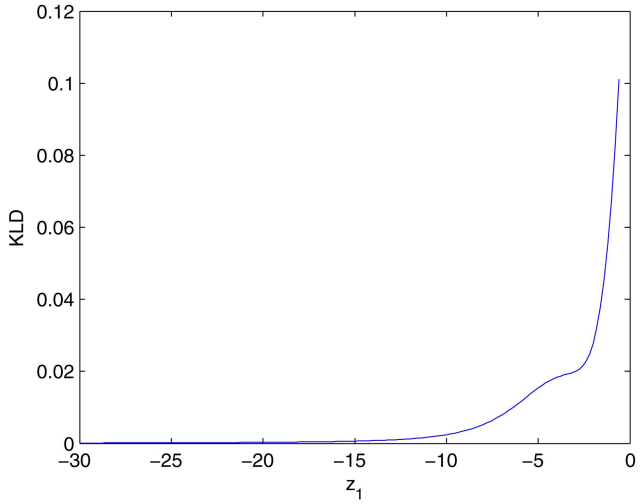


Fig. 8. Kullback-Leibler divergence on the interval $[0,\pi]$ between a Bingham pdf with $\mathbf{M} = \mathbf{I}_{2\times 2}$, $\mathbf{Z} = \mathrm{diag}(z_1, 0)$ and a Gaussian pdf with equal mode and standard deviation. For small uncertainties ($z_1 < -15$, which corresponds to a standard deviation of about 11°), the Gaussian and Bingham distributions are almost indistinguishable. However, for large uncertainties, the Gaussian approximation becomes quite poor.

divergence

$$\int_0^{\pi} f([\cos(\theta),\sin(\theta)]^T) \log\left(\frac{f([\cos(\theta),\sin(\theta)]^T)}{f^G(\theta,\mu,\sigma)}\right) d\theta \tag{70}$$

between one mode of a Bingham pdf for $d = 2$ and a corresponding one-dimensional Gaussian pdf on the semicircle.

APPENDIX B. RELATION TO VON MISES DISTRIBUTION

The Bingham distribution for $d = 2$ is closely related to the von Mises distribution. We can exploit this fact at some points in this paper.

DEFINITION 5 A von Mises distribution [15] is given by the probability density function

$$f^{\mathrm{VM}}(\phi;\mu,\kappa) = \frac{1}{2\pi I_0(\kappa)} \exp(\kappa\cos(\phi-\mu)) \tag{71}$$

for $\phi \in [0,2\pi)$, location parameter $\mu \in [0,2\pi)$ and concentration parameter $\kappa > 0$, where $I_0(\kappa)$ is the modified Bessel function [1] of order 0.

Based on this definition, we can show an interesting relation between Bingham and von Mises distributions [35].

LEMMA 5 *For the circular case, every Bingham density is equal to a von Mises density rescaled to* $[0,\pi)$ *and repeated on* $[\pi,2\pi)$.

PROOF We can reparameterize a Bingham distribution with $d = 2$ by substituting $\underline{x} = [\cos(\theta),\sin(\theta)]^T$ and

$$\mathbf{M} = \begin{bmatrix} -\sin(\nu) & \cos(\nu) \\ \cos(\nu) & \sin(\nu) \end{bmatrix}, \quad \mathbf{Z} = \begin{bmatrix} z_1 & 0 \\ 0 & 0 \end{bmatrix} \tag{72}$$

to attain the von Mises distribution. With

$$\mathbf{MZM}^T = z_1 \begin{bmatrix} \sin^2(\nu) & -\cos(\nu)\sin(\nu) \\ -\cos(\nu)\sin(\nu) & \cos^2(\nu) \end{bmatrix}, \tag{73}$$

this yields the pdf

$$\begin{aligned} f(\theta) &= \frac{1}{F}\exp([\cos(\theta),\sin(\theta)]\mathbf{MZM}^T[\cos(\theta),\sin(\theta)]^T) \\ &= \frac{1}{F}\exp(z_1(\cos(\theta)\sin(\nu)-\sin(\theta)\cos(\nu))^2) \qquad (74) \\ &= \frac{1}{F}\exp(z_1\sin^2(\theta-\nu))) \qquad (75) \end{aligned}$$

according to $\sin(a-b) = \sin(a)\cos(b) - \cos(a)\sin(b)$. Now we apply $\sin^2(a) = \frac{1}{2}(1-\cos(2a))$ and get

$$f(\theta) = \frac{1}{F}\exp\left(\frac{z_1}{2}\right)\exp\left(-\frac{z_1}{2}\cos(2\theta-2\nu)\right), \tag{76}$$

which exactly matches a von Mises distribution with $\phi = 2\theta, \mu = 2\nu$, and $\kappa = -z_1/2$ that has been repeated twice, i.e., $\theta \in [0,2\pi)$ and $\phi \in [0,4\pi)$.

This property can be exploited to derive a formula for the normalization constant of the Bingham distribution.

LEMMA 6 *For* $d = 2$, *the normalization constant is given by*

$$F = 2\pi \cdot I_0\left(\frac{z_1}{2}\right)\exp\left(\frac{z_1}{2}\right) \tag{77}$$

with derivatives

$$\frac{\partial}{\partial z_1}F = \pi\exp\left(\frac{z_1}{2}\right)\left(I_1\left(\frac{z_1}{2}\right)+I_0\left(\frac{z_1}{2}\right)\right) \tag{78}$$

$$\frac{\partial}{\partial z_2}F = \pi\exp\left(\frac{z_1+z_2}{2}\right) \cdot\left(I_0\left(\frac{z_1-z_2}{2}\right)-I_1\left(\frac{z_1-z_2}{2}\right)\right) \tag{79}$$

PROOF In order to consider the derivative with respect to z_2, we first consider a Bingham density with arbitrary z_2, which yields

$$f(\underline{x}) = \frac{1}{F}\exp\left(\underline{x}^T\mathbf{M}\begin{bmatrix} z_1 & 0 \\ 0 & z_2 \end{bmatrix}\mathbf{M}^T\underline{x}\right) \tag{80}$$

$$= \frac{1}{F}\exp\left(\underline{x}^T\mathbf{M}\begin{bmatrix} z_1 - z_2 & 0 \\ 0 & 0 \end{bmatrix}\mathbf{M}^T\underline{x} + z_2 \cdot \underline{x}^T\mathbf{M}\mathbf{M}^T\underline{x}\right)$$

$$= \frac{\exp(z_2)}{F}\exp\left(\underline{x}^T\mathbf{M}\begin{bmatrix} z_1 - z_2 & 0 \\ 0 & 0 \end{bmatrix}\mathbf{M}^T\underline{x}\right). \tag{81}$$

We use the formula for the normalization constant of a von Mises distribution to obtain

$$\frac{\exp(z_2)}{F} = \frac{1}{2\pi I_0\left(\frac{z_1 - z_2}{2}\right)}. \tag{82}$$

Solving this equation for F and substituting $z_2 = 0$ shows (77). The derivatives are calculated with [1, eq. 9.6.27].

APPENDIX C. RELATION TO VON MISES-FISHER DISTRIBUTION

The von Mises-Fisher distribution is a hyperspherical generalization of the von Mises distribution.

DEFINITION 6 A von Mises-Fisher distribution [7] is given by the pdf

$$f^{\mathrm{VMF}}(\underline{x};\underline{\mu},\kappa) = C_d(\kappa)\exp(\kappa\underline{\mu}^T\underline{x}) \tag{83}$$

with

$$C_d(\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{p/2}I_{p/2-1}(\kappa)} \tag{84}$$

for $\underline{x} \in S_{d-1}$, location parameter $\underline{\mu} \in S_{d-1}$ and scalar concentration parameter $\kappa > 0$, where $I_n(\kappa)$ is the modified Bessel function [1] of order n.

Unlike the Bingham distribution, the von Mises-Fisher distribution is unimodal and not antipodally symmetric, but radially symmetric around the axis of $\underline{\mu}$. We note that by use of hyperspherical coordinates, we can reformulate the pdf of the von Mises-Fisher distribution as

$$f^{\mathrm{VMF}}(\cdot;\kappa) : [0,\pi] \to \mathbb{R}^+, \tag{85}$$

$$f^{\mathrm{VMF}}(\phi;\kappa) = C_d(\kappa)\exp(\kappa\cos(\phi))\sin^{d-1}(\phi), \tag{86}$$

$$\text{where } \phi = \angle(\underline{\mu},\underline{x}). \tag{87}$$

The term $\sin^{d-1}(\phi)$ arises as a volume-correcting term when the substitution rule is applied. Using this definition, we can show an interesting relation between certain Bingham distributions and the von Mises-Fisher distribution.

LEMMA 7 *For a Bingham distribution with $z_1 = \cdots = z_{d-1}$ with pdf $f(\cdot)$, we have the relation*

$$f^{VMF}(\theta;\kappa) = (2\cos(\theta))^{d-1} \cdot f(\theta) \tag{88}$$

to the von Mises-Fisher distribution.

PROOF We consider $\mathbf{Z} = \mathrm{diag}(z_1\ldots,z_1,0)$ and $\mathbf{M} = [\cdots \mid \underline{\mu}]$. From the Bingham pdf, we obtain

$$f(\underline{x}) = \frac{1}{F}\exp(\underline{x}^T\mathbf{M}\mathbf{Z}\mathbf{M}^T\underline{x}) \tag{89}$$

$$= \frac{1}{F}\exp(\underline{x}^T\mathbf{M}\mathrm{diag}(z_1\ldots,z_1,0)\mathbf{M}^T\underline{x}) \tag{90}$$

$$= \frac{1}{F}\exp(\underline{x}^T\mathbf{M}\mathrm{diag}(0\ldots,0,-z_1)\mathbf{M}^T\underline{x} + z_1\underline{x}^T\mathbf{M}\mathbf{M}^T\underline{x}) \tag{91}$$

$$= \frac{\exp(z_1)}{F}\exp(-z_1\underline{x}^T\mathbf{M}\mathrm{diag}(0\ldots,0,1)\mathbf{M}^T\underline{x}). \tag{92}$$

We use the fact that the last column of $\mathbf{M}$ contains the mode $\underline{\mu}$ and obtain

$$f(\underline{x}) = \frac{\exp(z_1)}{F}\exp(-z_1\underline{x}^T\underline{\mu}\underline{\mu}^T\underline{x}) \tag{93}$$

$$= \frac{\exp(z_1)}{F}\exp(-z_1(\underline{\mu}^T\underline{x})^2) \tag{94}$$

$$= \frac{\exp(z_1)}{F}\exp(-z_1(\cos(\angle(\underline{x},\underline{\mu}))^2). \tag{95}$$

By using the trigonometric identity $\cos^2(x) = (1 + \cos(2x))/2$, we obtain

$$f(\underline{x}) = \frac{\exp\left(\frac{z_1}{2}\right)}{F}\exp\left(-\frac{z_1}{2}\cos(2\angle(\underline{x},\underline{\mu}))\right). \tag{96}$$

Substitution of spherical coordinates as above yields the pdf $f : [0,\pi/2] \to \mathbb{R}^+$,

$$f(\theta) = \frac{\exp\left(\frac{z_1}{2}\right)}{F}\exp\left(-\frac{z_1}{2}\cos(2\theta)\right)\sin^{d-1}(\theta). \tag{97}$$

On the other hand, the von Mises-Fisher pdf can be stated as

$$f^{VMF}(\phi;\kappa) = C_d(\kappa)\exp(\kappa\cos(\phi))\sin^{d-1}(\phi). \tag{98}$$

We set $\kappa = -z_1/2$ and $\phi = 2\theta$, which yields

$$f^{VMF}(\theta;\kappa) \tag{99}$$

$$= C_d\left(-\frac{z_1}{2}\right)\exp\left(-\frac{z_1}{2}\cos(2\theta)\right)\sin^{d-1}(2\theta) \tag{100}$$

$$= \frac{\sin^{d-1}(2\theta)}{\sin^{d-1}(\theta)} \cdot f(\theta) = (2\cos(\theta))^{d-1} \cdot f(\theta). \tag{101}$$

This fact can be used to simplify the maximum likelihood estimation when the underlying samples are (or can be assumed to be) generated by an isotropic Bingham distribution, i.e., when the corresponding density is circularly symmetric around the modes If the samples are reweighted by a factor of $(2\cos(\theta))^{d-1}$ and their angle around the mean is doubled, a von Mises-Fisher maximum likelihood estimate can be performed

to obtain κ and subsequently z_1. This can be advantageous, because the maximum likelihood estimate for a von Mises-Fisher distribution is computationally less demanding than for the Bingham distribution [43].

APPENDIX D. RELATION TO KENT DISTRIBUTION

Furthermore, it should be noted that the d-dimensional Bingham distribution is a special case of the d-dimensional Kent distribution [20]. The Kent distribution is also commonly referred to as the Fisher-Bingham distribution because it is a generalization of both the von Mises-Fisher and the Bingham distribution.

DEFINITION 7 The pdf of the Kent distribution is given by

$$f(\underline{x}) \propto \exp(\kappa \underline{\mu}^T \underline{x} + \sum_{j=2}^{d} \beta_j (\underline{\gamma}_j^T \underline{x})^2), \tag{102}$$

where $\underline{x} \in S_{d-1}$, and $\underline{\mu} \in S_{d-1}$ is the location parameter, $\kappa \geq 0$ is the concentration around μ, the directions $\underline{\gamma}_2, \ldots, \underline{\gamma}_d \in S_{d-1}$ are orthogonal and have corresponding concentrations $\beta_2 \geq \cdots \geq \beta_d \in \mathbb{R}$.

It can be seen that for $\kappa = 0$, this yields a Bingham distribution. The vectors $\underline{\gamma}_2, \ldots, \underline{\gamma}_d$ correspond to the $\mathbf{M}$ matrix and the coefficients $\beta_2, \ldots, \beta_d$ correspond to the diagonal of the $\mathbf{Z}$ matrix. This fact allows the application of methods developed for the Kent distribution such as [24], [26] in conjunction with the Bingham distribution. For $\beta_2 = \cdots = \beta_d = 0$, the Kent distribution reduces to a von Mises-Fisher distribution.

APPENDIX E. PROOF OF LEMMA 2.

PROOF The covariance of the composition

$$\mathbf{C} = \mathrm{Cov}(\underline{x} \oplus \underline{y}) \tag{103}$$

$$= \mathrm{Cov}\left(\begin{pmatrix} x_1 y_1 - x_2 y_2 \\ x_1 y_2 + x_2 y_1 \end{pmatrix}\right) \tag{104}$$

$$= \begin{pmatrix} \mathrm{Var}(x_1 y_1 - x_2 y_2) & \mathrm{Cov}(x_1 y_1 - x_2 y_2, x_1 y_2 + x_2 y_1) \\ * & \mathrm{Var}(x_1 y_2 + x_2 y_1) \end{pmatrix}$$

can be obtained by calculating the matrix entries individually. For the first entry we get

$$c_{11} = \mathrm{Var}(x_1 y_1 - x_2 y_2) \tag{105}$$

$$= \mathrm{E}((x_1 y_1 - x_2 y_2)^2) - (\mathrm{E}(x_1 y_1 - x_2 y_2))^2 \tag{106}$$

$$= \mathrm{E}(x_1^2 y_1^2 - 2 x_1 y_1 x_2 y_2 + x_2^2 y_2^2) - (\mathrm{E}(x_1 y_1) - \mathrm{E}(x_2 y_2))^2 \tag{107}$$

$$= \mathrm{E}(x_1^2)\mathrm{E}(y_1^2) - 2\mathrm{E}(x_1 x_2)\mathrm{E}(y_1 y_2) + \mathrm{E}(x_2^2)\mathrm{E}(y_2^2) \tag{108}$$

$$- (\underbrace{\mathrm{E}(x_1)}_{0}\underbrace{\mathrm{E}(y_1)}_{0} - \underbrace{\mathrm{E}(x_2)}_{0}\underbrace{\mathrm{E}(y_2)}_{0})^2 \tag{109}$$

$$= a_{11} b_{11} - 2 a_{12} b_{12} + a_{22} b_{22}. \tag{110}$$

We use independence of $\underline{x}$ and $\underline{y}$ in (107), linearity of the expectation value in (108), and symmetry of the Bingham in (109). Analogously we calculate

$$c_{22} = \mathrm{Var}(x_1 y_2 - x_2 y_1) \tag{111}$$

$$= \mathrm{E}((x_1 y_2 - x_2 y_1)^2) - (\mathrm{E}(x_1 y_2 - x_2 y_1))^2 \tag{112}$$

$$= \mathrm{E}(x_1^2 y_2^2 - 2 x_1 y_1 x_2 y_2 + x_2^2 y_1^2) - (\mathrm{E}(x_1 y_2) - \mathrm{E}(x_2 y_1))^2 \tag{113}$$

$$= \mathrm{E}(x_1^2)\mathrm{E}(y_2^2) - 2\mathrm{E}(x_1 x_2)\mathrm{E}(y_1 y_2) + \mathrm{E}(x_2^2)\mathrm{E}(y_1^2) - (\underbrace{\mathrm{E}(x_1)}_{0}\underbrace{\mathrm{E}(y_2)}_{0} - \underbrace{\mathrm{E}(x_2)}_{0}\underbrace{\mathrm{E}(y_1)}_{0})^2 \tag{114}$$

$$= a_{11} b_{22} - 2 a_{12} b_{12} + a_{22} b_{11}. \tag{115}$$

The off-diagonal entry can be calculated similarly

$$c_{12} = \mathrm{Cov}(x_1 y_1 - x_2 y_2, x_1 y_2 + x_2 y_1) \tag{116}$$

$$= \mathrm{E}((x_1 y_1 - x_2 y_2) \cdot (x_1 y_2 + x_2 y_1)) - \mathrm{E}(x_1 y_1 - x_2 y_2) \cdot \mathrm{E}(x_1 y_2 + x_2 y_1) \tag{117}$$

$$= \mathrm{E}(x_1^2 y_1 y_2 - x_1 x_2 y_2^2 + x_1 x_2 y_1^2 - x_2^2 y_1 y_2) - (\mathrm{E}(x_1)\mathrm{E}(y_1) - \mathrm{E}(x_2)\mathrm{E}(y_2)) \cdot (\mathrm{E}(x_1)\mathrm{E}(y_2) + \mathrm{E}(x_2)\mathrm{E}(y_1)) \tag{118}$$

$$= a_{11} b_{12} - a_{12} b_{22} + a_{12} b_{11} - a_{22} b_{12}. \tag{119}$$

Because $\mathbf{C}$ is a symmetrical matrix, this concludes the proof of Lemma 2.

REFERENCES

[1] Abramowitz, M., and Stegun, I. A. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 10th ed. Dover, New York, 1972.

[2] Antone, M., and Teller, S. Scalable, Absolute Position Recovery for Omni-directional Image Networks. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* (2001), vol. 1, pp. I-398–I-405.

[3] Azmani, M., Reboul, S., Choquel, J.-B., and Benjelloun, M. A Recursive Fusion Filter for Angular Data. In *IEEE International Conference on Robotics and Biomimetics (ROBIO 2009)* (2009), pp. 882–887.

[4] Bingham, C. *Distributions on the Sphere and on the Projective Plane*. PhD thesis, Yale University, 1964.

[5] Bingham, C. An Antipodally Symmetric Distribution on the Sphere. *The Annals of Statistics 2*, 6 (Nov. 1974), 1201–1225.

[6] Fisher, N. I., Lewis, T., and Embleton, B. J. *Statistical Analysis of Spherical Data*. Cambridge University Press, 1987.

[7] Fisher, R. Dispersion on a Sphere. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences 217*, 1130 (1953), 295–305.

[8] Gilitschenski, I., Kurz, G., Julier, S. J., and Hanebeck, U. D. Unscented Orientation Estimation Based on the Bingham Distribution. *arXiv preprint: Systems and Control (cs.SY)* (2013).

[9] Glover, J.
libbingham
Bingham Statistics Library, 2013.

[10] Glover, J., Bradski, G., and Rusu, R.
Monte Carlo Pose Estimation with Quaternion Kernels and the Bingham Distribution.
In *Proceedings of Robotics: Science and Systems (RSS 2011)* (Los Angeles, USA, 2011).

[11] Glover, J., and Kaelbling, L. P.
Tracking 3-D Rotations with the Quaternion Bingham Filter.
Tech. rep., MIT, Mar. 2013.

[12] Glover, J., and Kaelbling, L. P.
Tracking the Spin on a Ping Pong Ball with the Quaternion Bingham Filter.
In *IEEE Conference on Robotics and Automation (ICRA)* (2014). To Appear.

[13] Hertzberg, C., Wagner, R., Frese, U., and Schröder, L.
Integrating Generic Sensor Fusion Algorithms with Sound State Representations Through Encapsulation of Manifolds.
Information Fusion 14, 1 (Jan. 2013), 57–77.

[14] Herz, C. S.
Bessel Functions of Matrix Argument.
Annals of Mathematics 61, 3 (1955), 474–523.

[15] Jammalamadaka, S. R., and Sengupta, A.
Topics in Circular Statistics.
World Scientific Pub Co Inc, 2001.

[16] Julier, S., and LaViola, J.
On Kalman Filtering With Nonlinear Equality Constraints.
IEEE Transactions on Signal Processing 55, 6 (2007), 2774–2784.

[17] Julier, S. J., and Uhlmann, J. K.
Unscented Filtering and Nonlinear Estimation.
Proceedings of the IEEE 92, 3 (Mar. 2004), 401–422.

[18] Jupp, P. E., and Mardia, K. V.
Maximum Likelihood Estimators for the Matrix Von Mises-Fisher and Bingham Distributions.
Annals of Statistics 7 (3) (1979), 599–606.

[19] Kalman, R. E.
A New Approach to Linear Filtering and Prediction Problems.
Transactions of the ASME Journal of Basic Engineering 82 (1960), 35–45.

[20] Kent, J. T.
The Fisher-Bingham Distribution on the Sphere.
Journal of the Royal Statistical Society. Series B (Methodological) 44, 1 (1982), 71–80.

[21] Kent, J. T.
Asymptotic Expansions for the Bingham Distribution.
Journal of the Royal Statistical Society. Series C (Applied Statistics) 36 (2) (1987), 139–144.

[22] Kent, J. T.
The Complex Bingham Distribution and Shape Analysis.
Journal of the Royal Statistical Society. Series B (Methodological) (1994), 285–299.

[23] Koev, P., and Edelman, A.
The Efficient Evaluation of the Hypergeometric Function of a Matrix Argument.
Math. Comp. 75 (2006), 833–846.

[24] Koyama, T., Nakayama, H., Nishiyama, K., and Takayama, N.
Holonomic Gradient Descent for the Fisher-Bingham Distribution on the d-dimensional Sphere.
ArXiv e-prints (Jan. 2012).

[25] Kuipers, J. B.
Quaternions and Rotation Sequences,
vol. 66. Princeton University Press, 2002.

[26] Kume, A., and Wood, A. T. A.
Saddlepoint Approximations for the Bingham and Fisher-Bingham Normalising Constants.
Biometrika 92, 2 (2005), 465–476.

[27] Kume, A., and Wood, A. T. A.
On the Derivatives of the Normalising Constant of the Bingham Distribution.
Statistics & Probability Letters 77, 8 (2007), 832–837.

[28] Kunze, K., and Schaeben, H.
The Bingham Distribution of Quaternions and Its Spherical Radon Transform in Texture Analysis.
Mathematical Geology 36 (2004), 917–943.

[29] Kurz, G., Faion, F., and Hanebeck, U. D.
Constrained Object Tracking on Compact One-dimensional Manifolds Based on Directional Statistics.
In *Proceedings of the Fourth IEEE GRSS International Conference on Indoor Positioning and Indoor Navigation (IPIN 2013)* (Montbeliard, France, Oct. 2013).

[30] Kurz, G., Gilitschenski, I., and Hanebeck, U. D.
Recursive Nonlinear Filtering for Angular Data Based on Circular Distributions.
In *Proceedings of the 2013 American Control Conference (ACC 2013)* (Washington D. C., USA, June 2013).

[31] Kurz, G., Gilitschenski, I., and Hanebeck, U. D.
Nonlinear Measurement Update for Estimation of Angular Systems Based on Circular Distributions.
In *Proceedings of the 2014 American Control Conference (ACC 2014)* (Portland, Oregon, USA, June 2014).

[32] Kurz, G., Gilitschenski, I., Julier, S. J., and Hanebeck, U. D.
Recursive Estimation of Orientation Based on the Bingham Distribution.
In *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)* (Istanbul, Turkey, July 2013).

[33] Love, J. J.
Encyclopedia of Geomagnetism & Paleomagnetism.
Springer, Dordrecht, The Netherlands, 2007, ch. Bingham statistics, pp. 45–47.

[34] Luke, Y.
Algorithms for the Computation of Mathematical Functions.
Computer science and applied mathematics. Academic Press, 1977.

[35] Mardia, K.
Characterizations of Directional Distributions.
In *A Modern Course on Statistical Distributions in Scientific Work*, vol. 17. Springer Netherlands, 1975, pp. 365–385.

[36] Mardia, K.
Directional Statistics in Geosciences.
Communications in Statistics—Theory and Methods 10, 15 (1981), 1523–1543.

[37] Mardia, K. V., and Jupp, P. E.
Directional Statistics,
1 ed. Wiley, 1999.

[38] Mardia, K. V., and Zemroch, P. J.
Table of Maximum Likelihood Estimates for the Bingham Distribution.
Journal of Statistical Computation and Simulation 6, 1 (1977), 29–34.

[39] Megía, I. S.-M.
Which Spheres Admit a Topological Group Structure?
Revista de la Academia de Ciencias Exactas, Físico-Químicas y Naturales de Zaragoza 62 (2007), 75–79.

[40] Muirhead, R. J.
Aspects of Multivariate Statistical Theory.
Wiley, 1982.

[41] Muller, K. E.
Computing the Confluent Hypergeometric Function, M(a,b,x).
Numerische Mathematik 90, 1 (2001), 179–196.

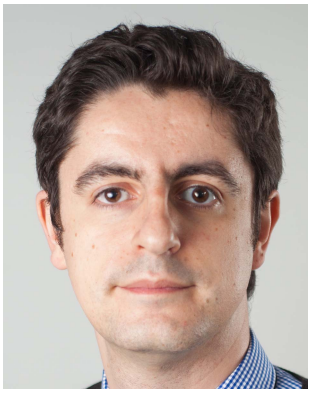
[42] Shiryaev, A. N.
Probability,
2nd ed. Springer, 1995.

[43] Sra, S.
A Short Note on Parameter Approximation for von Mises-Fisher Distributions: and a Fast Implementation of Is (x).
Computational Statistics 27, 1 (2012), 177–190.

[44] Stienne, G., Reboul, S., Azmani, M., Choquel, J., and Benjelloun, M.
A Multi-sensor Circular Particle Filter Applied to the Fusion of the GPS-L2C channels.
In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on* (2011).

[45] Stienne, G., Reboul, S., Azmani, M., Choquel, J., and Benjelloun, M.
A Multi-temporal Multi-sensor Circular Fusion Filter.
Information Fusion 18 (July 2013), 86–100.

Gerhard Kurz received his Dipl.-Inform. in computer science from the Karlsruhe Institute of Technology (KIT), Germany, in 2012. Currently, he is working towards a PhD degree at the Intelligent Sensor-Actuator-Systems Laboratory, Karlsruhe Institute of Technology (KIT), Germany. His research interests are in the field of medical data fusion, nonlinear estimation, and directional filtering.



Igor Gilitschenski is a research assistant at the Intelligent Sensor-Actuator-Systems Laboratory, Karlsruhe Institute of Technology (KIT), where he is working towards a PhD degree in computer science. Before joining the KIT, he obtained his diploma degree in mathematics (Dipl.-Math.) from the University of Stuttgart. His research interests include stochastic filtering theory and dynamic state estimation with a focus on nolinear systems and nonlinear domains.

Simon Julier is a Reader in Situational Awareness Systems in the Department of Computer Science. He received a DPhil in 1997 from the Robotics Research Group, Oxford University on the role of process models in navigation and estimation systems where he helped to developed the unscented Kalman filter and covariance intersection. He has worked extensively on mapping, localization, user interfaces, distributed data fusion and distributed multi-target tracking. These include a project to use multiple UAVs for search and rescue in wilderness environments, the development of distributed multi-target tracking algorithms (with the use of finite set statistics), agent-based simulation models for logistics planning in construction and the development of mobile augmented reality systems, with a major UK construction company, to visualize construction sequencing in a station in London. In 2013 he was elected a Distinguished Lecturer for the IEEE Society on Aerospace and Electronic Systems.

Uwe D. Hanebeck is a chaired professor of Computer Science at the Karlsruhe Institute of Technology (KIT) in Germany and director of the Intelligent Sensor-Actuator-Systems Laboratory (ISAS). Since 2005, he is the chairman of the Research Training Group RTG 1194 "Self-Organizing Sensor-Actuator-Networks" financed by the German Research Foundation.

Prof. Hanebeck obtained his Ph.D. degree in 1997 and his habilitation degree in 2003, both in Electrical Engineering from the Technical University in Munich, 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. Research results are applied to various application topics like localization, human-robot-interaction, assistive systems, sensor-actuator-networks, medical engineering, distributed measuring system, and extended range telepresence. Research is pursued in many academic projects and in a variety of cooperations with industrial partners.

Uwe D. Hanebeck was the General Chair of the "2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)," Program Co-Chair of the "11th International Conference on Information Fusion (Fusion 2008)," Program Co-Chair of the "2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008)," Regional Program Co-Chair for Europe for the "2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)," and will be General Chair of the "19th International Conference on Information Fusion (Fusion 2016)." He is a Member of the Board of Directors of the International Society of Information Fusion (ISIF), Editor-in-chief of its Journal of Advances in Information Fusion (JAIF), and associate editor for the letter category of the IEEE Transactions on Aerospace and Electronic Systems (TAES). He is author and coauthor of more than 300 publications in various high-ranking journals and conferences.

LRKF Revisited: The Smart Sampling Kalman Filter (S^2KF)

JANNIK STEINBRING
UWE D. HANEBECK

An accurate Linear Regression Kalman Filter (LRKF) for nonlinear systems called Smart Sampling Kalman Filter (S^2KF) is introduced. In order to get a better understanding of this new filter, a general introduction to Nonlinear Kalman Filters based on statistical linearization and LRKFs is given. The S^2KF is based on a new low-discrepancy Dirac mixture approximation of Gaussian densities. This approximation comprises an arbitrary number of optimally and deterministically placed samples in the relevant regions of the state space, so that the filter resolution can be adapted to either achieve high-quality results or to meet computational constraints. The S^2KF contains the UKF with equally weighted samples as a special case when using the same amount of samples. With an increasing number of samples, the new filter converges to the (typically unfeasible) exact analytic statistical linearization. Hence, the S^2KF can be seen as the ultimate generalization of all LRKFs such as the UKF, sigma-point filters, higher-order variants etc., as it homogeneously covers the state space with a freely chosen number of samples. It is evaluated against state-of-the-art LRKFs by performing nonlinear prediction and extended target tracking.

Manuscript received January 17, 2014; revised July 2, 2014; released for publication October 13, 2014.

Authors' address: Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany (e-mail: jannik.steinbring@kit.edu, uwe.hanebeck@ieee.org).

I. INTRODUCTION

We consider estimating the hidden state of a discrete-time stochastic nonlinear dynamic system based on noisy measurements through Bayesian inference. This is an important problem in many fields of current research such as (extended) object and group tracking [1]–[6], human motion tracking [7], object shape estimation [5], [8], [9], robotics [10], or estimation of extrinsic camera parameters [11].

Bayesian inference is a versatile approach for performing state estimation, but in general one has to cope with complex, e.g., multi-modal or non-Gaussian, state and noise probability density functions, which prohibits almost always the derivation of closed-form solutions. Particle Filters [12], [13] are elaborate Bayesian estimation techniques that try to deal with and maintain particle approximations of such complex densities. However, their main drawbacks are the high computational effort due to large sample sets, the problem of sample degeneration, non-reproducible results, and the need for explicit likelihood functions.

Therefore, simplifications are required in order to derive more efficient but still powerful estimators. A common first step of simplification is to get rid of maintaining the true complex state density by approximating it as a single Gaussian. Estimators using this approximation are grouped into the class of Gaussian Filters. But even with this convenient state density, closed-form solutions for state prediction, and especially for incorporating newly received measurements into the state estimate, are rarely available. Hence, specific Gaussian Filters are needed that try to overcome this problem by delivering approximative solutions for the state prediction and measurement updates. Such filters are for example the Gaussian Particle Filter (GPF) [14] or the Progressive Gaussian Filter (PGF) [15], [16] that make use of sample representations of the occurring Gaussian densities. Nevertheless, these filters, and in particular their measurement updates, are still costly.

For that reason, a further common step is to simplify the measurement update by computing linear approximations of the nonlinear mapping between the hidden system state and the noisy measurements, that is, obtaining a linear relationship between them. Such linearizations can be computed in several ways, e.g., using Taylor series, as will be discussed below. A key result of this obtained linear relationship is the possibility to perform backward inference without an explicit likelihood function. Instead, the well-known Kalman Filter formulas can be used [17]. Consequently, these estimators are referred to as *Nonlinear Kalman Filters*, as the Kalman Filter is applied to nonlinear systems [18].

In case of an already linear system corrupted by additive Gaussian noise, no linearization is required and the resulting filter is identical to the Kalman Filter [17], which yields optimal estimation results in the sense of a Minimum Mean Square Error (MMSE) [2]. However,

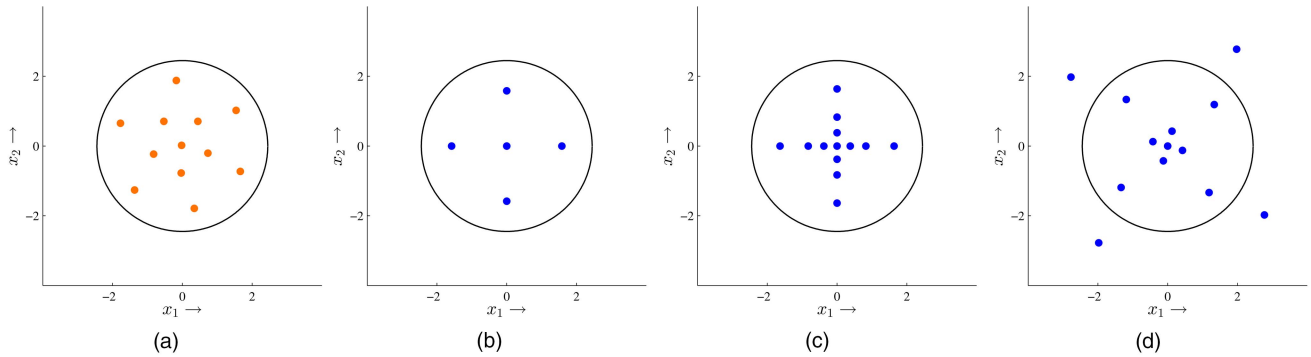


Fig. 1. Sampling of a two-dimensional standard normal distribution by the new S^{2}KF (orange points) and state-of-the-art LRKFs (blue points). Covariance matrices with confidence interval of 95% (black circles). (a) New S^{2}KF with 12 samples. (b) UKF with equal weights. (c) GF with 13 samples. (d) RUKF with 13 samples.

in case of nonlinear systems linearization is required which might be a strong simplification depending on the degree of the nonlinearity. The consequence is a diminished estimation performance compared to the more general Gaussian Filters not making use of such linearization. An option to mitigate linearization errors and therefore improve the estimation quality is to reduce the degree of nonlinearity by augmenting the actual nonlinear measurement model with additional, properly chosen mappings as proposed in [19].

One way to perform such linearization is *statistical linearization* [20], [21]. Basically, implementing a Nonlinear Kalman Filter based on statistical linearization only amounts to calculating the first- and second-order moments of (nonlinear) transformed densities, depending on the given system and measurement equations. For some equations, including polynomials, trigonometric functions, and their combinations, these moments can be calculated analytically [22]. Hence, this provides the filter based on statistical linearization with the best possible estimation quality and is referred to as analytic statistical linearization. However, this approach requires an individual treatment of each occurring equation which is time-consuming, error-prone, and prevents a generic filter applicable to any system and measurement equation, regardless of its complexity.

A widespread solution for these problems are sample-based approaches, where the occurring state and noise densities are represented as a set of samples, selected in a random or deterministic way. This allows to perform statistical linearization in the form of *statistical linear regression* [21], [23]. Nonlinear Kalman Filters making use of statistical linear regression are called Linear Regression Kalman Filters (LRKFs) [23], [24]. As a consequence of using samples instead of continuous densities, time and measurement updates have to be adapted in order to handle this density representation. On the one hand, the samples are propagated individually through the given system and measurement equations. On the other hand, occurring analytic moment calculations are turned into their sample-based counterparts, i.e., sample mean and sample covariance. Of course, this introduces a further approximation step (compared to the analytic statistical linearization) that may negatively affect the estimation performance. Nevertheless, employing an LRKF offers several advantages. First, due to the lack of an explicit use of a likelihood, the problem of sample degeneration is avoided,[1] and second, we obtain a generic filter that allows us to work with black box systems, e.g., systems given as (binary) programs, or to switch easily between different system and measurement models without any additional effort. Moreover, this facilitates filter design in the sense of rapid prototyping, as a newly designed system or measurement model can be tried out immediately.

Despite all these advantages of LRKFs, in order to improve overall estimation quality of filters based on statistical linearization, a mixture of analytic and sample-based moment calculation (semi-analytic approach) [22], [25] should be used whenever possible.

A. Contribution

In this paper, we introduce a new LRKF called the Smart Sampling Kalman Filter (S^{2}KF), which can be seen as the ultimate generalization of all LRKFs. For that purpose, we compute deterministic approximations of multivariate standard normal distributions comprising a predefined arbitrary number of optimally placed samples in the relevant regions of the state space (see Fig. 1(a)). These sets of deterministically chosen samples serve as the fundamental basis for the new filter. In contrast to approaches using non-deterministic sampling, this lets the filter compute reproducible results and is more efficient, as a much smaller amount of samples has to be employed.

By simply increasing the number of employed samples, the new filter converges to the analytic statisti-

[1]This is in contrast to filters explicitly using a likelihood, where backward inference implies a sample re-weighting that typically leads to a significantly reduced amount of samples contributing to the computation of the posterior moments, and consequently, to inaccurate results.

cal linearization as the resulting approximation of the standard normal distribution becomes more accurate. Moreover, this approach requires only a single and intuitive optimization parameter, i.e., the number of utilized samples. This makes filter fine-tuning simple, even for people not very familiar with sample-based Kalman filtering. Moreover, as opposed to state-of-the-art LRKFs, the number of samples are completely independent of the concrete dimension of the normal distribution and, hence, can be chosen freely. There are no restrictions such as a linear or exponential increase. This offers the possibility to (automatically) adapt the number of utilized samples for each time step individually depending on the concrete filtering problem, e.g., use less samples in situations of mild nonlinearities and more samples in case of high nonlinearities.

B. Related Work

One of the most popular LRKFs is the Unscented Kalman Filter (UKF) [26], [27]. It employs $2N + 1$ systematically chosen, axis-aligned samples, where N denotes the dimension of the standard normal distribution required for the time and measurement updates (see Fig. 1(b)). One of its greatest advantages is the ease with which the sample set can be created as well as the low computational effort due to the small and fixed amount of used samples. However, this property is also its main drawback. First, it is not possible to increase the number of samples for a concrete dimension N in order to improve the estimation quality. A consequence is that the even moments of a Gaussian greater than the second-order cannot precisely matched [27]. Second, the state space coverage suffers from the fact that the samples are placed solely on some principal axes.[2] Both of these factors have a negative impact on the estimation quality. And third, the small amount of employed samples makes it possible to compute non-positive definite covariance matrices and, thus, makes it hard for filtering applications to work reliably. For example, consider the nonlinear transformation of a Gaussian density using a sine-shaped function. If all samples of the Gaussian fall onto the zeros, the transformed Gaussian will be a density with zero variance [28].

Another drawback of the UKF is a rather unintuitive parameter that controls the sample spread and weighting. Besides the use of heuristics, maximum likelihood estimators can also be employed for determining these parameters. In [29], the authors select a limited set of possible values for the scaling parameter. During a filter step they perform an update for all selected scaling values individually and then choose the update that best matches the given measurement. Instead of simply trying various parameters during a filter step, the authors in [28] propose a parameter determination based on a Gaussian process optimization. Both approaches can improve the estimation quality, but also introduce new parameters (the possible scaling values and parameters controlling the optimization) that have to be determined in some way. Moreover, despite the additional computational effort due to the several computed updates for one filter step, the number of samples remains the same and, hence, the problems of insufficient state space coverage and non-positive definite covariance matrices are left unchanged.

A first step to improve the situation is done by the Gaussian Filter (GF) [30]. It enhances the sampling by deterministically placing an arbitrary number of samples on each principal axis (see Fig. 1(c)). Although the number of samples can easily be adjusted, which solves the problem of a fixed amount of samples and makes the covariance computation more reliable, the state space coverage still remains sparse due to the axes-only sample placement.

In order to overcome the problem of a sparse state space coverage, a non-deterministic sampling approach called Randomized Unscented Kalman Filter (RUKF) is introduced in [31]. Here, the moments for the time and measurement updates are calculated with the aid of an iterative stochastic integration rule, where each iteration uses an additional UKF sample set with random scaling and rotation (see Fig. 1(d)). In contrast to simple Gaussian random sampling, this guarantees that mean and covariance are always captured correctly. Furthermore, it is possible to create sample sets of arbitrary size with samples not only placed on the principal axes of the state space. Hence, the entire state space is covered in an adjustable manner. However, even though no complex parameters are required and the state space coverage is improved, this approach relies on the law of large numbers. Therefore, a large amount of samples is required to produce satisfying estimation results, particularly in larger state spaces. In addition, estimation results are not reproducible due to its non-deterministic nature. This is based on the fact that during each time and measurement update an individual set of samples is drawn randomly, which makes the filter outcome unpredictable.[3]

Table I summarizes the advantages and disadvantages of these state-of-the-art LRKFs and the new S^2KF. We emphasize that only the S^2KF is capable of producing reproducible results by using an arbitrary amount of samples placed in the entire state space (not only on the axes).

A completely different approach to compute the moments required by statistical linearization is to approximate the nonlinear system and measurement models with the aid of polynomials. That is, instead of the Gaus-

[2]Depending on the matrix square root method used for transforming the sample set to a non-standard Gaussian, i.e., the matrix factorization of the involved covariance matrices. For example, the Cholesky decomposition or the eigendecomposition.

[3]An option would be to reuse a single randomly generated sample set for all updates. However, this would conflict with the idea of random sampling, where sample sets representing the Gaussian distribution unfavorably are averaged out over time.

TABLE I
Comparison of important filter properties between state-of-the-art LRKFs and the new Smart Sampling Kalman Filter.

LRKF	**UKF**	**GF**	**RUKF**	**S^{2}KF**
Sample placement constraint	Axes	Axes	None	None
User-defined sample amount	No	Yes	Yes	Yes
Filter results	Reproducible	Reproducible	Not reproducible	Reproducible

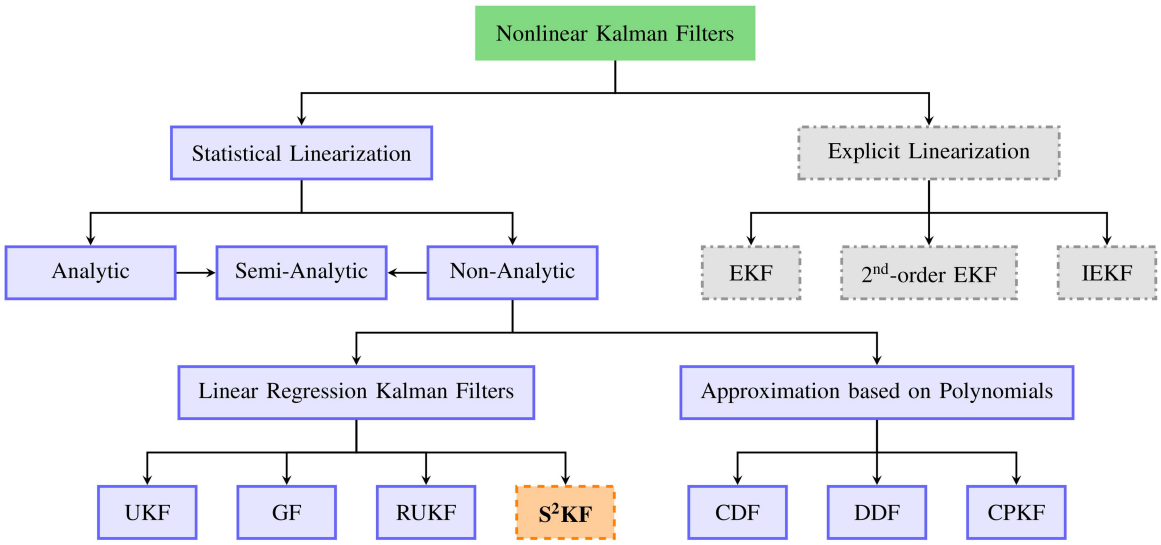


Fig. 2. Taxonomy of the discussed Nonlinear Kalman Filters relying on statistical linearization (solid blue), including the new S^{2}KF (dashed orange), and explicit linearization (dash-dotted gray).

sian distributions, the nonlinear models themselves get approximated. By doing so, formulas can be obtained that allow derivative-free and closed-form moment calculations which require only multiple evaluations of the nonlinear models. One option is to use polynomial interpolations for approximating the nonlinear models. Such filters are for example the Central Difference Filter (CDF) [32] and the Divided Difference Filter (DDF) [33]. Although these filters evaluate the nonlinear models at the same points as the UKF does, the filter results are different due to their different ways of computing the desired moments [21]. Another option is to approximate the nonlinear models by means of Chebyshev polynomials series expansion which results in the Chebyshev Polynomial Kalman Filter (CPKF) as proposed in [34]. Here, the actual polynomial approximation is obtained by using discrete cosine transformations. However, the proposed approach only works for a one-dimensional state space.

In contrast to statistical linearization, an explicit linearization of the system and measurement models based on Taylor series approximation is also possible. That is, the Kalman Filter formulas are still being used, only the type of linearization is changing. The Extended Kalman Filter (EKF) uses first-order Taylor expansions at the prior state mean for system and measurement model linearization, whereas its iterated version, the Iterated Extended Kalman Filter (IEKF), tries to improve estimation quality by finding a better point for linearization to take a given measurement into account [18]. Second-order variants of the EKF exist [18], [35], but the additional complexity has prohibited its widespread use [13]. One problem of this type of linearization is the need for explicit derivatives. In the best case, these can be taken analytically which unfortunately entails the same problems occurring in case of analytic statistical linearization: no easy exchange between different system and measurement models is possible, and it is time-consuming and error-prone. In all other cases, approximations of the derivatives will be inevitable. Another problem is that the linearization is only performed at a single point, that is, not the entire statistical information of the prior state estimate is taken into account during linearization. This typically leads to inferior estimation results compared to statistical linearization [22]. Moreover, this makes the filter also sensitive to the specific point used for the linearization, that is, to the prior state mean.

Fig. 2 shows a taxonomy of the above discussed Nonlinear Kalman Filters including the new S^{2}KF (dashed orange). It underlines the important difference between those filters relying on statistical linearization (solid blue) and the EKF variants using explicit linearization (dash-dotted gray). Additionally, all the fil-

ters in the bottom row try to achieve results as close as possible to the analytic statistical linearization.

C. Overview

The remainder of this paper is structured as follows. In Sec. II, we give a detailed formulation of the general Gaussian filtering problem using Bayesian inference. Then, in Sec. III, the class of Nonlinear Kalman Filters based on statistical linearization is presented. Sec. IV describes the transition from analytic statistical linearization to statistical linear regression in the form of the general LRKF approach. After that, in Sec. V, we describe how to compute optimal standard normal approximations using the idea of Localized Cumulative Distributions. Based on this, we introduce the new S^2KF. Extensive evaluation of the new filter is performed in Sec. VI. Finally, the conclusions are presented in Sec. VII.

II. PROBLEM FORMULATION

We consider estimating the hidden state $\underline{x}_k$ of a discrete-time stochastic nonlinear dynamic system based on noisy measurements $\tilde{\underline{y}}_k$.[4] The dynamic system is modeled by the system equation

$$\underline{x}_k = \underline{a}_k(\underline{x}_{k-1}, \underline{w}_k) \tag{1}$$

and the measurement equation

$$\underline{y}_k = \underline{h}_k(\underline{x}_k, \underline{v}_k), \tag{2}$$

where $\underline{y}_k$ denotes the measurement random variable from which the measurements $\tilde{\underline{y}}_k$ originate, and $\underline{w}_k$ as well as $\underline{v}_k$ Gaussian white noise. It is assumed that both noise processes are mutually independent and also independent of the system state. The system equation (1) models the temporal evolution of the system state, whereas the measurement equation (2) models the relation between the received noisy measurements $\tilde{\underline{y}}_k$ and the not direct observable system state $\underline{x}_k$.

We denote the probability density function (pdf) of the state at time step k conditioned on the k received measurements $\tilde{\underline{y}}_1, \ldots, \tilde{\underline{y}}_{k-1}, \tilde{\underline{y}}_k$ as

$$f_k^e(\underline{x}_k) = f(\underline{x}_k \mid \tilde{\underline{y}}_k, \tilde{\underline{y}}_{k-1}, \ldots, \tilde{\underline{y}}_1) = f(\underline{x}_k \mid \tilde{\underline{y}}_{k:1}), \tag{3}$$

and the predicted state density, i.e., the pdf of the state at time step k conditioned only on the measurements $\tilde{\underline{y}}_1, \ldots, \tilde{\underline{y}}_{k-2}, \tilde{\underline{y}}_{k-1}$, as

$$f_k^p(\underline{x}_k) = f(\underline{x}_k \mid \tilde{\underline{y}}_{k-1}, \tilde{\underline{y}}_{k-2}, \ldots, \tilde{\underline{y}}_1) = f(\underline{x}_k \mid \tilde{\underline{y}}_{k-1:1}). \tag{4}$$

The noise pdfs are given by

$$f_k^w(\underline{w}_k) = \mathcal{N}(\underline{w}_k; \hat{\underline{w}}_k, \mathbf{C}_k^w)$$

and

$$f_k^v(\underline{v}_k) = \mathcal{N}(\underline{v}_k; \hat{\underline{v}}_k, \mathbf{C}_k^v),$$

with means $\hat{\underline{w}}_k$ and $\hat{\underline{v}}_k$, and covariance matrices $\mathbf{C}_k^w$ and $\mathbf{C}_k^v$, respectively.

As computing the true conditional state pdfs (3) and (4) is intractable, our goal is to maintain Gaussian approximations of (3) and (4) recursively over time and incorporate new measurements by exploiting Bayes' rule. Such a recursive Bayesian estimator consists of two alternating steps, namely the time update and the measurement update.

A. Time Update

The objective of the time update, also called prediction step, is to propagate the last known Gaussian state estimate $f_{k-1}^e(\underline{x}_{k-1})$ (from the past) to the present by exploiting the given system model (1) in the form of its state-transition density $f_k^a(\underline{x}_k \mid \underline{x}_{k-1})$. This yields the predicted state estimate $f_k^p(\underline{x}_k)$ according to the Chapman-Kolomogorov equation [2]

$$\begin{aligned} f_k^p(\underline{x}_k) &= \int f_k^a(\underline{x}_k \mid \underline{x}_{k-1}) \cdot f_{k-1}^e(\underline{x}_{k-1}) d\underline{x}_{k-1} \\ &= \int\int \delta(\underline{x}_k - \underline{a}_k(\underline{x}_{k-1}, \underline{w}_k)) \\ &\quad \cdot f_{k-1}^e(\underline{x}_{k-1}) \cdot f_k^w(\underline{w}_k) d\underline{x}_{k-1} d\underline{w}_k, \end{aligned}$$

where $\delta(\cdot)$ denotes the Dirac delta function.

However, even though the state density $f_{k-1}^e(\underline{x}_{k-1})$ is Gaussian, this in general does not hold for the predicted state density $f_k^p(\underline{x}_k)$. Therefore, we have to perform a subsequent moment matching in order to fulfill our forced Gaussian state approximation. This is done by computing the predicted state mean

$$\begin{aligned} \hat{\underline{x}}_k^p &= \int \underline{x}_k \cdot f_k^p(\underline{x}_k) d\underline{x}_k \\ &= \int\int \underline{a}_k(\underline{x}_{k-1}, \underline{w}_k) \\ &\quad \cdot f_{k-1}^e(\underline{x}_{k-1}) \cdot f_k^w(\underline{w}_k) d\underline{x}_{k-1} d\underline{w}_k, \end{aligned} \tag{5}$$

and the predicted state covariance

$$\begin{aligned} \mathbf{C}_k^p &= \int (\underline{x}_k - \hat{\underline{x}}_k^p) \cdot (\underline{x}_k - \hat{\underline{x}}_k^p)^{\mathrm{T}} \cdot f_k^p(\underline{x}_k) d\underline{x}_k \\ &= \int\int \underline{a}_k(\underline{x}_{k-1}, \underline{w}_k) \cdot \underline{a}_k(\underline{x}_{k-1}, \underline{w}_k)^{\mathrm{T}} \\ &\quad \cdot f_{k-1}^e(\underline{x}_{k-1}) \cdot f_k^w(\underline{w}_k) d\underline{x}_{k-1} d\underline{w}_k - \hat{\underline{x}}_k^p \cdot (\hat{\underline{x}}_k^p)^{\mathrm{T}} \end{aligned} \tag{6}$$

of $f_k^p(\underline{x}_k)$, and finally approximating the predicted state density according to

$$f_k^p(\underline{x}_k) \approx \mathcal{N}(\underline{x}_k; \hat{\underline{x}}_k^p, \mathbf{C}_k^p). \tag{7}$$

This Gaussian state distribution will serve as basis for the measurement update.

B. Measurement Update

The measurement update or filter step incorporates a given measurement $\tilde{\underline{y}}_k$ at time step k into the predicted

[4] Vectors are underlined, matrices are printed in bold face, and the subscript k denotes the discrete time step.

state estimate (7) to correct it. This is done by using Bayes' rule. However, this requires that the measurement model (2) is turned into its corresponding likelihood function $f_k^h(\tilde{\underline{y}}_k \mid \underline{x}_k)$ by assuming that the current measurement $\tilde{\underline{y}}_k$ is conditionally independent of the already processed measurements $\tilde{\underline{y}}_{k-1:1}$ given the predicted state estimate. Then, the corrected state estimate can be obtained according to

$$f_k^e(\underline{x}_k) = \frac{f_k^h(\tilde{\underline{y}}_k \mid \underline{x}_k) \cdot f_k^p(\underline{x}_k)}{f_k^y(\tilde{\underline{y}}_k \mid \tilde{\underline{y}}_{k-1:1})}, \tag{8}$$

where $f_k^y(\tilde{\underline{y}}_k \mid \tilde{\underline{y}}_{k-1:1})$ is only a normalization constant. More precisely, the measurement distribution $f_k^y(\underline{y}_k \mid \tilde{\underline{y}}_{k-1:1})$ encodes how probable a distinct measurement is, given all prior received measurements $\tilde{\underline{y}}_{k-1:1}$. A concrete measurement $\tilde{\underline{y}}_k$ in turn is a realization of this distribution.

Equation (8) can be rewritten in the form of the joint density $f_k^{x,y}(\underline{x}_k, \underline{y}_k \mid \tilde{\underline{y}}_{k-1:1})$ of prior state and measurement as

$$f_k^e(\underline{x}_k) = \frac{f_k^{x,y}(\underline{x}_k, \tilde{\underline{y}}_k \mid \tilde{\underline{y}}_{k-1:1})}{f_k^y(\tilde{\underline{y}}_k \mid \tilde{\underline{y}}_{k-1:1})}. \tag{9}$$

Thus, the Bayesian measurement update can be interpreted as a given measurement $\tilde{\underline{y}}_k$ determining where to slice the joint density $f_k^{x,y}(\underline{x}_k, \underline{y}_k \mid \tilde{\underline{y}}_{k-1:1})$ in order to get the posterior state density $f_k^e(\underline{x}_k)$.

As with the predicted state density, the obtained posterior state density $f_k^e(\underline{x}_k)$ is not necessarily Gaussian. Consequently, the posterior state density also has to be reapproximated as a Gaussian by means of moment matching afterwards.

C. Bayesian Estimator

The alternating use of the introduced time and measurement updates, together with a given initial state estimate

$$f_0^e(\underline{x}_0) \approx \mathcal{N}(\underline{x}_0; \hat{\underline{x}}_0^e, \mathbf{C}_0^e)$$

with initial mean $\hat{\underline{x}}_0^e$ and initial covariance $\mathbf{C}_0^e$, yields the desired recursive state estimation in form of a Bayesian estimator. It is important to note that this estimator is a restricted variant of the general recursive Bayesian estimator, as we force the state distribution to be Gaussian all the time.

III. NONLINEAR KALMAN FILTERING BASED ON STATISTICAL LINEARIZATION

Although the estimator introduced in Sec. II is a much simpler variant of the general Bayesian estimator, its measurement update is still demanding. First, an explicit likelihood function is required, which is hard to derive in case of non-additive measurement noise. Second, even if one is at hand, it is still almost always impossible to compute the measurement update analytically.

However, as we already force the posterior state density $f_k^e(\underline{x}_k)$ to be Gaussian, the measurement update can be strongly simplified by additional approximating the joint density of prior state and measurement in (9) as a Gaussian, that is,

$$f_k^{x,y}(\underline{x}_k, \underline{y}_k \mid \tilde{\underline{y}}_{k-1:1}) \approx \mathcal{N}\left(\begin{bmatrix} \underline{x}_k \\ \underline{y}_k \end{bmatrix}; \begin{bmatrix} \hat{\underline{x}}_k^p \\ \hat{\underline{y}}_k \end{bmatrix}, \begin{bmatrix} \mathbf{C}_k^p & \mathbf{C}_k^{x,y} \\ (\mathbf{C}_k^{x,y})^{\mathrm{T}} & \mathbf{C}_k^y \end{bmatrix} \right), \tag{10}$$

where $\hat{\underline{y}}_k$ and $\mathbf{C}_k^y$ denote the measurement mean and covariance, and $\mathbf{C}_k^{x,y}$ the cross-covariance matrix of state and measurement. As a direct consequence of this simplification, the posterior state density becomes also Gaussian [2]

$$\begin{aligned} f_k^e(\underline{x}_k) &\approx \frac{\mathcal{N}\left(\begin{bmatrix} \underline{x}_k \\ \tilde{\underline{y}}_k \end{bmatrix}; \begin{bmatrix} \hat{\underline{x}}_k^p \\ \hat{\underline{y}}_k \end{bmatrix}, \begin{bmatrix} \mathbf{C}_k^p & \mathbf{C}_k^{x,y} \\ (\mathbf{C}_k^{x,y})^{\mathrm{T}} & \mathbf{C}_k^y \end{bmatrix} \right)}{f_k^y(\tilde{\underline{y}}_k \mid \tilde{\underline{y}}_{k-1:1})} \\ &= \mathcal{N}(\underline{x}_k; \hat{\underline{x}}_k^e, \mathbf{C}_k^e), \end{aligned} \tag{11}$$

with posterior mean

$$\hat{\underline{x}}_k^e = \hat{\underline{x}}_k^p + \mathbf{C}_k^{x,y} \cdot (\mathbf{C}_k^y)^{-1} \cdot (\tilde{\underline{y}}_k - \hat{\underline{y}}_k) \tag{12}$$

and covariance

$$\mathbf{C}_k^e = \mathbf{C}_k^p - \mathbf{C}_k^{x,y} \cdot (\mathbf{C}_k^y)^{-1} \cdot (\mathbf{C}_k^{x,y})^{\mathrm{T}}, \tag{13}$$

which in fact are the well-known Kalman Filter formulas [17]. Hence, an estimator that uses this measurement update is called Nonlinear Kalman Filter.

Fig. 3 illustrates the Nonlinear Kalman Filter measurement update in case of a scalar state and measurement. The exemplary Gaussian joint density of state and measurement $f^{x,y}(x, y)$ is depicted in Fig. 3(a) and is sliced by the given measurement $\tilde{y}$ in Fig. 3(b) to obtain the posterior state density $f^e(x)$. It should be noted that the variance of the state density $f^e(x)$, i.e., the uncertainty of the posterior state estimate, is smaller than the one of the prior state density $f^p(x)$ due to the existing correlation between prior state and measurement, that is, $\mathbf{C}^{x,y} \neq 0$ (a non-axis-aligned Gaussian joint density). Furthermore, the state mean also changes due to the additional difference between the expected measurement $\hat{y}$ and the given measurement $\tilde{y}$.

As a result of (10), the measurement distribution $f_k^y(\underline{y}_k \mid \tilde{\underline{y}}_{k-1:1})$ becomes a Gaussian, too (see Fig. 3(a)). By this means, the relation between prior state and measurement, i.e., the measurement model (2), gets *implicitly linearized*. This is a direct consequence of the fact that there exists always an equivalent linear transformation from the prior Gaussian state distribution to this Gaussian measurement distribution.

In order to obtain the posterior Gaussian density (11), that is, perform the measurement update, the three moments $\hat{\underline{y}}_k$, $\mathbf{C}_k^y$, and $\mathbf{C}_k^{x,y}$ are required. Based on the given measurement model $\underline{h}_k(\underline{x}_k, \underline{v}_k)$, measurement noise

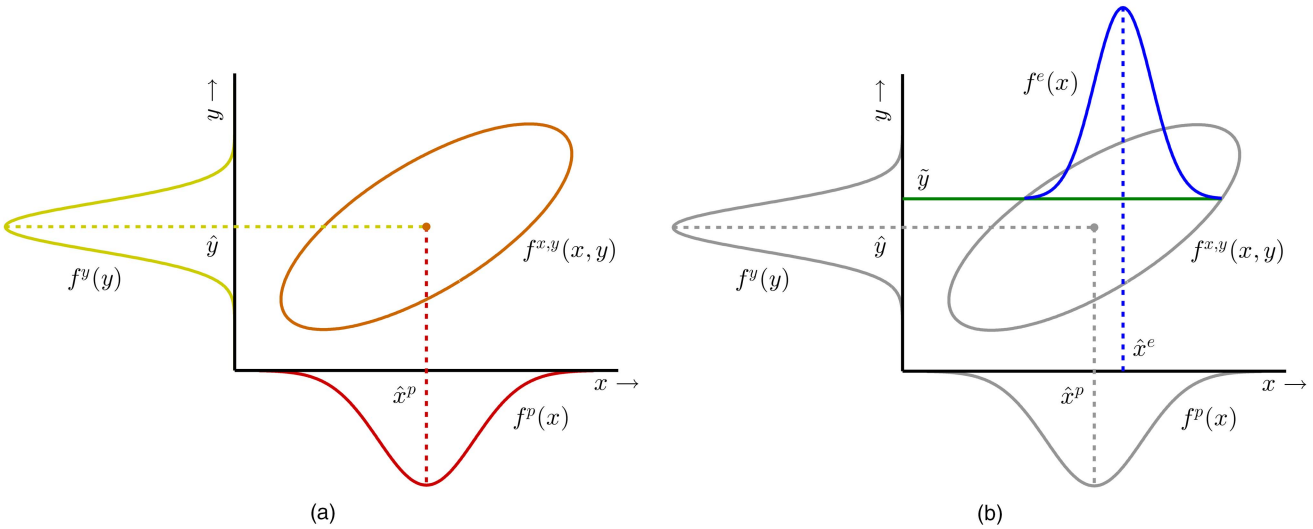


Fig. 3. Linearized measurement update in case of a scalar state x and measurement y. For readability, the time index k is omitted here. (a) Prior state density (red), measurement density (yellow), and Gaussian joint density (orange) of prior state and measurement. (b) The given measurement $\tilde{y}$ slices the joint density (green line) to obtain the posterior state density (blue).

density $f_k^v(\underline{v}_k)$, and predicted state density $f_k^p(\underline{x}_k)$, we can compute the measurement mean according to

$$\begin{aligned}\hat{\underline{y}}_k &= \int \underline{y}_k \cdot f_k^y(\underline{y}_k) d\underline{y}_k \\ &= \int\int \underline{h}_k(\underline{x}_k, \underline{v}_k) \cdot f_k^p(\underline{x}_k) \cdot f_k^v(\underline{v}_k) d\underline{x}_k d\underline{v}_k, \end{aligned} \quad (14)$$

the measurement covariance according to

$$\begin{aligned}\mathbf{C}_k^y &= \int (\underline{y}_k - \hat{\underline{y}}_k) \cdot (\underline{y}_k - \hat{\underline{y}}_k)^{\mathrm{T}} \cdot f_k^y(\underline{y}_k) d\underline{y}_k \\ &= \int\int \underline{h}_k(\underline{x}_k, \underline{v}_k) \cdot \underline{h}_k(\underline{x}_k, \underline{v}_k)^{\mathrm{T}} \\ &\quad \cdot f_k^p(\underline{x}_k) \cdot f_k^v(\underline{v}_k) d\underline{x}_k d\underline{v}_k - \hat{\underline{y}}_k \cdot \hat{\underline{y}}_k^{\mathrm{T}}, \end{aligned} \quad (15)$$

and the state measurement cross-covariance according to

$$\begin{aligned}\mathbf{C}_k^{x,y} &= \int\int (\underline{x}_k - \hat{\underline{x}}_k^p) \cdot (\underline{y}_k - \hat{\underline{y}}_k)^{\mathrm{T}} \\ &\quad \cdot f_k^{x,y}(\underline{x}_k, \underline{y}_k) d\underline{x}_k d\underline{y}_k \\ &= \int\int \underline{x}_k \cdot \underline{h}_k(\underline{x}_k, \underline{v}_k)^{\mathrm{T}} \\ &\quad \cdot f_k^p(\underline{x}_k) \cdot f_k^v(\underline{v}_k) d\underline{x}_k d\underline{v}_k - \hat{\underline{x}}_k^p \cdot \hat{\underline{y}}_k^{\mathrm{T}}, \end{aligned} \quad (16)$$

respectively. This moment calculation approach yields the so-called statistical linearization, as the implicit linearization of the measurement model takes the entire statistical information of the prior state estimate and the measurement noise into account. The result is a Nonlinear Kalman Filter based on statistical linearization. Such implicit linearization can be obtained in several ways, for example, by computing all the moments analytically or by computing them approximatively using samples.

One should keep in mind that this simplified measurement update comes at the expense of a diminished state estimation quality, depending on the degree of the concrete measurement model nonlinearity. In other words, the implicit linearization of a measurement model that is highly nonlinear around the prior state estimate can lead to large errors in the posterior state estimate compared to the unmodified measurement update introduced in Sec. II-B.

IV. THE LINEAR REGRESSION KALMAN FILTER

The Nonlinear Kalman Filter introduced in Sec. III requires the calculation of certain moments to perform time and measurement updates. Doing this analytically provides the Nonlinear Kalman Filter based on statistical linearization with the best possible estimation quality, and should be the means of choice whenever feasible. But, in case of non-existent closed-form solutions, or complicated system and measurement equations, approximative moment calculations have to be performed.

One way to achieve this is to replace the occurring state and noise densities with proper *Dirac mixture* densities, that is, sample-based density representations. This turns the statistical linearization into an approximate statistical linear regression. Consequently, all Nonlinear Kalman Filters using this technique, regardless of whether random or deterministic sampling is used, fall in the class of Linear Regression Kalman Filters.

As only a limited number of samples can be used, this approach always entails a *density approximation*. Therefore, Linear Regression Kalman Filters possess an, in general, inferior estimation quality compared to Nonlinear Kalman Filters based on analytic statistical linearization. Nevertheless, these filters are still efficient and, as no analytic moment calculation is required, are much easier to use.

A. Dirac Mixtures

A Dirac mixture approximation of an arbitrary density function $f_k(\underline{s}_k)$ of an N-dimensional random vector $\underline{s}_k$, encompassing L samples, has the form of

$$\sum_{i=1}^{L} \alpha_{k,i} \cdot \delta(\underline{s}_k - \underline{s}_{k,i}), \tag{17}$$

with sample positions $\underline{s}_{k,i}$ and positive scalar sample weights $\alpha_{k,i}$, for which

$$\sum_{i=1}^{L} \alpha_{k,i} = 1$$

holds [22], [36]. Therefore, the information of the true density $f_k(\underline{s}_k)$ is lossy encoded in the $L \cdot (N+1)$ Dirac mixture parameters. These parameters can be determined in a random fashion by drawing samples randomly according to the true density $f_k(\underline{s}_k)$, or in a deterministic fashion by systematically minimizing a certain distance measure between the true density $f_k(\underline{s}_k)$ and its Dirac mixture approximation (17). Moreover, a combination of both techniques is also possible.

B. Time Update

Our goal is to compute the necessary moments (5) and (6) for the Nonlinear Kalman Filter time update based on Dirac mixtures. Therefore, we have to replace the density product $f_{k-1}^{e}(\underline{x}_{k-1}) \cdot f_k^{w}(\underline{w}_k)$ with an appropriate Dirac mixture. Of course, each density could be approximated separately and the product of the resulting Dirac mixtures built afterwards. However, the result of this density product would be the Cartesian product of the employed state and noise Dirac mixtures, i.e., a Dirac mixture with $L \cdot M$ samples, where L and M denote the respective number of samples of the state and noise Dirac mixtures. This approach would not scale efficiently with an increasing number of employed samples. Fig. 4 illustrates this problem in case of scalar state x_{k-1} and system noise w_k. The state density is approximated with $L = 9$ samples whereas the Dirac mixture for the system noise employs $M = 5$ samples.

Nevertheless, we can do better by exploiting the fact that the state $\underline{x}_{k-1}$ as well as the system noise $\underline{w}_k$ are independent of each other and their respective densities, $f_{k-1}^{e}(\underline{x}_{k-1})$ and $f_k^{w}(\underline{w}_k)$, are Gaussian. That is, the product is equivalent to their, also Gaussian, joint density $f_k^{x,w}(\underline{x}_{k-1}, \underline{w}_k)$ with a zero cross-covariance matrix $\mathbf{C}_k^{x,w}$. Hence, we can avoid the Cartesian product by directly approximating the joint density

$$\begin{aligned} f_k^{x,w}(\underline{x}_{k-1}, \underline{w}_k) &= f_{k-1}^{e}(\underline{x}_{k-1}) \cdot f_k^{w}(\underline{w}_k) \\ &= \mathcal{N}\left(\begin{bmatrix} \underline{x}_{k-1} \\ \underline{w}_k \end{bmatrix}; \begin{bmatrix} \hat{\underline{x}}_{k-1}^{e} \\ \hat{\underline{w}}_k \end{bmatrix}, \begin{bmatrix} \mathbf{C}_{k-1}^{e} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_k^{w} \end{bmatrix} \right) \end{aligned}$$

using L_k^p samples according to

$$\sum_{i=1}^{L_k^p} \alpha_{k,i}^{p} \cdot \delta\left(\begin{bmatrix} \underline{x}_{k-1} \\ \underline{w}_k \end{bmatrix} - \begin{bmatrix} \underline{x}_{k-1,i} \\ \underline{w}_{k,i} \end{bmatrix} \right), \tag{18}$$

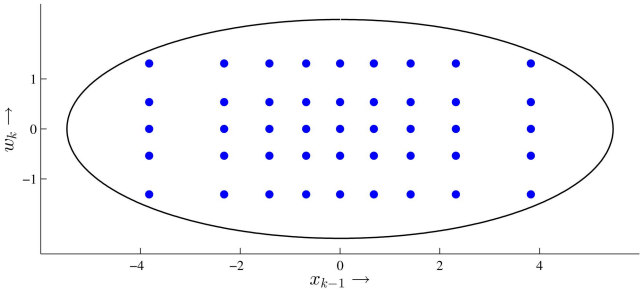


Fig. 4. Cartesian product of separate scalar state and system noise Dirac mixture approximations (blue dots). Covariance matrix (black ellipse) of the true Gaussian joint density $f_k^{x,w}(x_{k-1}, w_k)$ with confidence interval of 95%.

where $\alpha_{k,i}^{p}$ denotes the sample weights and $[\underline{x}_{k-1,i}^{\mathrm{T}}, \underline{w}_{k,i}^{\mathrm{T}}]^{\mathrm{T}}$ the sample positions in the joint space of state and system noise. Plugging this into (5) and (6), and exploiting the Dirac sifting property, we obtain the desired predicted state sample mean

$$\hat{\underline{x}}_k^{p} \approx \sum_{i=1}^{L_k^p} \alpha_{k,i}^{p} \cdot \underline{a}_k(\underline{x}_{k-1,i}, \underline{w}_{k,i}), \tag{19}$$

and predicted state sample covariance

$$\begin{aligned} \mathbf{C}_k^{p} \approx \sum_{i=1}^{L_k^p} & \alpha_{k,i}^{p} \cdot \underline{a}_k(\underline{x}_{k-1,i}, \underline{w}_{k,i}) \\ & \cdot \underline{a}_k(\underline{x}_{k-1,i}, \underline{w}_{k,i})^{\mathrm{T}} - \hat{\underline{x}}_k^{p} \cdot (\hat{\underline{x}}_k^{p})^{\mathrm{T}}, \end{aligned} \tag{20}$$

respectively.

C. Measurement Update

The LRKF measurement update can be computed in the same manner. First, we approximate the joint density

$$\begin{aligned} f_k^{x,v}(\underline{x}_k, \underline{v}_k) &= f_k^{p}(\underline{x}_k) \cdot f_k^{v}(\underline{v}_k) \\ &= \mathcal{N}\left(\begin{bmatrix} \underline{x}_k \\ \underline{v}_k \end{bmatrix}; \begin{bmatrix} \hat{\underline{x}}_k^{p} \\ \hat{\underline{v}}_k \end{bmatrix}, \begin{bmatrix} \mathbf{C}_k^{p} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_k^{v} \end{bmatrix} \right) \end{aligned}$$

of prior state and measurement noise with the Dirac mixture

$$\sum_{i=1}^{L_k^e} \alpha_{k,i}^{e} \cdot \delta\left(\begin{bmatrix} \underline{x}_k \\ \underline{v}_k \end{bmatrix} - \begin{bmatrix} \underline{x}_{k,i} \\ \underline{v}_{k,i} \end{bmatrix} \right) \tag{21}$$

encompassing L_k^e samples with weights $\alpha_{k,i}^{e}$ and positions $[\underline{x}_{k,i}^{\mathrm{T}}, \underline{v}_{k,i}^{\mathrm{T}}]^{\mathrm{T}}$. Second, plugging this into (14), (15), and (16) yields the measurement sample mean

$$\hat{\underline{y}}_k \approx \sum_{i=1}^{L_k^e} \alpha_{k,i}^{e} \cdot \underline{h}_k(\underline{x}_{k,i}, \underline{v}_{k,i}), \tag{22}$$

measurement sample covariance

$$\begin{aligned} \mathbf{C}_k^{y} \approx \sum_{i=1}^{L_k^e} & \alpha_{k,i}^{e} \cdot \underline{h}_k(\underline{x}_{k,i}, \underline{v}_{k,i}) \\ & \cdot \underline{h}_k(\underline{x}_{k,i}, \underline{v}_{k,i})^{\mathrm{T}} - \hat{\underline{y}}_k \cdot \hat{\underline{y}}_k^{\mathrm{T}}, \end{aligned} \tag{23}$$

and state measurement sample cross-covariance

$$\mathbf{C}_k^{x,y} \approx \sum_{i=1}^{L_k^e} \alpha_{k,i}^e \cdot \underline{x}_{k,i} \cdot \underline{h}_k(\underline{x}_{k,i}, \underline{v}_{k,i})^{\mathrm{T}} - \hat{\underline{x}}_k^p \cdot \hat{\underline{y}}_k^{\mathrm{T}}, \tag{24}$$

respectively. Finally, the desired posterior state mean and covariance are computed by using the Kalman Filter formulas (12) and (13).

D. The LRKF

Algorithm 1 summarizes the general procedure of a Linear Regression Kalman Filter. It is important to note that the Dirac mixture approximations (18) and (21) can be determined in completely different ways (although this is usually not the case) and do not have to utilize the same number of samples. Moreover, in case of pure additive system or measurement noise, the moment calculation can be simplified so that only the state distribution has to be sampled. This reduces the computational burden and improves the estimation quality of the LRKF.

ALGORITHM 1 *Linear Regression Kalman Filter*

Set $f_0^e(\underline{x}_0) = \mathcal{N}(\underline{x}_0, \hat{\underline{x}}_0, \mathbf{C}_0)$
for $k = 1, 2, \ldots$ **do**
  Time Update:
  Compute Dirac mixture approximation (18)
  Compute predicted state moments $\hat{\underline{x}}_k^p$ and $\mathbf{C}_k^p$ according to (19) and (20)
  Set $f_k^p(\underline{x}_k) = \mathcal{N}(\underline{x}_k; \hat{\underline{x}}_k^p, \mathbf{C}_k^p)$
  if measurement $\tilde{\underline{y}}_k$ is available **then**
    Measurement Update:
    Compute Dirac mixture approximation (21)
    Compute measurement moments $\hat{\underline{y}}_k$, $\mathbf{C}_k^y$, and $\mathbf{C}_k^{x,y}$ according to (22), (23), and (24)
    Compute posterior state moments $\hat{\underline{x}}_k^e$ and $\mathbf{C}_k^e$ according to (12) and (13)
    Set $f_k^e(\underline{x}_k) = \mathcal{N}(\underline{x}_k; \hat{\underline{x}}_k^e, \mathbf{C}_k^e)$
  else
    Set $f_k^e(\underline{x}_k) = f_k^p(\underline{x}_k)$
  end if
end for

V. THE SMART SAMPLING KALMAN FILTER

Sec. IV dealt with the general Linear Regression Kalman Filter. In order to use it, appropriate Dirac mixture approximations of the non-standard Gaussian joint densities (18) and (21) have to be determined, i.e., sets of samples with their respective positions and weights.

Our goal is to create sample sets in a deterministic manner encompassing an arbitrary number of equally weighted samples placed in the entire relevant regions of the state space, i.e., not only on the principal axes. For that reason, we turn this density approximation problem into an optimization problem by utilizing a Dirac mixture approximation procedure based on a combination of the Localized Cumulative Distribution (LCD) and a modified Cramér-von Mises distance as described in [37], [38].

Even though the LCD approach can approximate any non-standard Gaussian, it is computationally expensive due to its costly optimization procedure and, thus, is not well suited for online filter execution. But thanks to the deterministic manner of the LCD approach, we can reuse a single computed Dirac mixture approximation for every time and measurement update. By this means, we circumvent the costly online optimization of a non-standard Gaussian by computing a Dirac mixture approximation of a standard normal distribution offline and only transforming it online (during time and measurement updates) to any non-standard Gaussian using the Mahalanobis transformation [39]. In the following, we recapitulate this optimization problem and its main definitions from [38].

A. The LCD Approach

The considered problem is to determine the optimal sample positions $\underline{s}_i$ of an equally weighted Dirac mixture approximation

$$\frac{1}{L} \sum_{i=1}^{L} \delta(\underline{s} - \underline{s}_i) \tag{25}$$

of an N-dimensional standard normal distribution $\mathcal{N}(\underline{s}; \underline{0}, \mathbf{I})$. We denote the L samples, that is, the $N \cdot L$ sampling parameters, as the set

$$S := \{\underline{s}_1, \ldots, \underline{s}_L\}.$$

In order to determine S in an optimal way, we have to assess the quality of the Dirac mixture approximation (25) by defining some distance measure between both densities. Unfortunately, the classical Cumulative Distribution Function (CDF), which is often used for one-dimensional distributions, cannot be used for the multivariate case due to its non-uniqueness and asymmetry [37]. A solution is to use the Localized Cumulative Distribution, which considers the probability mass around each point of the distribution in a certain manner.

DEFINITION V.1 (Localized Cumulative Distribution) Let $f(\underline{s})$ be an N-dimensional density function. The corresponding Localized Cumulative Distribution is defined as

$$F(\underline{m}, b) = \int_{\mathbb{R}^N} f(\underline{s}) \cdot K(\underline{s} - \underline{m}, b) d\underline{s},$$

with $b \in \mathbb{R}_+$ and the symmetric and integrable kernel

$$K(\underline{s} - \underline{m}, b) = \prod_{k=1}^{N} \exp\left(-\frac{1}{2} \frac{(s^{(k)} - m^{(k)})^2}{b^2} \right).$$

Here, $\underline{m}$ characterizes the location of the kernel and b its size.

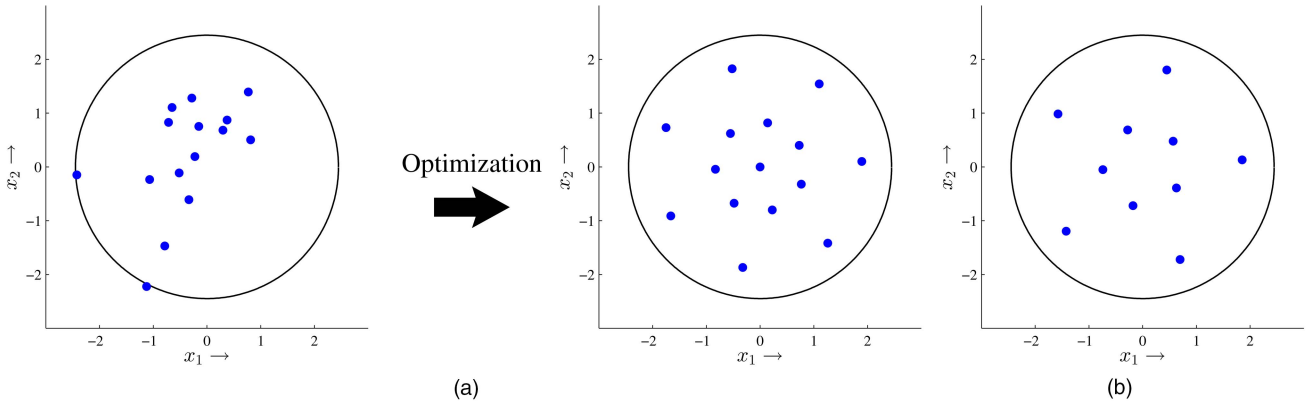


Fig. 5. LCD sampling of a two-dimensional standard normal distribution. Covariance matrices with confidence interval of 95% (black circles). The excellent state space coverage can be clearly seen. (a) LCD approach with 15 samples. Random Dirac mixture initialization on the left and optimization result (final Dirac mixture) on the right. (b) An optimization result in case of 10 samples.

Using the Dirac sifting property, the LCD of the Dirac mixture (25) can be obtained according to

$$F_{\mathrm{DM}}(S,\underline{m},b) = \frac{1}{L}\sum_{i=1}^{L}\prod_{k=1}^{N}\exp\left(-\frac{1}{2}\frac{(s_i^{(k)}-m^{(k)})^2}{b^2}\right),$$

whereas the LCD of an N-dimensional standard Gaussian is given as

$$F_{\mathcal{N}}(\underline{m},b) = \frac{b^N}{\left(\sqrt{1+b^2}\right)^N}\prod_{k=1}^{N}\exp\left(-\frac{1}{2}\frac{(m^{(k)})^2}{(1+b^2)}\right).$$

Now, we can compare both densities by comparing their respective LCDs using a modified Cramér-von Mises distance defined as follows.

DEFINITION V.2 (Modified Cramér-von Mises Distance) The modified Cramér-von Mises distance D between two LCDs $F(\underline{m},b)$ and $\tilde{F}(\underline{m},b)$ is given by

$$D = \int_{\mathbb{R}_+} w(b)\int_{\mathbb{R}^N}(F(\underline{m},b)-\tilde{F}(\underline{m},b))^2 d\underline{m}db$$

with weighting function

$$w(b) = \begin{cases}\dfrac{1}{b^{N-1}}, & b\in(0,b_{\max}]\\ 0, & \text{elsewhere.}\end{cases}$$

The modified Cramér-von Mises distance between the LCDs $F_{\mathrm{DM}}(\cdot,\cdot,\cdot)$ and $F_{\mathcal{N}}(\cdot,\cdot)$ is given by

$$D(S) = D_1 - 2D_2(S) + D_3(S) \quad \text{with } D_i = \int_{\mathbb{R}_+} P_i db, \tag{26}$$

and the sample-independent part

$$P_1 = \frac{\pi^{N/2}b^{N+1}}{\left(\sqrt{1+b^2}\right)^N},$$

as well as the sample-dependent parts

$$P_2(S) = \frac{(2\pi)^{N/2}b^{N+1}}{L\left(\sqrt{1+2b^2}\right)^N}\cdot\sum_{i=1}^{L}\exp\left(-\frac{1}{2}\sum_{k=1}^{N}\frac{(s_i^{(k)})^2}{1+2b^2}\right)$$

and

$$P_3(S) = \frac{\pi^{N/2}b}{L^2}\cdot\sum_{i=1}^{L}\sum_{j=1}^{L}\exp\left(-\frac{1}{2}\sum_{k=1}^{N}\frac{(s_i^{(k)}-s_j^{(k)})^2}{2b^2}\right).$$

Given this distance measure, the optimal sample positions $\underline{s}_i$ are computed as follows. One starts by randomly choosing initial sampling parameters S, i.e., placing L N-dimensional samples randomly in state space, where L is the cardinality of the desired optimal Dirac mixture approximation and N the dimension of the considered standard normal distribution. Then, an optimization procedure, e.g., a quasi-Newton method (L-BFGS) [40], [41], changes these initial sampling parameters S, i.e., moves the samples in state space, such that the distance measure (26) between the standard normal and its Dirac mixture approximation is minimized. Thus, we perform a *global optimization* as all sample positions $\underline{s}_i$ are optimized at once. Another solution would be to use greedy optimizations such as [42], where an existent Dirac mixture is extended by simply adding additional samples and leaving the existing samples unchanged. Unfortunately, this leads to suboptimal approximation results and, hence, is not considered here.

Fig. 5(a) illustrates the proposed LCD approach for the case of a two-dimensional standard normal distribution and $L = 15$ samples. The random initialization is shown on the left, whereas the final optimal approximation is shown on the right. Fig. 5(b) depicts another optimization result with $L = 10$ samples.

Now, given a non-standard Gaussian distribution

$$\mathcal{N}(\underline{z};\hat{\underline{z}},\mathbf{C}^z) \tag{27}$$

during filter execution, i.e., the joint densities $f_k^{x,w}(\underline{x}_{k-1},\underline{w}_k)$ and $f_k^{x,v}(\underline{x}_k,\underline{v}_k)$, we compute the matrix square root $\sqrt{\mathbf{C}^z}$ of $\mathbf{C}^z$ using the Cholesky decomposition,[5] and individually translate, rotate, and scale each

[5] Other matrix square root operations, such as the eigendecomposition, are also possible.

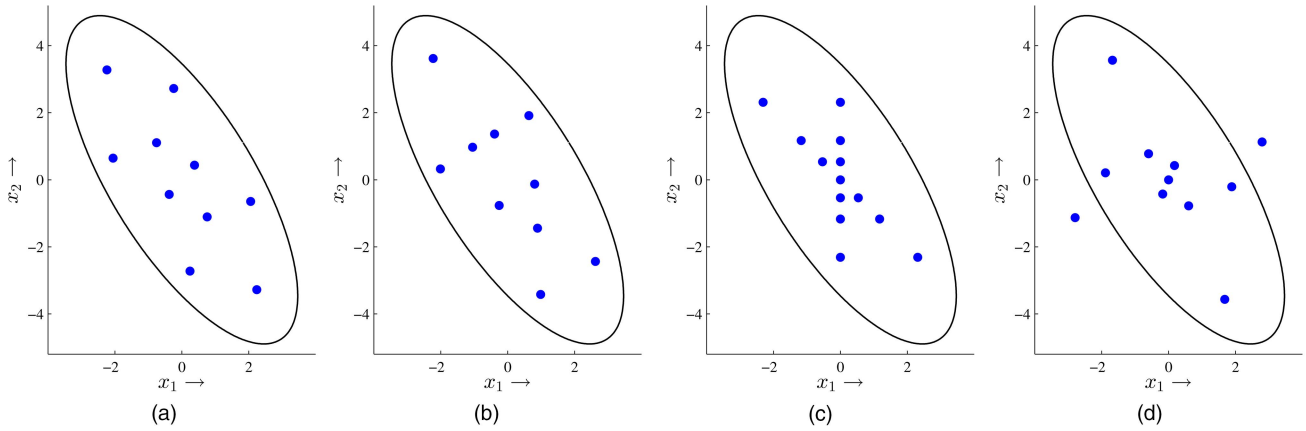


Fig. 6. Difference between the direct LCD approximation (a) and the standard normal LCD approximation (same samples as depicted in Fig. 5(b)) with subsequent transformation using a Cholesky decomposition (b). For comparison, we transformed sample sets from the GF and the RUKF (same samples as depicted in Fig. 1) as well. (a) Direct approximation. (b) Suboptimal approximation. (c) GF approximation. (d) RUKF approximation.

sample of (25) according to

$$\underline{z}_i = \sqrt{\mathbf{C}^z} \cdot \underline{s}_i + \hat{\underline{z}} \quad \forall i \in \{1,\ldots,L\}, \tag{28}$$

so that the new sample set

$$Z := \{\underline{z}_1,\ldots,\underline{z}_L\}$$

forms the Dirac mixture approximation

$$\frac{1}{L}\sum_{i=1}^{L}\delta(\underline{z}-\underline{z}_i)$$

of the non-standard Gaussian (27). Note that these non-standard Gaussian samples are still equally weighted.

It is important to note that this combination of standard normal approximation with subsequent transformation delivers only suboptimal results compared to a direct LCD-based approximation of the non-standard Gaussian. Fig. 6 exemplifies this problem for a two-dimensional non-standard, i.e., rotated and scaled, Gaussian distribution and Dirac mixtures with 10 samples. The direct LCD approximation of the considered Gaussian is depicted in Fig. 6(a). One can see an optimal, regular placement of the samples, covering the relevant state space regions homogeneously. In contrast to this, a suboptimal solution is shown in Fig. 6(b). Here, larger regions of the relevant state space are uncovered, e.g., the top-left region of the Gaussian or its middle part. Nevertheless, all state-of-the-art LRKFs suffer from the problem of a suboptimal non-standard Gaussian approximation, as they rely on the online Mahalanobis transformation, too. To demonstrate this, we transformed sample sets from the GF and the RUKF as well (see Fig. 6(c) and Fig. 6(d)).

B. The New LRKF

By using offline computed LCD-based Dirac mixture approximations of standard normal distributions (25) in combination with online transformations (28) during the LRKF time and measurement updates, we introduce the new Smart Sampling Kalman Filter (S^{2}KF), with its powerful feature of using an arbitrary number of optimally placed samples in the entire state space. There exist no sampling constraints such as axis-aligned samples or that the number of samples must be a multiple of the state dimension as with the UKF, GF, or RUKF.

As will be shown in the evaluation, with an increasing number of used samples in (25) the S^{2}KF converges to the analytic statistical linearization as the resulting Dirac mixture approximation of the standard normal distribution becomes more accurate. In contrast to the UKF with its fixed-size sample set, this allows an extensive evaluation of the given system and measurement models, as more and more samples become available in the relevant regions of the state space. Moreover, this makes a non-positive definite state covariance matrix very unlikely and the filter more reliable. As a consequence, the estimation quality can be easily improved by simply increasing the number of employed samples, which offers an intuitive optimization parameter. Of course, this effect is also true for filters relying on random sampling. But, due to the optimal sample placement, the S^{2}KF converges much faster, so that already a small number of samples provides an excellent estimation quality. Regarding the filter complexity, assuming that the Dirac mixture approximations of the required standard normal distributions are already computed, the computational complexity of the S^{2}KF grows only linearly with the number of used samples L for a fixed dimension N.

One should keep in mind that the LCD approach cannot create unique Dirac mixture approximations for a given dimension N and number of samples L as the standard normal distribution is rotation-invariant and the optimization procedure is initialized with random samples. However, due to the *reuse* of offline computed Dirac mixture approximations, the S^{2}KF results still become reproducible. In other words, executing the S^{2}KF

with the same inputs multiple times, i.e., model parameters, initial state, and measurements, will always produce the same results as the same sample approximations are used for each execution.

C. The Sample Cache

The proposed S^{2}KF needs several LCD-based Dirac mixture approximations of standard normal distributions, one for each required combination of dimension N and number of employed samples L, depending on the concrete filtering problem (state dimension, noise dimension, and selected filter accuracy). An option to obtain these approximations would be to recompute all required approximations before each program start. However, for large dimensions and/or number of samples, this can be very time-consuming. Moreover, the estimation results from different program executions would be different as always new sample sets would be used.

For that reason, we introduce a sample storage called *Sample Cache*. Whenever a requested sample set for a given combination of dimension N and number of samples L is not available during S^{2}KF execution, it is computed on demand,[6] that is, *transparent for the user*, and subsequently stored persistently in the file system for later reuse. Over time, the Sample Cache grows and the necessity for time-consuming sample generation becomes more unlikely. Of course, if the user knows all the needed approximations before filter execution, all of them can be computed and stored in Sample Cache in advance so that no sample computation is required at all during filter execution.

VI. EVALUATION

In this section, we compare the new S^{2}KF with state-of-the-art LRKFs by performing recursive state estimation using various nonlinear system and measurement equations. In the first evaluation, the focus lies on nonlinear prediction, whereas in the second evaluation the filters have to cope with nonlinear measurement updates.

As every LRKF is an approximation of the Kalman Filter based on analytic statistical linearization, an LRKF estimate should be as close as possible to the this estimate. A considerable difference between both estimates can only result from inaccurate moment calculations by the LRKF and, hence, its utilized Gaussian sampling technique. Consequently, in order to assess the investigated LRKFs and their used sampling techniques properly, the state estimates obtained by analytic statistical linearization will serve in both evaluations as reference (ground truth) here. Using another ground truth, for example the true system state, would be unfavorable to detect such inaccurate moment calculations. The reason is that an LRKF estimate that is close to the true system state does not indicate whether the moments were calculated correctly, as the estimate obtained by analytic statistical linearization might be quite different from the true system state.

A. Batch Reactor

We consider the gas-phase reaction proposed in [43], resulting in the estimation problem consisting of a two-dimensional state $\underline{x}_k = [x_{a,k}, x_{b,k}]^{\mathrm{T}}$, which obeys the time-invariant nonlinear system model

$$\begin{aligned}\underline{x}_k &= \underline{a}(\underline{x}_{k-1}, \Delta t, \underline{w}) \\ &= \underline{x}_{k-1} + \Delta t \cdot \begin{bmatrix} -0.32 \cdot x_{a,k-1}^2 \\ 0.16 \cdot x_{a,k-1}^2 \end{bmatrix} + \underline{w}, \end{aligned} \tag{29}$$

with input $\Delta t = 0.1$ and time-invariant, additive, and zero-mean Gaussian white noise $\underline{w}$ with covariance

$$\mathbf{C}^w = \mathrm{diag}(10^{-5}, 10^{-5}).$$

Over time, we receive scalar measurements $\tilde{y}_k$ according to the time-invariant linear mapping

$$y_k = h(\underline{x}_k, v) = [1 \;\; 1] \cdot \underline{x}_k + v, \tag{30}$$

where v denotes time-invariant, additive, and zero-mean Gaussian white noise with variance $\mathbf{C}^v = 0.1$.

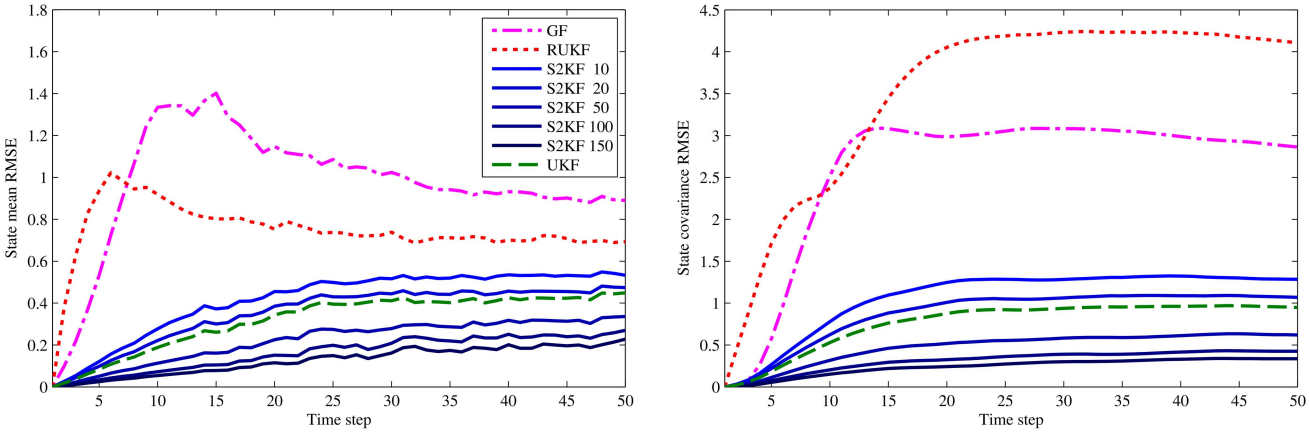
We compare the following estimators:

- exact, analytic statistical linearization using [44], for which closed-form expressions are given in the Appendix,
- the UKF with equally weighted samples,
- the GF with 25 samples on each principal axis,
- the RUKF with 12 iterations, and finally
- the new S^{2}KF with 10, 20, 50, 100, and 150 samples, respectively, in order to demonstrate the convergence of the S^{2}KF towards the analytic statistical linearization.

As the measurement equation is linear in this simulation, we can accurately evaluate the nonlinear prediction performance of the investigated filters. More precisely, the measurement update is calculated in closed-form by all filters, i.e., the optimal closed-form Kalman Filter update is used. Additionally, due to the fact that the system model (29) is corrupted by pure additive noise, sampling is reduced to the two-dimensional state space (see Sec. IV-D). Table II summarizes the resulting numbers of samples used by each LRKF for the prediction step.

The simulation consists of $R = 1000$ Monte Carlo runs. For each Monte Carlo run, the true system state is obtained by initializing it with $\underline{x}_0 = [3, 1]^{\mathrm{T}}$ and recursively propagating it 50 times, together with noise realizations of $\underline{w}$, through the system model (29), resulting in a simulation with 50 time steps. Additionally, we simulate one noisy measurement each time step by using (30) together with a noise realization of v. All filters are initialized with mean $\hat{\underline{x}}_0^e = [0.5, 3.5]^{\mathrm{T}}$ and covariance $\mathbf{C}_0^e = \mathrm{diag}(10, 10)$.

[6] As a consequence, in such a case the filter execution stalls until the sample set is computed.



(a) (b)

Fig. 7. Batch reactor simulation over 50 time steps. (a) State mean RMSE $\bar{x}_k$. (b) State covariance RMSE $\bar{C}_k$.

TABLE II
Employed LRKFs and their respective sampling settings for the batch reactor simulation.

LRKF	Number of samples	Sample placement
UKF	$2 \cdot 2 + 1 = 5$	Axes only
GF	$24 \cdot 2 + 1 = 49$	Axes only
RUKF	$12 \cdot (2 \cdot 2) + 1 = 49$	Entire state space
S^2KF	10	Entire state space
S^2KF	20	Entire state space
S^2KF	50	Entire state space
S^2KF	100	Entire state space
S^2KF	150	Entire state space

In order to assess the estimation quality of each LRKF, we compute the Root Mean Square Error (RMSE) of their posterior state mean over all simulation runs with respect to the analytic statistical linearization posterior state mean, that is,

$$\bar{x}_k = \sqrt{\frac{1}{R}\sum_{r=1}^{R} \|\hat{\underline{x}}_k^{(r)} - \hat{\underline{x}}_{a,k}^{(r)}\|_2^2},$$

where $\hat{\underline{x}}_k^{(r)}$ denotes the respective LRKF state mean and $\hat{\underline{x}}_{a,k}^{(r)}$ the state mean of the analytic statistical linearization. The results are depicted in Fig. 7(a). Here, the GF shows a very high RMSE at the beginning. Over time, the RMSE decreases but remains at a relatively high level. The RUKF does not possess such extreme RMSE but it is constantly at a higher level compared to the UKF and S^2KF estimates. As opposed to this, the UKF delivers quite good results although the GF uses much more samples per axis. Moreover, all S^2KF instances are also much better than the GF and RUKF, and the S^2KF instances using 50 or more samples deliver the best posterior state means of all investigated LRKFs. The expected convergence of the S^2KF with an increasing number of samples towards the analytic statistical linearization can be clearly seen.

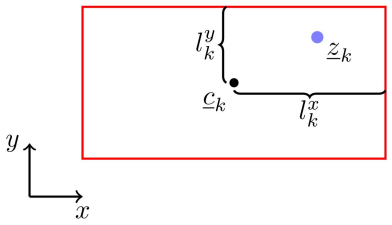


Fig. 8. Axis-aligned extended rectangular target with position $\underline{c}_k$, extent $\underline{l}_k$, and a target surface point $\underline{z}_k$.

Moreover, we compute the RMSE for their posterior state covariance in a similar manner according to

$$\bar{C}_k = \sqrt{\frac{1}{R}\sum_{r=1}^{R} \|\mathbf{C}_k^{(r)} - \mathbf{C}_{a,k}^{(r)}\|^2},$$

where $\|\cdot\|$ denotes the Frobenius norm, $\mathbf{C}_k^{(r)}$ the respective LRKF state covariance, and $\mathbf{C}_{a,k}^{(r)}$ the state covariance of the analytic statistical linearization. When looking at the results shown in Fig. 7(b), one should notice that the GF as well as the RUKF estimate themselves much too uncertain compared to the analytic moment calculation. Both errors increase quickly and decrease only at a slow pace over time. In contrast, the covariance of the UKF is much closer to the analytic statistical linearization than these filters. However, as with the state mean, the S^2KF instances can outperform the UKF and its convergence towards analytic statistical linearization is as expected.

B. Extended Target Tracking

In this section, we evaluate the S^2KF by means of tracking an extended target modeled as multiplicative noise. Our goal is to estimate the position $\underline{c}_k = [c_k^x, c_k^y]^\mathrm{T}$ and extent $\underline{l}_k = [l_k^x, l_k^y]^\mathrm{T}$ of a two-dimensional axis-aligned rectangular target (see Fig. 8). The hidden system state is given by $\underline{x}_k = [\underline{l}_k^\mathrm{T}, \underline{c}_k^\mathrm{T}]^\mathrm{T}$.

In order to incorporate target information into our state estimate, we assume uniformly distributed, noisy measurements stemming from the surface of the target.

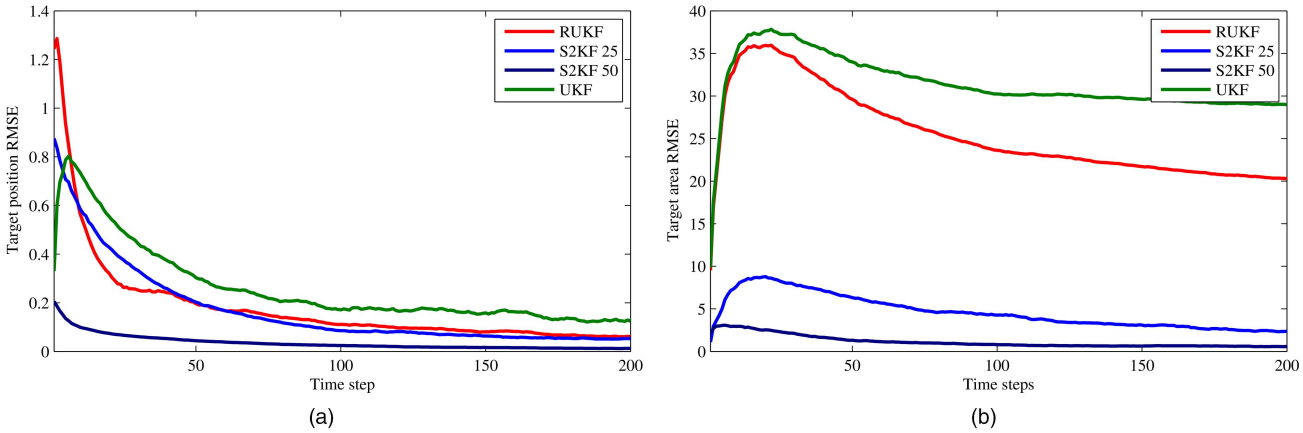


Fig. 9. Extended target tracking evaluation results. (a) RMSE for the target position. (b) RMSE for the target area.

For this purpose, we extend the approach proposed in [6]. The basic idea is that each point of the target surface can be reached by scaling the axis lengths l_k^x and l_k^y individually and adding the center $\underline{c}_k$, i.e.,

$$\underline{z}_k = \mathbf{H} \cdot \underline{l}_k + \underline{c}_k,$$

with uncorrelated state independent multiplicative noise

$$\mathbf{H} = \mathrm{diag}(h^x, h^y).$$

As the measurements are uniformly distributed, h^x and h^y also have to be uniformly distributed in the interval $[-1,1]$ (see Fig. 8). Taking the measurement noise into account yields the preliminary nonlinear measurement equation

$$\underline{m}_k = \underline{z}_k + \underline{w} = \mathbf{H} \cdot \underline{l}_k + \underline{c}_k + \underline{w}, \tag{31}$$

where $\underline{w}$ denotes additive, zero-mean Gaussian white noise with unit covariance.

Unfortunately, as mentioned in [6], linear estimators, including the S^{2}KF as well, are unsuitable for tracking extended targets modeled this way. To overcome this issue, we pick up on the author's suggestion and extend the measurement equation (31) to match the best quadratic estimator according to

$$\underline{y}_k = \begin{bmatrix} \underline{m}_k \\ \underline{m}_k^2 \end{bmatrix} = \begin{bmatrix} \mathbf{H} \cdot \underline{l}_k + \underline{c}_k + \underline{w} \\ (\mathbf{H} \cdot \underline{l}_k + \underline{c}_k + \underline{w})^2 \end{bmatrix}.$$

To keep things simple, this evaluation uses a static target with extent $\underline{l} = [4,2]^\mathrm{T}$ located at $\underline{c} = [3,5]^\mathrm{T}$. Thus, the temporal evolution of $\underline{x}_k$ is modeled as random walk, i.e., employing the linear system equation

$$\underline{x}_k^p = \underline{x}_{k-1}^e + \underline{v},$$

where $\underline{v}$ is an additive, zero-mean Gaussian white noise with covariance

$$\mathbf{C}^v = \mathrm{diag}(10^{-4}, 10^{-4}, 10^{-3}, 10^{-3}).$$

We compare the following estimators:

- exact, analytic statistical linearization using [44],
- the UKF with $2 \cdot 8 + 1 = 17$ equally weighted samples,
- the RUKF with 10 iterations (resulting in $10 \cdot (2 \cdot 8) + 1 = 161$ samples), and
- two S^{2}KF instances with 25 and 50 samples, respectively, in order to demonstrate the convergence of the S^{2}KF towards the analytic statistical linearization.

Due to the fact that the S^{2}KF and the RUKF require a measurement noise described in terms of a Gaussian distribution,[7] and while the UKF only considers the first two moments of the measurement noise, we approximate the uniformly distributed multiplicative noise $\mathbf{H}$ as Gaussian distribution by means of moment matching. The simulation consists of $R = 100$ Monte Carlo runs. For each Monte Carlo run, the initial state estimate is set to $\hat{\underline{x}}_0^e = [1,1,0,0]^\mathrm{T}$ and $\mathbf{C}_0^e = \mathbf{I}_4$, and at each time step we receive a single noisy measurement from the target surface.

Similar to the batch reactor simulation, we assess the estimation quality of each LRKF by computing the RMSE of their target position estimate over all simulation runs with respect to the analytic statistical linearization target position estimate, that is,

$$\bar{c}_k = \sqrt{\frac{1}{R} \sum_{r=1}^{R} \| \hat{\underline{c}}_k^{(r)} - \hat{\underline{c}}_{a,k}^{(r)} \|_2^2},$$

where $\hat{\underline{c}}_k^{(r)}$ denotes the respective LRKF position estimate and $\hat{\underline{c}}_{a,k}^{(r)}$ the analytic statistical linearization position estimate. The results of the target position RMSE are depicted in Fig. 9(a). Here, all LRKFs quickly decrease their RMSE over time. However, the UKF converges to a little higher RMSE than the other LRKFs. The S^{2}KF using 50 samples converges quickly to an error nearly zero and yields the best estimation result of all LRKFs.

Additionally, we compare their target extent estimate by computing the RMSE of the estimated target area in

[7] This is a consequence of the fact that these filters rely on explicit sampling a Gaussian distribution.

a similar manner according to

$$\bar{A}_k = \sqrt{\frac{1}{R}\sum_{r=1}^{R}(4\cdot(\hat{l}_k^x)^{(r)}\cdot(\hat{l}_k^y)^{(r)} - 4\cdot(\hat{l}_{a,k}^x)^{(r)}\cdot(\hat{l}_{a,k}^y)^{(r)})^2},$$

where $[\hat{l}_k^x, \hat{l}_k^y]^{(r)}$ denotes the respective LRKF target extent estimate and $[\hat{l}_{a,k}^x, \hat{l}_{a,k}^y]^{(r)}$ the analytic statistical linearization target extent estimate. Fig. 9(b) shows the RMSE for the estimated target area. The 50 sample S^{2}KF is the only filter that directly converges to a small error, whereas the UKF and the RUKF quickly increase to a high RMSE and decrease only at a very low rate. The UKF area estimate is clearly incorrect, as its area error converges to approximately 28 m^2. As opposed to this, the S^{2}KF using 25 samples corrects its estimate relatively fast.

Fig. 10 shows a representative simulation run after incorporating 75 measurements. One can see that the UKF leaves its initial state estimate of a target of 4 m^2 completely unchanged which coincides with its area RMSE of approximately 28 m^2, and that the RUKF estimates the target much too small. In contrast to this, the S^{2}KF using 50 samples is almost identical to the analytic statistical linearization estimate. The general S^{2}KF convergence towards the analytic statistical linearization concerning both target position and extent is evident as the 50 sample instance yields the much better tracking results than the S^{2}KF using 25 samples.

C. Summary

The performed evaluations showed a general problem of sample-based filtering: not only the amount of samples and their placement are important for the estimation results, but also their interaction with the underlying system and measurement models. This was seen in the two following cases. On the one hand, the axis-aligned placement of the UKF samples deliver quite good results with the batch reactor model, but completely failed during the extended object tracking. On the other hand, the RUKF had problems in both evaluations although it places its samples not only the axes. In contrast, with the ability to use an arbitrary amount of samples with optimal placement in the relevant regions of the state space, the new S^{2}KF can easily be tuned to perform well in both filtering problems.

VII. CONCLUSIONS

In this paper, we introduced a new accurate LRKF called Smart Sampling Kalman Filter (S^{2}KF). It is based on LCD-based Dirac mixture approximations of standard normal distributions comprising an arbitrary number of samples, which are placed optimally in the relevant regions of the state space, that is, not only on the principal axes. Hence, the S^{2}KF can be seen as the ultimate generalization of all sample-based Nonlinear Kalman Filters.

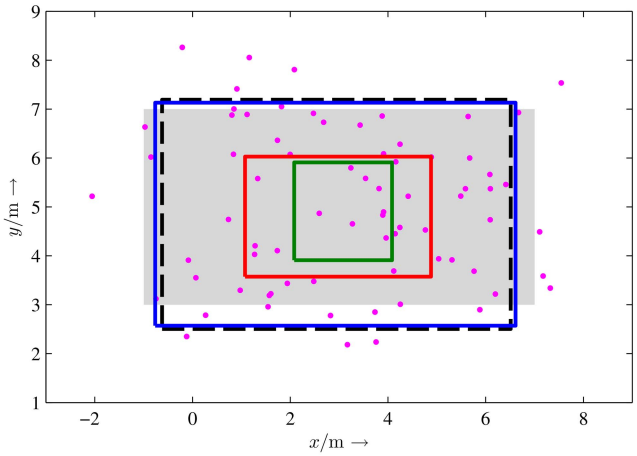


Fig. 10. Representative simulation run with extended target (gray rectangle), randomly generated noisy measurements (magenta dots), analytic statistical linearization estimate (black dashed line), 50 sample S^{2}KF estimate (blue line), UKF estimate (green line), and RUKF estimate (red line).

First, we gave a general introduction to Gaussian estimators and Nonlinear Kalman Filters. We explained the idea of analytic statistical linearization and its approximation in form of the Linear Regression Kalman Filter. After that, we described the optimal Gaussian sampling using the LCD approach and based on this introduced the S^{2}KF. Moreover, we proposed the idea of a Sample Cache that stores computed Dirac mixture approximations of standard normal distributions persistent in the file system for later reuse. Finally, we evaluated the S^{2}KF by means a nonlinear prediction scenario and extended target tracking against state-of-the-art LRKFs. The new filter showed the expected convergence towards the analytic statistical linearization and outperformed state-of-the-art LRKFs including the UKF and RUKF.

As the S^{2}KF encompasses the same structure as the UKF, the S^{2}KF can easily replace it in order to enhance existing and future filtering applications. Morever, the S^{2}KF can be directly used in Gaussian mixture LRKFs [25], [45] and LRKF square root implementations such as described in [46].

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their positive feedback and helpful suggestions.

APPENDIX

Our goal is to calculate the predicted state mean and covariance of the nonlinear dynamic system considered in Sec. VI-A in closed-form. Given the state mean

$$\underline{\hat{x}}_{k-1}^e = [\hat{x}_{a,k-1}, \hat{x}_{b,k-1}]^{\mathrm{T}}$$

and state covariance

$$\mathbf{C}_{k-1}^e = \begin{bmatrix} c_{aa,k-1}^x & c_{ab,k-1}^x \\ c_{ab,k-1}^x & c_{bb,k-1}^x \end{bmatrix}$$

from the last time step $k-1$, and the system noise statistics

$$\underline{\hat{w}}_k = [\hat{x}_{a,k}, \hat{x}_{b,k}]^{\mathrm{T}}$$

and

$$\mathbf{C}_k^w = \begin{bmatrix} c_{aa,k}^w & 0 \\ 0 & c_{bb,k}^w \end{bmatrix}$$

from the current time step k, we obtain the moments

$$\begin{aligned}
\mathrm{E}[x_{a,k-1}^2] &= \hat{x}_{a,k-1}^2 + c_{aa,k-1}^x, \\
\mathrm{E}[x_{a,k-1}^3] &= \hat{x}_{a,k-1}^3 + 3 \cdot \hat{x}_{a,k-1} \cdot c_{aa,k-1}^x, \\
\mathrm{E}[x_{a,k-1}^4] &= \hat{x}_{a,k-1}^4 + 6 \cdot \hat{x}_{a,k-1}^2 \cdot c_{aa,k-1}^x \\
&\quad + 3 \cdot (c_{aa,k-1}^x)^2,
\end{aligned}$$

$$\begin{aligned}
\mathrm{E}[x_{b,k-1}^2] &= \hat{x}_{b,k-1}^2 + c_{bb,k-1}^x, \\
\mathrm{E}[x_{a,k-1} x_{b,k-1}] &= \hat{x}_{a,k-1} \hat{x}_{b,k-1} + c_{ab,k-1}^x, \\
\mathrm{E}[x_{a,k-1}^2 x_{b,k-1}] &= \hat{x}_{a,k-1}^2 \hat{x}_{b,k-1} + c_{aa,k-1}^x \cdot \hat{x}_{b,k-1} \\
&\quad + 2 \cdot c_{ab,k-1}^x \cdot \hat{x}_{a,k-1}, \\
\mathrm{E}[w_{a,k}^2] &= \hat{x}_{a,k}^2 + c_{aa,k}^w, \\
\mathrm{E}[w_{b,k}^2] &= \hat{x}_{b,k}^2 + c_{bb,k}^w,
\end{aligned}$$

respectively. Using these moments together with $p = -0.32$ and $q = 0.16$, we obtain the predicted state mean according to

$$\underline{\hat{x}}_k^p = \begin{bmatrix} \hat{x}_{a,k-1} + p \cdot \Delta t \cdot \mathrm{E}[x_{a,k-1}^2] \\ \hat{x}_{b,k-1} + q \cdot \Delta t \cdot \mathrm{E}[x_{a,k-1}^2] \end{bmatrix} + \underline{\hat{w}}_k,$$

and the predicted state covariance matrix according to

$$\mathbf{C}_k^p = \begin{bmatrix} m_{aa,k} & m_{ab,k} \\ m_{ab,k} & m_{bb,k} \end{bmatrix} - \underline{\hat{x}}_k^p \cdot (\underline{\hat{x}}_k^p)^{\mathrm{T}},$$

with

$$\begin{aligned}
m_{aa,k} &= \mathrm{E}[x_{a,k-1}^2] + (p \cdot \Delta t)^2 \cdot \mathrm{E}[x_{a,k-1}^4] + \mathrm{E}[w_{a,k}^2] \\
&\quad + 2 \cdot (p \cdot \Delta t \cdot \mathrm{E}[x_{a,k-1}^3] + \hat{x}_{a,k-1} \cdot \hat{x}_{a,k} \\
&\quad + p \cdot \Delta t \cdot \mathrm{E}[x_{a,k-1}^2] \cdot \hat{x}_{a,k}), \\
m_{bb,k} &= \mathrm{E}[x_{b,k-1}^2] + (q \cdot \Delta t)^2 \cdot \mathrm{E}[x_{a,k-1}^4] + \mathrm{E}[w_{b,k}^2] \\
&\quad + 2 \cdot (q \cdot \Delta t \cdot \mathrm{E}[x_{a,k-1}^2 x_{b,k-1}] + \hat{x}_{b,k-1} \cdot \hat{x}_{b,k} \\
&\quad + q \cdot \Delta t \cdot \mathrm{E}[x_{a,k-1}^2] \cdot \hat{x}_{b,k}), \\
m_{ab,k} &= \mathrm{E}[x_{a,k-1} x_{b,k-1}] + q \cdot \Delta t \cdot \mathrm{E}[x_{a,k-1}^3] \\
&\quad + \hat{x}_{a,k-1} \cdot \hat{x}_{b,k} + p \cdot \Delta t \cdot \mathrm{E}[x_{a,k-1}^2 x_{b,k-1}] \\
&\quad + p \cdot q \cdot \Delta t^2 \cdot \mathrm{E}[x_{a,k-1}^4] \\
&\quad + p \cdot \Delta t \cdot \mathrm{E}[x_{a,k-1}^2] \cdot \hat{x}_{b,k} \\
&\quad + \hat{x}_{a,k} \cdot (\hat{x}_{b,k-1} + q \cdot \Delta t \cdot \mathrm{E}[x_{a,k-1}^2] + \hat{x}_{b,k}).
\end{aligned}$$

REFERENCES

[1] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon
Beyond the Kalman Filter: Particle Filters for Tracking Applications.
Artech House Publishers, 2004.

[2] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan
Estimation with Applications to Tracking and Navigation.
New York Chichester Weinheim Brisbane Singapore Toronto: Wiley-Interscience, 2001.

[3] Johann Wolfgang Koch
"Bayesian Approach to Extended Object and Cluster Tracking Using Random Matrices,"
IEEE Transactions on Aerospace and Electronic Systems, vol. 44, no. 3, pp. 1042–1059, Jul. 2008.

[4] Frederik Beutler, Marco F. Huber, and Uwe D. Hanebeck
"Semi-Analytic Stochastic Linearization for Range-Based Pose Tracking,"
in *Proceedings of the 2010 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2010)*, Salt Lake City, USA, Sep. 2010, pp. 44–49.

[5] Marcus Baum and Uwe D. Hanebeck
"Shape Tracking of Extended Objects and Group Targets with Star-Convex RHMs,"
in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, USA, Jul. 2011, pp. 1–8.

[6] Marcus Baum, Florian Faion, and Uwe D. Hanebeck
"Modeling the Target Extent with Multiplicative Noise,"
in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012, pp. 2406–2412.

[7] Licong Zhang, Jürgen Sturm, Daniel Cremers, and Dongheui Lee
"Real-Time Human Motion Tracking Using Multiple Depth Cameras,"
in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012)*, Vilamoura, Portugal, Oct. 2012, pp. 2389–2395.

[8] Marcus Baum and Uwe D. Hanebeck
"Fitting Conics to Noisy Data Using Stochastic Linearization,"
in *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)*, San Francisco, USA, Sep. 2011, pp. 2050–2055.

[9] Antonio Zea, Florian Faion, Marcus Baum, and Uwe D. Hanebeck
"Level-Set Random Hypersurface Models for Tracking Non-Convex Extended Objects,"
in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.

[10] Sebastian Thrun, Wolfram Burgard, and Dieter Fox
Probabilistic Robotics.
Cambridge, London: MIT Press, 2005.

[11] Florian Faion, Patrick Ruoff, Antonio Zea, and Uwe D. Hanebeck
"Recursive Bayesian Calibration of Depth Sensors with Non-Overlapping Views,"
in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012, pp. 757–762.

[12] Arnaud Doucet and Adam M. Johansen
"A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later,"
in *Oxford Handbook of Nonlinear Filtering*, 2011, pp. 656–704.

[13] Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp
"A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking,"
IEEE Transactions on Signal Processing, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[14] Jayesh H. Kotecha and Petar M. Djuric
"Gaussian Particle Filtering,"
IEEE Transactions on Signal Processing, vol. 51, no. 10, pp. 2592–2601, Oct. 2003.

[15] Jannik Steinbring and Uwe D. Hanebeck
"Progressive Gaussian Filtering Using Explicit Likelihoods,"
in *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, Jul. 2014.

[16] Uwe D. Hanebeck
"PGF 42: Progressive Gaussian Filtering with a Twist,"
in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.

[17] Rudolf E. Kalman
"A New Approach to Linear Filtering and Prediction Problems,"
in *Transaction of the ASME—Journal of Basic Engineering*, Mar. 1960, pp. 35–45.

[18] Dan Simon
Optimal State Estimation,
1st ed. Wiley & Sons, 2006.

[19] X. Rong Li and Yu Liu
"Generalized Linear Minimum Mean-Square Error Estimation,"
in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013, pp. 1819–1826.

[20] Tor Steinar Schei
"A Finite-Difference Method for Linearization in Nonlinear Estimation Algorithms,"
Automatica, vol. 33, no. 11, pp. 2053–2058, 1997.

[21] Rudolph van der Merwe
"Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models,"
Ph.D. dissertation, OGI School of Science & Engineering, Oregon Health & Science University, Portland, Apr. 2004.

[22] Marco F. Huber, Frederik Beutler, and Uwe D. Hanebeck
"Semi-Analytic Gaussian Assumed Density Filter,"
in *Proceedings of the 2011 American Control Conference (ACC 2011)*, San Francisco, USA, Jun. 2011.

[23] Tine Lefebvre, Herman Bruyninckx, and Joris De Schutter
"Kalman Filters for Non-Linear Systems: A Comparison of Performance,"
International Journal of Control, vol. 77, no. 7, pp. 639–653, May 2004.

[24] ———
"Appendix A: The Linear Regression Kalman Filter,"
in *Nonlinear Kalman Filtering for Force-Controlled Robot Tasks*, ser. Springer Tracts in Advanced Robotics. Berlin Heidelberg: Springer, 2005, vol. 19.

[25] Marco F. Huber, Frederik Beutler, and Uwe D. Hanebeck
"(Semi-)Analytic Gaussian Mixture Filter,"
in *Proceedings of the 18th IFAC World Congress (IFAC 2011)*, Milano, Italy, Aug. 2011.

[26] Simon J. Julier and Jeffrey K. Uhlmann
"A New Extension of the Kalman Filter to Nonlinear Systems,"
in *11th Int. Symp. Aerospace/Defense Sensing, Simulation and Controls*, 1997, pp. 182–193.

[27] ———
"Unscented Filtering and Nonlinear Estimation,"
in *Proceedings of the IEEE*, vol. 92, Mar. 2004, pp. 401–422.

[28] Ryan Turner and Carl Edward Rasmussen
"Model Based Learning of Sigma Points in Unscented Kalman Filtering,"
Neurocomputing, vol. 80, pp. 47–53, 2012.

[29] Jindrich Dunik, Miroslav Simandl, and Ondrej Straka
"Unscented Kalman Filter: Aspects and Adaptive Setting of Scaling Parameter,"
IEEE Transactions on Automatic Control, vol. 57, no. 9, pp. 2411–2416, Sep. 2012.

[30] Marco F. Huber and Uwe D. Hanebeck
"Gaussian Filter Based on Deterministic Sampling for High Quality Nonlinear Estimation,"
in *Proceedings of the 17th IFAC World Congress (IFAC 2008)*, vol. 17, Seoul, Republic of Korea, Jul. 2008.

[31] Jindrich Dunik, Ondrej Straka, and Miroslav Simandl
"The Development of a Randomised Unscented Kalman Filter,"
in *Proceedings of the 18th IFAC World Congress*, Milano, Italy, Aug. 2011, pp. 8–13.

[32] Kazufumi Ito and Kaiqi Xiong
"Gaussian Filters for Nonlinear Filtering Problems,"
IEEE Transactions on Automatic Control, vol. 45, no. 5, pp. 910–927, May 2000.

[33] Magnus Nørgaard, Niels K. Poulsen, and Ole Ravn
"New Developments in State Estimation for Nonlinear Systems,"
Automatica, vol. 36, no. 11, pp. 1627–1638, 2000.

[34] Marco F. Huber
"Chebyshev Polynomial Kalman Filter,"
Digital Signal Processing, vol. 23, no. 5, pp. 1620–1629, Sep. 2013.

[35] Michael Roth and Fredrik Gustafsson
"An Efficient Implementation of the Second Order Extended Kalman Filter,"
in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Jul. 2011.

[36] Oliver C. Schrempf and Uwe D. Hanebeck
"Dirac Mixture Approximation for Nonlinear Stochastic Filtering,"
in *Informatics in Control, Automation and Robotics—Selected Papers from the International Conference on Informatics in Control, Automation and Robotics 2007*, ser. Lecture Notes in Electrical Engineering. Springer, Sep. 2008, vol. 24, pp. 287–300.

[37] Uwe D. Hanebeck, Marco F. Huber, and Vesa Klumpp
"Dirac Mixture Approximation of Multivariate Gaussian Densities,"
in *Proceedings of the 2009 IEEE Conference on Decision and Control (CDC 2009)*, Shanghai, China, Dec. 2009.

[38] Igor Gilitschenski and Uwe D. Hanebeck
"Efficient Deterministic Dirac Mixture Approximation of Gaussian Distributions,"
in *Proceedings of the 2013 American Control Conference (ACC 2013)*, Washington D.C., USA, Jun. 2013.

[39] Wolfgang Härdle and Léopold Simar
Applied Multivariate Statistical Analysis,
2nd ed. Berlin Heidelberg: Springer, 2008.

[40] Jorge Nocedal
"Updating Quasi-Newton Matrices with Limited Storage,"
Mathematics of Computation, vol. 35, no. 151, pp. 773–782, Jul. 1980.

[41] Dong C. Liu and Jorge Nocedal
"On the Limited Memory BFGS Method for Large Scale Optimization,"
Mathematical Programming, vol. 45, no. 1–3, pp. 503–528, Aug. 1989.

[42] Uwe D. Hanebeck and Oliver C. Schrempf
"Greedy Algorithms for Dirac Mixture Approximation of Arbitrary Probability Density Functions,"
in *Proceedings of the 2007 IEEE Conference on Decision and Control (CDC 2007)*, New Orleans, USA, Dec. 2007, pp. 3065–3071.

[43] Bruno O. S. Teixeira, Leonardo A. B. Tôrres, Luis A. Aguirre, and Dennis S. Bernstein
"On Unscented Kalman Filtering with State Interval Constraints,"
Journal of Process Control, vol. 20, no. 1, pp. 45–57, 2010.

[44] Raymond Kan
"From Moments of Sum to Moments of Product,"
Journal of Multivariate Analysis, vol. 99, no. 3, pp. 542–554, Mar. 2008.

[45] Ondrej Straka, Jindrich Dunik, and Miroslav Simandl
"Gaussian Sum Unscented Kalman Filter with Adaptive Scaling Parameters,"
in *Proceedings of the 14th International Conference on Information Fusion (Fusion 2011)*, Chicago, USA, Jul. 2011, pp. 1–8.

[46] Rudolph van der Merwe and Eric A. Wan
"The Square-Root Unscented Kalman Filter for State and Parameter-Estimation,"
in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01)*, vol. 6, Salt Lake City, USA, May 2001, pp. 3461–3464.

Jannik Steinbring received his Dipl.-Inform. in computer science from the Karlsruhe Institute of Technology (KIT), Germany, in 2012. Currently, he is working towards a Ph.D. degree at the Intelligent Sensor-Actuator-Systems Laboratory, Karlsruhe Institute of Technology (KIT), Germany. His research interests are in the fields of nonlinear state estimation and extended object tracking.

Uwe D. Hanebeck is a chaired professor of Computer Science at the Karlsruhe Institute of Technology (KIT) in Germany and director of the Intelligent Sensor-Actuator-Systems Laboratory (ISAS). Since 2005, he is the chairman of the Research Training Group RTG 1194 "Self-Organizing Sensor-Actuator-Networks" financed by the German Research Foundation. Prof. Hanebeck obtained his Ph.D. degree in 1997 and his habilitation degree in 2003, both in Electrical Engineering from the Technical University in Munich, 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. Research results are applied to various application topics like localization, human-robot-interaction, assistive systems, sensor-actuator-networks, medical engineering, distributed measuring system, and extended range telepresence. Research is pursued in many academic projects and in a variety of cooperations with industrial partners.

Uwe D. Hanebeck was the General Chair of the "2006 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)," Program Co-Chair of the "11th International Conference on Information Fusion (Fusion 2008)," Program Co-Chair of the "2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008)," Regional Program Co-Chair for Europe for the "2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)," and will be General Chair of the "19th International Conference on Information Fusion (Fusion 2016)." He is a Member of the Board of Directors of the International Society of Information Fusion (ISIF), Editor-in-chief of its *Journal of Advances in Information Fusion (JAIF)*, and associate editor for the letter category of the *IEEE Transactions on Aerospace and Electronic Systems (TAES)*. He is author and coauthor of more than 300 publications in various high-ranking journals and conferences.

An Exploration of the Impacts of Three Factors in Multimodal Biometric Score Fusion: Score Modality, Recognition Method, and Fusion Process

YUFENG ZHENG
ERIK BLASCH

Operational applications for human identification require high credibility in order to determine or verify a person's identity to a desired confidence level. Multimodal biometric score fusion (MBSF) can significantly improve detection, recognition, and identification performance of humans. The goals of this research are to explore the impact of each factor in a MBSF process and to determine the most important (key) factor. The following are three main factors that will be investigated and discussed in this paper: score modality, recognition method, and fusion process. Specifically, score modality is defined as imaging device (hardware) for biometric data acquisition. Recognition method is defined as matching algorithm (software) for biometric score calculation. A fusion process such as arithmetic fusion, classifier-based fusion, or density-based fusion, is used to combine biometric scores. The hidden Markov model (HMM) is also applied to the MBSF process as a baseline comparison. The accuracy of human identification is measured with a verification rate. A new metric, *relative rate increase* (RRI), is proposed to evaluate the performance improvement using score fusion. Several recognition methods (two to four matchers) and four fusion processes (mean, linear discriminant analysis, k-nearest neighbors, and HMM) are compared over four multimodal databases in our experiments. The experimental results show that the score modality is the dominant factor in biometric score fusion. The fusion process becomes more important in a single modality fusion. Adding more recognition methods into the fusion process has the least impact on fusion improvement.

Manuscript received December 11, 2013; revised June 10, 2014 and October 1, 2014; released for publication October 2, 2014.

Refereeing of this contribution was handled by Ramona Georgescu.

This work was supported by the Department of Defense Research and Education Program (under Grant No. W911NF-12-1-0065) administered by the Army Research Office and the Office of Naval Research.

Authors' addresses: Y. Zheng, Alcorn State University, 1000 ASU Drive, Alcorn State, MS 39096, USA (e-mail: yzheng@alcorn.edu). E. Blasch, US Air Force Research Laboratory (AFRL), 525 Brooks Rd, Rome, NY 13441, USA (e-mail: erik.blasch@us.af.mil).

1557-6418/14/$17.00

1 INTRODUCTION

Data fusion can be performed at different levels, e.g., pixel, feature, score, and decision. Accordingly, the corresponding data preprocessing is also different for each level. For example, pixel-level and feature-level image fusion usually require registration and normalization (to align multiple images); while score-level fusion only requires normalization. Decision-level fusion such as a majority voting probably has the least requirements for preprocessing as the results are compiled from scores. The scope of fusion discussed in this paper is focused on multimodal biometric score fusion (MBSF). The source scores may originate from different types of devices, called *modality* (e.g., fingerprints, face images), and/or from variant analysis software, called *matcher* or *recognition* (e.g., linear discriminant analysis algorithm, Elastic Bunch Graphing Method (EBGM) algorithm) for face recognition. Score-level fusion usually involves score normalization, score fusion, and decision fusion. Score normalization (refer to Section 2.1) and decision fusion (refer to Section 3.2.2) may have some effects on the results of score fusion; however, the impacts of score modality (related with hardware), recognition method (software), and fusion process (post-processing in a hybrid human identification system) will be emphasized and investigated in this paper.

There are several types of score fusion methods: arithmetic combination of fusion scores, classifier-based fusion, and density-based fusion. In arithmetic fusion, the final score is a value of predefined function, f, with the input of normalized scores, $(s_1, s_2, \ldots)$. The output of such a fusion process, S_{F}, is computed by

$$S_{\mathrm{F}} = f(s_1, s_2, \ldots, s_n), \tag{1}$$

where f stands for a fusion function or a set of fusion rules. f may be implemented by a simple arithmetic operation [15] such as taking the summation, average, product, minimum, maximum, median, majority vote, or by exploiting a Naive Bayes model [16]. In classifier-based fusion (referred to as *classifier fusion*), a classifier is first trained with the labeled score data, and then tested with unlabeled scores [4], [9]. The choices of classifiers include linear discriminant analysis (LDA) [8], k-nearest neighbors (KNN), artificial neural network (ANN) [14], and/or a support vector machine (SVM) [6]. In density-based fusion, a multi-dimensional density function is estimated with the score dataset, and then it can predict the probability of any given score vector [23], [28]. Nandakumar et al. [21] proposed a density-based fusion method where the likelihood ratio was estimated by Gaussian mixture model (GMM). Their experimental results [21] showed that the likelihood ratio fusion outperformed any single matcher and other fusion processes (like sum rule with min-max). A *Hidden Markov model* (HMM) was recently proposed for MBSF (referred to as *HMM fusion* [31]), which can

flexibly combine multiple scores from different modalities and/or from variant matchers. The early experimental results [31] showed that the HMM fusion was the most accurate and credible method in comparison to mean fusion and KNN fusion.

The security applications of a human identification system require achieving greater accuracy, efficiency, and credibility to robustly determine a person's identity (ID). It is clear that the MBSF process can significantly improve human identification performance [13], [21], [26], [27], [30], [31]. The set of literature focused on the advances of specific score fusion methods and its performance improvement typically evaluate the complete system. For example, Toh et al. [27] introduced a reduced multivariate polynomial model for multimodal biometric decision fusion (using three scores from fingerprint, speech and hand geometry), and they found that local learning and global decision did better than just fusing all three results at once. Ross and Jain [26] conducted a set of experiments in combining multimodal biometric scores (from face, fingerprint, and hand geometry), and their results indicated that the sum rule performed better than the decision tree and linear discriminant classifiers. Our early work [31] also focused on the discussion of performance improvement with the HMM fusion method. To the authors' knowledge, there are few published works that explore the key parameters that influence score fusion. Part of the reason may be lack of multimodal score databases and no effective metrics for fusion improvement evaluations across different methods and databases. Recognizing the key factor of score fusion will help design an accurate and credible human ID system to meet the critical needs of security applications. For instance, assuming that a human ID system permits a fusion with only two scores, should two modalities (one matcher per modality, e.g., fingerprint and face), or two matchers on one modality (e.g., fingerprint) be used? What is the impact of various fusion processes (e.g., HMM versus KNN) over different scenarios?

The main purpose of this research is to discover the key factor of multimodal biometric score fusion. Four fusion methods, mean, LDA, KNN, and HMM, are tested and compared using four biometric score datasets, wherein the HMM fusion is specifically configured for score fusion. Additionally, a new metric (called *relative rate increase*) is introduced for fusion improvement measurement. Our experiments reveal that score modality is the key factor in a score fusion scenario, which is meaningful to integrate and configure a multimodal biometric system. The rest of this paper is arranged as follows. The score normalization and fusion evaluation are depicted in Section 2. The score fusion processes including HMM fusion are described in Section 3. Experimental results, comparisons, and discussions are presented in Section 4. Finally, conclusions are drawn in Section 5.

2 SCORE NORMALIZATION AND FUSION EVALUATION

Multimodal biometric scores are computed with different modalities and algorithms, which may be similarity values (e.g., confidence values, probabilities, or logarithm probabilities), or distance measures (e.g., Euclidian distance, Hamming distance, or Mahalanobis distance). The variant source scores may contrast in a variety of ranges. Score normalization is required before score fusion. Meanwhile, fusion evaluation is needed to compare the performance of different fusion processes. To evaluate fusion performance, it is required that all original scores are either similarity scores or distance scores (but not the mix of similarity and distance). Converting a similarity score to a distance score is straightforward because of their *reciprocal* relationship.

2.1 Score Normalization

Prior to score fusion, score normalization is expected since the multimodal scores are heterogeneous and thus have varying dynamic ranges. The large variances of multimodal scores are caused either by different matching algorithms or by different natures of biometrical data. There are many normalization methods proposed in literature. Jain et al. [13] reported that min-max, z-score, and tanh normalization techniques, followed by a simple sum of scores fusion method, resulted in a superior GAR (genuine accept rate). It was also shown that both min-max and z-score methods are sensitive to outliers; whereas the tanh normalization method is both robust and efficient. The score data used in our experiments were obtained in the controlled lab environment (with less noise), thus a standard z-score *normalization* procedure is applied to all biometric scores,

$$\mathbf{S}_{\mathrm{N}} = (\mathbf{S}_0 - \boldsymbol{\mu}_0)/\boldsymbol{\sigma}_0, \tag{2}$$

where $\mathbf{S}_N$ is the normalized score vector, $\mathbf{S}_0$ is the original score vector, and $\boldsymbol{\mu}_0$ and $\boldsymbol{\sigma}_0$ denote the mean and standard deviation of original scores, respectively.

2.2 Fusion Evaluation

2.2.1 Verification Rate.

Genuine score is the matching score resulting from two samples of one user; while *impostor score* is the matching score of two samples originating from different users. *Genuine accept rate* (GAR) is the fraction of genuine scores exceeding the threshold; whereas *false accept rate* (FAR) is the fraction of impostor scores exceeding the threshold. A receiver operating characteristic (ROC) curve is computed from the FAR and true positive rate (TPR). On an open dataset (the query user may not be contained in the database), GAR/FAR/ROC area can be computed by choosing a threshold. On a closed dataset (the query user is surely included in the database), the identification performance can be measured by a *verification rate* (also called identification

rate or recognition rate), denoted as R_V, the percentage of correctly identified users over the total number of users. In our experiments, verification rate (VR) is used to evaluate the fusion performance since all users (i.e., subjects) are guaranteed in the database. Of course, the VR value may vary with a preset threshold. In a single-matcher evaluation, top-1 matching (e.g., the shortest distance) is used, while in a score fusion evaluation, the default threshold of each classifier is used. Finally, keep in mind that it is necessary to convert all multimodal scores to either similarity scores or distance scores before score fusion.

2.2.2 Relative Rate Increase.

The performance improvement using score fusion cannot be properly measured by using the absolute difference of two verification rates. For example, improving R_V from 80% to 90% seems to be more difficult than the improvement from 98% to 99%. Generally speaking, we know that the improvement of R_V via score fusion becomes more and more difficult when the original rate is approaching 100%. We propose to use a *relative rate increase* (denoted as RRI) to evaluate the fusion improvement.

$$\text{RRI} = \frac{\text{ARI}}{1-\overline{R_V}} = \frac{R_F - \overline{R_V}}{1-\overline{R_V}}, \tag{3}$$

where R_F is the verification rate via score fusion; $\overline{R_V}$ is the mean of original verification rates from individual modalities or matchers. $\text{ARI} = R_F - \overline{R_V}$ is the *absolute rate increase* (ARI), which may not precisely measure the performance improvement as stated above. $\text{RRI} \in (0,1]$; the higher, the better. According to the RRI definition, two fusion improvements, from 80% to 90% and from 98% to 99%, are equivalent, and their RRI = 0.50. It may be understood that the two improvements are "equivalent" in the sense of their difficulty levels and/or of the extent of their effort.

Many metrics could be devised, wherein the RRI metric seeks to measure the actual improvement against the total amount of possible improvement. With future large databases, the RRI metric would help in the quality of the fusion performance over the entire dataset (versus an assumed recognition performance with a small data set).

3 SCORE FUSION PROCESSES

In this section, arithmetic fusion and classifier fusion are briefly reviewed, and then HMM models are introduced for biometric score fusion.

3.1 Arithmetic Fusion and Classifier Fusion

Arithmetic fusion means to combine multiple scores by taking the summation, average (mean), product (called geometric mean), minimum, maximum, and median [15]. Majority vote is actually a kind of decision-level fusion, which requires the number of decision makers to be an odd number to avoid a possible tie. The mean fusion is selected in our experiments because it has the best performance of all aforementioned arithmetic fusion processes.

In *classifier fusion*, four frequently-used classification methods are discussed. These methods include linear discriminant analysis (LDA), k-nearest neighbor (KNN), artificial neural network (ANN), and support vector machine (SVM). The fusion results of LDA and KNN methods will be presented in our experiments due to their better performance on average [33], and thus these two methods are briefly described as follows, where the reader can find descriptions of ANN and SVM in the literature. The purpose of LDA is to predict group membership based on a linear combination of a set of predictor variables (i.e., a feature vector) [8]. The end result of the LDA procedure is a model (i.e., linear discriminant function, LDF) that allows prediction of group membership when only the predictor variables are known. The KNN method is usually deployed with a clustering technique. Fuzzy C-means (FCM) [3] is a data clustering technique wherein each data point belongs to a cluster to some degree that is specified by a membership grade. FCM starts with an initial guess of data membership and iteratively moves the cluster centers to the correct location within a data set. Once a certain number of clusters are formed by the FCM algorithm, the k-nearest neighbors can be found from those clusters using a Euclidean distance (between a testing feature vector and the clustered feature vectors). The probability of a given feature vector (multimodality scores) can be calculated with the labeled clusters.

To sufficiently use the sample data in classification evaluation, a *cross validation* method is applied to split original data into two groups for training and testing purposes. K-fold cross validation [25] is ideal for small databases. Notice that the divisions of k subsets ($k = 10$ used in our experiments) are based upon the users. If one user is grouped into Subset 1, then all scores of that user (including all his/her genuine and impostor scores) belong to Subset 1.

3.2 Hidden Markov Model for Multimodal Score Fusion

3.2.1 Basics on Hidden Markov Models.

In the past two decades, HMM models have emerged as a powerful tool for modeling stochastic processes and pattern sequences. Originally, HMMs have been applied to the domain of speech recognition and have become the dominating technology [24]. In recent years, they have attracted growing interest in computational molecular biology, bioinformatics, mine detection [12], handwritten character/word recognition [19], face and gesture recognition, shape recognition, image database retrieval, and other computer vision applications [5]. Generally speaking, an HMM is a model of a stochastic process that produces a sequence of random observa-

tion vectors at discrete times according to an underlying *Markov chain*. At each observation time, the Markov chain may be in one of N states $\{s_1,\ldots,s_N\}$ (hidden from the observation) and, given that the chain is in a certain state, there are probabilities of moving to other states, called the transition probabilities. An HMM is characterized by three sets of probability density functions: the *state transition probabilities* (**A**), the *observation symbol probabilities* (**B**), and the *initial state probabilities* ($\boldsymbol{\pi}$).

Let T be the length of the observation sequence (i.e., number of time steps; $t = 1,\ldots,T$), $\mathbf{O} = \{\mathbf{O}_1,\ldots,\mathbf{O}_T\}$ be the observation sequence, and $\mathbf{Q} = \{\mathbf{q}_1,\ldots,\mathbf{q}_T\}$ be the state sequence. The compact notation,

$$\lambda = (\mathbf{A},\mathbf{B},\boldsymbol{\pi}) \tag{4}$$

is generally used to indicate the complete parameter set of the HMM model, λ. In the above, $\mathbf{A} = \{a_{ij}\}$ is the state transition probability matrix, where $a_{ij} = P(q_t = s_j \mid q_{t-1} = s_i)$ for $i,j = 1,\ldots,N$; $\boldsymbol{\pi} = \{\pi_i\}$, where $\pi_i = P(q_1 = s_i)$, are the initial state probabilities. In the case of the *discrete HMM*, the observation vectors are commonly quantized into a finite set of symbols, $\{\mathbf{v}_1,\ldots,\mathbf{v}_M\}$ called the *codebook*. Each state is represented by a discrete probability density function and each symbol has a probability of occurring given that the system is in a given state. The observation symbol probability distribution $\mathbf{B} = \{b_i(\mathbf{O}_t)\}$ becomes a simple set of fixed probabilities for each class, i.e., $b_i(\mathbf{O}_t) = b_i(k) = P(\mathbf{v}_k \mid q_t = s_i)$, where $\mathbf{v}_k$ is the symbol of the nearest codebook of $\mathbf{O}_t$.

Three key problems [24] must be solved for the model defined in Eq. (4) to be useful in real world applications: the classification (*testing*) problem, the problem of finding an optimal state sequence (*tuning*), and the problem of estimating the model parameters (*training*). The classification problem involves computing the probability of an observation sequence $\mathbf{O} = \{\mathbf{O}_1,\ldots,\mathbf{O}_T\}$ given a model λ, i.e., $P(\mathbf{O} \mid \lambda)$. The Viterbi algorithm [20] is an efficient and formal technique for finding this maximum (optimal) state sequence and associate probability. The third problem is the training problem, i.e., how does one estimate the parameters of the model? First, all the states themselves must be estimated. Then the model parameters need to be estimated. In the discrete HMM, the codebook is first determined, usually using clustering techniques such as K-means [7] or fuzzy C-mean clustering algorithms [3]. The probability distribution **B** may be estimated either by fuzzy memberships [3] in a discrete HMM model, or by Gaussian mixture model (GMM) [10], [21] in a continuous HMM model. Then the parameters $(\mathbf{A},\mathbf{B},\boldsymbol{\pi})$ are estimated iteratively using the Baum-Welch algorithm [2].

3.2.2 HMMs for Multimodal Score Fusion.

The HMM fusion is a type of classifier fusion, but it significantly differs in data preparation and classification process. In the context of this paper, we need to distinguish two terms, multimodal scores and multi-matcher scores. *Multimodal* biometric scores (also referred to as *inter-modality* scores) result from different modalities (such as different hardware devices for imaging face and fingerprint); while *multi-matcher* scores (also referred to as *intra-modality* scores) result from different software algorithms but use the same modality (e.g., three face scores generated from three face recognition algorithms, respectively).

For HMM training, a large database with known users (labeled with user IDs) are expected, and thus a k-fold cross validation is utilized to satisfy this need. All scores are normalized and then organized as the inputs of HMM models using k-fold cross validation. The HMM model is adapted to a MBSF process and initialized with parameters like HMM(m,n,g), or denoted as $m \times n \times g$ HMM. Where m is the number of intra-modality scores (from m matchers upon one modality data) representing an *observation vector* in HMM, and n is the number of modalities corresponding to n hidden states, respectively. By placing n pieces of m-dimension observation vectors together, an observation sequence (over time, t) is formed. g is the number Gaussian components per state in a Gaussian mixture model (GMM). The GMM is applied to estimate the state probability density functions of each hidden state in a continuous HMM model.

Two HMM models are derived using genuine scores and impostor scores (in the training dataset), respectively. Given an observation sequence formed with multiple scores (of dimension $m \times n$) in the testing dataset, the two trained HMM models can compute the probabilities of being a genuine user and an impostor user, respectively. The user is identified as genuine if the probability given by the genuine HMM is higher. The details of the HMM model [31] and its adaption to a MBSF process are described as follows.

3.2.3 HMM Adaption to Multimodal Score Fusion.

The HMM models have wide applications in different fields and require proper data initialization for a specific application. In HMM score fusion, the observation vector $\mathbf{O}_t$ can be the m-dimensional intra-modality scores from m matchers. The observation sequence $\mathbf{O}(t,s)$ can be formed by combining n pieces of $\mathbf{O}_t$ from n modalities: $\mathbf{O}(t,s) = \{\mathbf{S}_{\mathrm{mn}}\}$. For example, there are 2 biometric modalities ($n = 2$; e.g., face, finger) and 2 matching algorithms (matchers) for each modality ($m = 2$). Thus, the length of $\mathbf{O}(t,s)$ is 4 (refer to NIST-Face-Fingerer database in Table 1a). The elements of **B** can be initialized with GMM, where the number of Gaussian models (g) in each state are usually fixed (e.g., $g = 3$) or automatically decided [10]. Notice that two HMM models, λ_{Gen} and λ_{Imp}, are actually trained using genuine scores and impostor scores, respectively; where their parameters can be estimated

TABLE 1a
Summary of four multimodal biometric databases.

Database	No. of Modalities	No. of Matchers	No. of Users	No. of Images	No. of Scores
NIST-Face-Finger	2: Finger & Face	4	517	3,102	1,069,156
NIST-Finger-Finger	1: Finger	2	6,000	24,000	72,000,000
NIST-Face	1: Face	2	3,000	6,000	36,000,000
ASUMS-Face-Face	2: Face (IR & DC)	6	96	576	110,592

TABLE 1b
Details of the trimmed databases (Sim. = Similarity; Dist. = Distance).

Database	Genuine : Impostor	HMM Models	Matchers	Score Type	Plot
NIST-Face-Finger_M1 (Face)	1,034 : 2,068	$2 \times 1 \times 3$	2 Face matchers	Sim., Sim.	Fig. 2a
NIST-Face-Finger_M2 (Finger)	1,034 : 2,068	$2 \times 1 \times 3$	Left, Right Finger	Sim., Sim.	Fig. 2b
NIST-Finger-Finger	12,000 : 24,000	$2 \times 1 \times 3$	Left, Right Finger	Sim., Sim.	Fig. 2c
NIST-Face	6,000 : 12,000	$2 \times 1 \times 5$	2 Face matchers	Sim., Sim.	Fig. 2d
ASUMS-Face-Face_M1 (IR)	576 : 1,152	$3 \times 1 \times 2$	FPB, LDA, EBGM	Dist., Dist., Sim.	Fig. 3a
ASUMS-Face-Face_M2 (DC)	576 : 1,152	$3 \times 1 \times 2$	FPB, LDA, EBGM	Dist., Dist., Sim.	Fig. 3b

Fig. 1. Sample faces from the ASUMS-Face-Face database: Notice that the two images (DC/visible, IR/thermal) shown at two neighboring columns were acquired from the same subject. The images are the aligned faces (320×320 pixels).

using the Baum-Welch algorithm [2]. An unlabeled biometric score sequence, $\mathbf{O}$, will be classified as a "genuine user" if $P_{\text{Gen}}(\mathbf{O} \mid \lambda_{\text{Gen}}) > P_{\text{Imp}}(\mathbf{O} \mid \lambda_{\text{Imp}}) + \eta$ (a simple decision rule); otherwise, $\mathbf{O}$ will be an "impostor user," where η is a small positive number empirically decided by experiments.

$$\mathbf{O} = \begin{cases} \text{Genuine User} & \text{if } P_{\text{Gen}}(\mathbf{O} \mid \lambda_{\text{Gen}}) > P_{\text{Imp}}(\mathbf{O} \mid \lambda_{\text{Imp}}) + \eta \\ \text{Impostor User} & \text{Otherwise} \end{cases} \tag{5}$$

In general, $m \geq 1$, $n \geq 1$, and $m \times n \geq 2$ are expected. In other words, at least two scores are required for HMM fusion. If the number of biometric modality is one ($n = 1$), then the number of matching scores from that modality must be two or greater (produced from different matching algorithms, e.g., LDA and EBGM [29] for face recognition). If there are two or more modalities ($n \geq 2$), in order to properly initialize and train the HMM models, the numbers of intra-modality scores ($m \geq 1$) derived from each modality must be same. There are usually more impostor scores than genuine scores in a biometric score dataset. To prevent a HMM model from being biased by the excessive impostor scores, the number of impostor scores used in training λ_{Imp} should be equivalent to the number of genuine scores used in training λ_{Gen}.

4 EXPERIMENTAL RESULTS AND DISCUSSIONS

The MBSF experiments were conducted on four biometric score databases and evaluated by reporting the verification rates (R_{V} and R_{F}) and the values of relative rate increase (RRI). Four fusion processes were selected to be reported in our experiments because of their better performance on average. The four fusion processes include one arithmetic fusion (mean fusion), two classifier fusions (LDA fusion and KNN fusion), and HMM fusion [31]. In the context, "modality" represents a biometric device (fingerprint, face); "matcher" is the software implementation of a "recognition method"; and "fusion method" means how to combine multiple scores (e.g., KNN fusion, HMM fusion). In the following discussion, Row 1 (or Column 1) referring to a table means the 1st row (or column) after the header row (or column).

4.1 Multimodal Scores and Experimental Design

Four biometric score databases (see Table 1a) were used in our experiments; three of which were from NIST-BSSR1 (Biometric Scores Set Release 1, from National Institute of Standards and Technology) [22], [31], and one of which was the face scores generated in our lab. Specifically, as shown in Table 1a, the *NIST-Face-Finger* database consists of a total of 1,069,156 biometric scores that were computed with 3,102 images from 517 users (individuals). Two face images, two left index fingerprints (images), and two right index fingerprints were acquired from each user; and then two face matching systems and one fingerprint matching system were applied to those images, respectively. So

TABLE 2a
The verification rates (%) of four fusion processes (R_F) across four databases.

Database	Single Matcher (R_V)	Mean Fusion	LDA Fusion	KNN Fusion	HMM Fusion (m,n,g)
NIST-Face	77.50, *81.02*	81.88	92.28	96.82	97.01 (2,1,5)
NIST-Finger-Finger	80.52, *87.88*	93.98	97.60	92.29	98.16 (2,1,3)
NIST-Face-Finger	89.17, 84.33 86.46, *92.65*	99.61	99.10	99.55	99.68 (2,2,5)
ASUMS-Face-Face	91.67, 93.75, 96.88 90.63, 93.75, *97.92*	100.00	99.48	98.48	99.83 (3,2,2)

TABLE 2b
The relative rate increase (RRI) of four fusion processes across four databases. $\overline{R_V}$ is the averaged R_V of all matchers.

Database	$\overline{R_V}$	Mean Fusion	LDA Fusion	KNN Fusion	HMM Fusion	(μ_{RRI},σ_{RRI})
NIST-Face	79.26	0.1263	0.6278	0.8467	0.8558	0.6142, 0.3419
NIST-Finger-Finger	84.20	0.6190	0.8481	0.5120	0.8835	0.7157, 0.1794
NIST-Face-Finger	88.15	0.9671	0.9240	0.9620	0.9730	0.9565, 0.0221
ASUMS-Face-Face	94.10	1.0000	0.9119	0.7424	0.9712	0.9064, 0.1153
(μ_{RRI},σ_{RRI})	(NA)	0.6781, 0.4062	0.8279, 0.1375	0.7658, 0.1915	0.9209, 0.0602	(NA)

there are 4 genuine scores for each user, two scores from two face matching systems, two scores from one fingerprint system but running on two fingerprints (left and right). There are two modalities (finger and face) and a total of four matchers in the NIST-Face-Finger database, and thus two $2 \times 2 \times 5$ HMM models [31] (for genuine and impostor, respectively) were initialized ($g = 5$ gave the best performance when varying g from 2 to 7[1]). The *NIST-Finger-Finger* database contains the scores from one fingerprint system running on two fingerprints (left and right); and then two $2 \times 1 \times 3$ HMM models were established. The *NIST-Face* database is comprised of the scores from two face matching systems; and two $2 \times 1 \times 5$ HMM models were created.

The *ASUMS-Face-Face* (Alcorn State University [ASU] MultiSpectral) database (Row 4 in Table 1a) includes the scores from three face recognition algorithms and from two modalities ASUIR (ASU long-wave Infrared) face images and ASUDC (ASU Digital Camera) face images (see Fig. 1). Three face recognition algorithms are linear discriminant analysis (LDA) [18], elastic bunch graph matching (EBGM) [29], and face pattern byte (FPB) [32]. The corresponding HMM models were configured as $3 \times 2 \times g$ (refer to Table 2a). The *ASUIR-Face* subset [32] includes thermal (long-wave infrared, IR) face images, whereas the *ASUDC-Face* subset consists of visible (digital camera, DC) face images from the same group of users. In these two sub-datasets, 3 face images were acquired from each user, where one randomly-selected image was used as *probe face* (i.e., a face image from a live camera) and the other two as *gallery faces* (i.e., face images from a database). Table 1b shows the comparative relations over the trimmed datasets (of reduced impostor scores) between numbers of genuine to impostor scores, parameters of HMM models ($m \times n \times g$), matchers, score type, and plots (also refer to Tables 2a, 3a, 4a).

The total number of scores is massive in that it mainly contains impostor scores. For example, ASUMS-Face-Face consists of 1,152 genuine scores and 109,440 impostor scores (for all 3 matchers and 2 modalities). All scores are normalized by using Eq. (2). An unbalanced training with the excessive impostor scores may result a biased or over-trained model. To avoid possible bias in model training as mentioned in Section 3, two impostor scores per matcher per user were randomly selected for training. All genuine scores plus reduced impostor scores are called "trimmed database." Arithmetic fusion used all scores (full databases), whereas HMM fusion and classifier fusion used *trimmed databases* (refer to Table 1b). The distributions of normalized scores of four trimmed databases are presented as scatter plots in Figs. 2–3, where the x-axis denotes Score 1 and the y-axis represents other scores. The distributions of 3 scores shown in Fig. 3 indicate low correlation of three scores. Figs. 2–3 also show that three NIST databases contain similarity scores (genuine scores are large); while the ASUMS database includes both similarity scores and distance scores (genuine scores are small). Notice that less impostor markers shown in Fig. 2c is because most impostor markers are behind (thus blocked by) the genuine markers.

Four fusion processes were tested across the four score databases (refer to Table 1a). The fusion results of mean fusion, LDA (with quadratic kernel), KNN, and HMM were reported in Table 2a. The HMM models were implemented and adapted upon the "Hidden Markov Model (HMM) Toolbox for Matlab" [20]. All HMM models were tested by varying the number of Gaussian components (g) from 2 to 7, the best results of HMM fusions (together with initialization parameters) are shown in Table 2a. The verification rates of original scores are presented in Column "Single Matcher" in

[1] g was determined empirically in the experiment from which the differences of g had a marginal impact on the results.

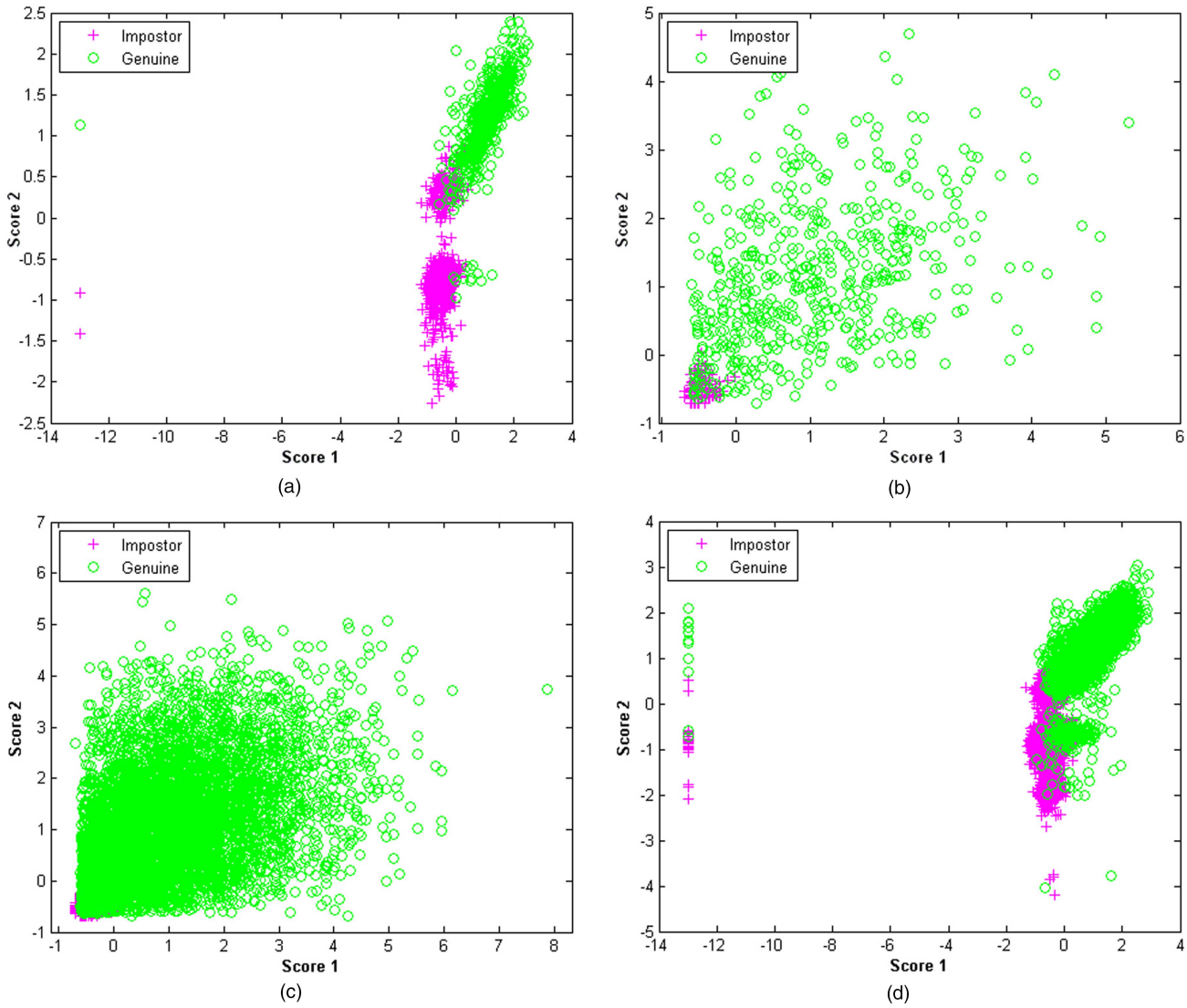


Fig. 2. Scatter plots (score distributions) of normalized multimodal biometric scores from three NIST databases (trimmed), where the x-axis is Score 1 and the y-axis is Score 2: (a) NIST-Face-Finger_M1 (Face); (b) NIST-Face-Finger_M2 (Finger); (c) NIST-Finger-Finger; (d) NIST-Face. Notice that all genuine scores and the two randomly-selected impostor scores per matcher per user are presented.

Table 2a, where the rightmost number (in *italic* font) is the single best performance.

4.2 Results and Discussions

The performance of individual matcher (R_V) and four fusion processes (R_F) on four databases are presented in Table 2a. It is clear that all four fusion approaches yield improvements compared to the corresponding single best matcher (SBM) on each database. Overall, the HMM fusion performs the best. It seems that the *mean fusion* performs very well on the multimodal databases (99.61% on NIST-Face-Finger and 100% on ASUMS-Face-Face). The possible reason might be that the genuine scores and the impostor scores on these two databases are well separated (refer to the score distributions shown in Figs. 2–3), which makes a linear separation (like mean fusion) ideal. Surprisingly, in another independent research [30], the weighted-sum score fusion reached the highest rate of 99% (SBM = 97%) when two weights were equal, which turned out to be a mean fusion (but the score distributions were not presented). The level of improvement will be analyzed using the values of *relative rate increase* (RRI).

The RRI values of four fusion processes are given in Table 2b. Table 2a and Table 2b are corresponding cell-by-cell except for the last row and the last column. Let us examine the rationality of RRI, which is proposed to measure the improvement of score fusion. The RRI value of mean fusion on NIST-Face is 0.1263 (the smallest value in Table 2b), which corresponds an absolute rate increase (ARI = 2.62%) from 79.26% ($\overline{R_V}$) to 81.88%. The RRI value of HMM fusion on NIST-Face-Finger is 0.9730 (the second largest value in Table 2b), which corresponds ARI = 11.53% (from 88.15% to 99.68%). There is a special case, RRI = 1.0000, for the mean fusion on ASUMS-Face-Face, which represents a

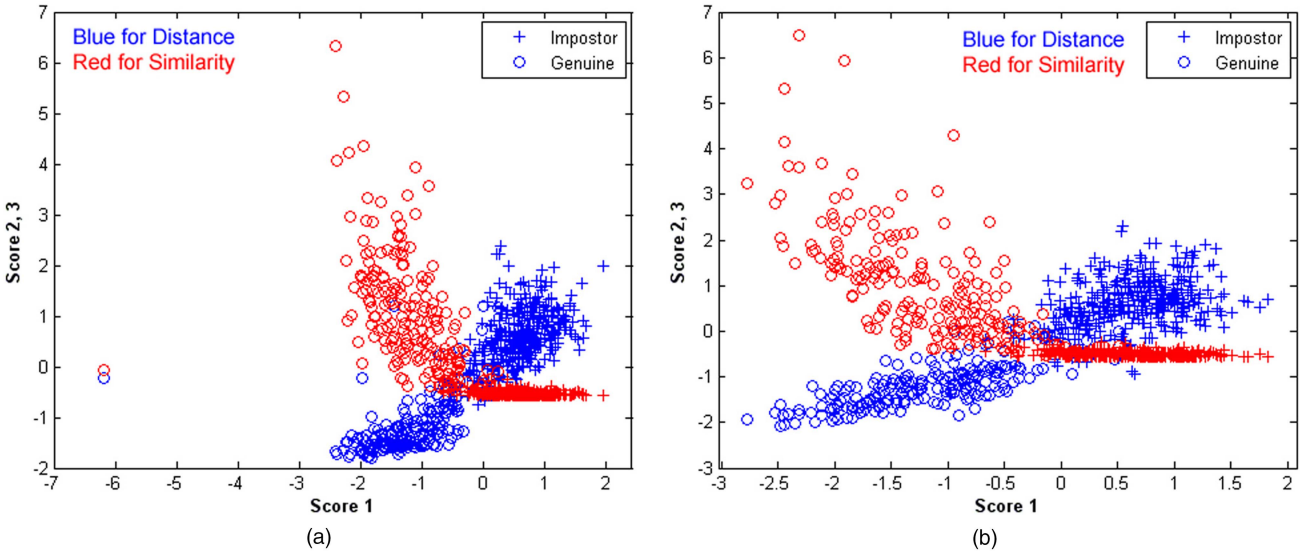


Fig. 3. Scatter plots (score distributions) of normalized face scores from two ASUMS datasets (trimmed), where the x-axis shows Score 1 from FPB (distance score), and the y-axis represents Score 2 from LDA (distance score shown in blue) and Score 3 from EBGM (similarity score shown in red): (a) Scores from ASUMS-Face-Face_M1 (IR); (b) Scores from ASUMS-Face-Face_M2 (DC). Notice that all genuine scores and the two randomly-selected impostor scores per matcher per subject are presented.

verification rate increase from 94.10% to 100%. Since $R_F = 100\%$ means a perfect fusion (i.e., a perfect human identification system), it is reasonable for RRI to take its maximum value, 1.0. On a large database (e.g., millions of users), RRI rarely reaches 1.0. According to the definition of RRI in Eq. (3), improving R_F from 90% to 100% (ARI = 10%) and from 99.9% to 100% (ARI = 0.1%), both will have RRI = 1.0, which makes sense in terms of difficulty or effort. In other words, the level of difficulty or the amount of effort for both cases may be equivalent.

In Table 2b, the means and standard deviations of RRI, denoted as $(\mu_{RRI}, \sigma_{RRI})$, in each row and in each column are listed in the last column and in the last row, respectively, where "NA" means not applicable. As shown in the bottom row of Table 2b, when averaging across four biometric databases, the HMM fusion has the highest μ_{RRI} and also the least σ_{RRI}. We may conclude that the *HMM fusion* is the best for MBSF in terms of accuracy (high improvement) and credibility (low variance). The LDA fusion is the second best. According to the rightmost column in Table 2b, when averaging across four fusion processes, the NIST-Face-Finger database gives the highest μ_{RRI} (0.9565) with the least σ_{RRI}. The ASUMS-Face-Face database is the second best ($\mu_{RRI} = 0.9064$). It is clear that *multimodal fusion* (NIST-Face-Finger and ASUMS-Face-Face, their averaged $\mu_{RRI} = 0.9314$) is superior to single-modal fusion (NIST-Finger-Finger and NIST-Face, their averaged $\mu_{RRI} = 0.6649$). It also makes sense that NIST-Face-Finger produces a better improvement than ASUMS-Face-Face since NIST-Face-Finger consists of truly diverse modalities (face and finger), whereas ASUMS-Face-Face contains two bands of face images (thermal and visible).

In Table 2b, μ_{RRI}(NIST-Face-Finger) = 0.9565 represents a *modality fusion* with multimodal scores when averaging fusions; μ_{RRI}(HMM_Fusion) = 0.9209 is from the best *fusion process* when mixing modalities and recognitions; and μ_{RRI}(NIST-Face) = 0.6142 is considered as a *recognition fusion* result with single-modal (face) scores when averaging fusions. It reveals that the importance of fusion factors from the highest to the lowest are as follows: score modality, fusion process, and recognition method. These three factors may interact with one another; however, we do not have sufficient data (power) to conduct an analysis of variance (ANOVA).

To investigate and verify key factor that influences the score fusion (i.e., sensitivity test), we need to separate three fusion factors: modality, recognition, and fusion. Thus two multimodal databases, NIST-Face-Finger and ASUMS-Face-Face, are selected, and divided into modality subsets (e.g., NIST-Face-Finger_M1 and NIST-Face-Finger_M2; refer to Table 1b) and recognition subsets (e.g., NIST-Face-Finger_R1 and NIST-Face-Finger_R2). For example, on NIST-Face-Finger_M1 (face), the mean fusion is performed by averaging the scores from two matchers (i.e., two recognition methods), which is used to research the impact of the recognition method. While on ASUMS-Face-Face_R2 (EBGM), the mean fusion is achieved by averaging the two EBGM scores from two modalities (i.e., IR and DC; see Fig. 1) and used to study the impact of score modality. The performance of individual matcher (R_V) and four fusion processes (R_F) on subsets are listed in Table 3a and Table 4a, and the relative rate increase (RRI) of four fusion processes on subsets and their $(\mu_{RRI}, \sigma_{RRI})$ are given in Table 3b and Table 4b. The following discussions are based on the results of RRI

TABLE 3a
The verification rates (%) of four fusion processes (R_F) on four subsets derived from the NIST-Face-Finger database.

Database	Single Matcher (R_V)	Mean Fusion	LDA Fusion	KNN Fusion	HMM Fusion (m,n,g)
NIST-Face-Finger_M1 (Face)	84.33, *89.17*	90.52	94.45	96.97	97.42 (2,1,3)
NIST-Face-Finger_M2 (Finger)	86.46, *92.65*	94.78	97.23	97.48	98.06 (2,1,3)
NIST-Face-Finger_R1 (Matcher 1)	84.33, 86.46	94.78	97.16	99.22	99.22 (1,2,2)
NIST-Face-Finger_R2 (Matcher 2)	89.17, 92.65	96.71	99.16	99.42	99.42 (1,2,3)

TABLE 3b
The relative rate increase (RRI) of four fusion processes on four subsets derived from the NIST-Face-Finger database.

Database	$\overline{R_V}$	Mean Fusion	LDA Fusion	KNN Fusion	HMM Fusion	(μ_{RRI},σ_{RRI})
NIST-Face-Finger_M1 (Finger)	86.75	0.2845	0.5811	0.7713	0.8053	0.6106, 0.2387
NIST-Face-Finger_M2 (Face)	89.56	0.5002	0.7348	0.7587	0.8143	0.7020, 0.1386
NIST-Face-Finger_R1 (Matcher 1)	85.39	0.6426	0.8055	0.9466	0.9466	0.8353, 0.1447
NIST-Face-Finger_R2 (Matcher 2)	90.91	0.6381	0.9076	0.9362	0.9362	0.8545, 0.1449
(μ_{RRI},σ_{RRI})	(NA)	0.5164, 0.1681	0.7573, 0.1372	0.8532, 0.1020	0.8756, 0.0762	(NA)

or μ_{RRI} since they can more properly evaluate the improvement than R_F.

As shown in Table 3b, μ_{RRI}(NIST-Face-Finger_R1) = 0.8353 and μ_{RRI}(NIST-Face-Finger_R2) = 0.8545 are significantly higher than μ_{RRI}(NIST-Face-Finger_M1) = 0.6106 and μ_{RRI}(NIST-Face-Finger_M2) = 0.7020, respectively. Further averaging the RRI values of two recognition subsets (Rows 3–4), we have μ_{RRI}(NIST-Face-Finger_Rn) = 0.8449, which is much higher than μ_{RRI}(NIST-Face-Finger_Mn) = 0.6563, where $n = 1,2$. The comparisons above indicate that *score modality* is a more important factor than *recognition method* (matcher) when averaging (or mixing) fusion processes. This statement complies with the conclusion from Table 2b, *multimodal fusion* is superior to single-modal fusion. Table 4b shows the same fact, where μ_{RRI}(ASUMS-Face-Face_Rn) = 0.6814 ($n = 1,2,3$) and μ_{RRI}(ASUMS-Face-Face_Mn) = 0.6574 ($n = 1,2$), although the difference is small as expected (due to less diversity in score modalities).

To further explore the impacts of fusion factors within one score database, the (μ_{RRI},σ_{RRI}) values of the combined modality subsets (Rows 1–2 in Table 3b and Rows 1–2 in Table 4b) and the (μ_{RRI},σ_{RRI}) values of the combined recognition subsets (Rows 3–4 in Table 3b and Rows 3–5 in Table 4b) are shown in Table 5. As seen before, the fusion of modalities is superior to the fusion of recognitions (matchers) with one exception (the LDA fusion on ASUMS-Face-Face database). The mean fusion results are not used in the following discussions due to their high variances (i.e., low credibility).

We shall make quantitative comparisons on NIST-Face-Finger database (Rows 1–2 in Table 5). When the fusion process is selected (fixed), for instance, with HMM fusion, the difference of μ_{RRI} values between the fusion of modalities and the fusion of recognitions is 0.1316, denoted as $\Delta\mu_{RRI}$(Modality, Recognition | HMM) = 0.9414 − 0.8098 = 0.1316. This big difference shows the fusion of modalities is much better than the fusion of recognitions. When the modalities are selected and the matchers (i.e., recognitions) are fixed (refer to Row 2 in Table 5), no big difference between different fusion processes is observed, for example, $\Delta\mu_{RRI}$(HMM, KNN | Recognition) = 0, and $\Delta\mu_{RRI}$(HMM, LDA | Recognition) = 0.0848. These comparisons show that the fusion of different modalities is a *dominant* factor, which makes the different fusion processes have less impact on fusion improvement. When the matchers are chosen and the modality is fixed (refer to Row 1 in Table 5), we have μ_{RRI}(HMM, KNN | Modality) = 0.0448, and μ_{RRI}(HMM, LDA | Modality) = 0.1518. These results show that the *fusion process* plays an important role when fusing multi-matcher scores from a single modality (i.e., without the dominant factor of modality). Note that in Table 5 the two identical entries at Row 2, Column 3 and 4 are just coincident.

Similar quantitative analyses on ASUMS-Face-Face database (Rows 3–4 in Table 5) are given as follows. $\Delta\mu_{RRI}$(Modality, Recognition | HMM) = 0.7231 − 0.6788 = 0.0443 reveals that the fusion of different modalities (thermal face and visible face) is slightly better than the fusion of recognitions but no longer a dominant factor. $\Delta\mu_{RRI}$(HMM, KNN | Recognition) = 0.1135 and μ_{RRI}(HMM, KNN | Modality) = 0.0876 indicate that the different fusion processes become a more important factor when the modality is not a dominant factor.

How to apply these findings to guide a MBSF development and application is discussed below. *Modality* is the key and dominant factor in score fusion, but adding more matcher scores to the fusion will improve the performance further. In fact, R_F(NIST-Face-Finger, HMM_Fusion) = 99.68% (4-score fusion shown in Table 2a) is higher than R_F(NIST-Face-Finger_R2, HMM_Fusion) =

TABLE 4a
The verification rates (%) of four fusion processes (R_F) on five subsets derived from the ASUMS-Face-Face database.

Database	Single Matcher (R_V)	Mean Fusion	LDA Fusion	KNN Fusion	HMM Fusion (m,n,g)
ASUMS-Face-Face_M1 (IR)	91.67, 93.75, *96.88*	96.88	98.97	97.61	98.45 (3,1,2)
ASUMS-Face-Face_M2 (DC)	90.63, 93.75, *97.92*	98.96	97.94	97.26	97.76 (3,1,2)
ASUMS-Face-Face_R1 (LDA)	91.67, 90.63	96.88	96.39	96.90	97.95 (1,2,3)
ASUMS-Face-Face_R2 (EBGM)	93.75, 93.75	97.92	98.28	99.49	99.14 (1,2,2)
ASUMS-Face-Face_R3 (FPB)	96.88, 97.92	100	98.62	98.28	98.80 (1,2,2)

TABLE 4b
The relative rate increase (RRI) of four fusion processes on five subsets derived from the ASUMS-Face-Face database.

Database	$\overline{R_V}$	Mean Fusion	LDA Fusion	KNN Fusion	HMM Fusion	(μ_{RRI},σ_{RRI})
ASUMS-Face-Face_M1 (IR)	94.10	0.4712	0.8254	0.5949	0.7373	0.6572, 0.1562
ASUMS-Face-Face_M2 (DC)	94.10	0.8237	0.6508	0.5356	0.6203	0.6576, 0.1210
ASUMS-Face-Face_R1 (LDA)	91.15	0.6475	0.5921	0.6497	0.7684	0.6644, 0.0743
ASUMS-Face-Face_R2 (EBGM)	93.75	0.6672	0.7248	0.9184	0.8624	0.7932, 0.1169
ASUMS-Face-Face_R3 (FPB)	97.40	1.0000	0.4692	0.3385	0.5385	0.5865, 0.2878
(μ_{RRI},σ_{RRI})	(NA)	0.7219, 0.1994	0.6525, 0.1345	0.6074, 0.2099	0.7054, 0.1272	(NA)

TABLE 5
The (μ_{RRI},σ_{RRI}) values of the combined modality subsets (Rows 1–2 in Table 3b and in Table 4b, respectively) and the (μ_{RRI},σ_{RRI}) values of the combined recognition subsets (the rest rows in Table 3b and in Table 4b, respectively).

Database	Mean Fusion	LDA Fusion	KNN Fusion	HMM Fusion	Fusion of What
NIST-Face-Finger_M1-M2	0.3924, 0.1525	0.6580, 0.1087	0.7650, 0.0089	0.8098, 0.0064	Recognitions/Matchers
NIST-Face-Finger_R1-R2	0.6403, 0.0032	0.8566, 0.0722	0.9414, 0.0074	0.9414, 0.0074	Modalities
ASUMS-Face-Face_M1-M2	0.6475, 0.2493	0.7381, 0.1234	0.5653, 0.0419	0.6788, 0.0827	Recognitions/Matchers
ASUMS-Face-Face_R1-R3	0.7716, 0.1981	0.5954, 0.1278	0.6355, 0.2902	0.7231, 0.1667	Modalities

99.42% (2-score fusion shown in Table 3a). The fusion process becomes very important when the score modalities are fixed, for instance, the fusion of multiple matchers of single modality. For example, imagine a human identification system of two modalities (face and finger) and of two matchers per modality that has $R_F = 99.68\%$ using HMM fusion, how can you further improve the system performance? According to the findings of this research, the recommended solution is first to add one more modality (e.g., voice or iris), then to develop a better fusion process than HMM, and/or to add more recognition methods (like Local Gabor Binary Patterns (LGBP) [30] for face recognition). Of course, using a high-performance matcher is always preferred. The implication hereby is that developing a better fusion process (e.g., better than HMM) will have a higher impact on fusion improvement (i.e., a larger RRI) than adding a third matcher into each modality.

A recent face recognition research [34] explored the performance improvement with the stereo fusion at three levels: image, feature, and score. The primary fusions investigated in that paper are *stereo fusion* with the stereo images captured from two identical cameras. Experimental results show that any level stereo fusion can improve the recognition performance. It seems that stereo image fusion and stereo feature fusion is better than stereo score fusion. However, the processes for the fusions at image level and feature level are more complicated (such as image registration). On the other hand, score fusion can be implemented without the knowledge of what images and what features, and can be performed flexibly by using variant score combinations from different cameras, modalities, and/or matchers. In addition, score fusion is faster than image fusion or feature fusion.

In the future we will sufficiently investigate and verify the current findings by developing more recognition methods and more fusion processes and by using more biometric modalities (like voice, iris, and palm geometry). A statistical analysis (e.g., ANOVA, ROCs [1]) will be conducted to study the interactions and significance of those fusion factors. We will also research the impacts of normalization procedures, decision rules, and image fusion techniques [17] on the MBSF process.

5 CONCLUSIONS

A set of experiments regarding multimodal biometric score fusion (MBSF) has been conducted in this research. A hidden Markov model (HMM) is tested for multimodal biometrics score fusion, which is the most accurate, reliable, and credible fusion process compared

to other three methods (mean, LDA, KNN). To evaluate and compare the improvement of variant fusion processes, a new metric, called *relative rate increase* (RRI), is proposed upon the concept of verification rate. The RRI metric has proved to be reasonably accurate in measuring the performance improvement resulting from MBSF. Based on the experimental results from four multimodal biometric databases, the findings can be summarized as follows. The *score modality* is the most important (key) factor in biometric score fusion which dominates the fusion result. When the number of score modalities is fixed, the fusion process becomes the next important factor to score fusion. Adding more recognition matchers has the least impact on fusion improvement. Another finding is that, different bands of face images (thermal and visible) are less diverse modalities than face and finger, which makes the score modality (of thermal faces and visible faces) no longer a dominant factor.

ACKNOWLEDGMENT

This research was supported by the Department of Defense Research and Education Program (under Grant No. W911NF-12-1-0065) administered by the Army Research Office and the Office of Naval Research.

REFERENCES

[1] S. Alsing, E. P. Blasch, and R. Bauer
Three-Dimensional Receiver Operating Characteristic (ROC) Trajectory Concepts for the Evaluation of Target Recognition Algorithms Faced with the Unknown Target Detection Problem,
Proc. SPIE. 3718, 1999.

[2] L. E. Baum, T. Petrie
Statistical Inference for Probability Functions of Finite State Markov Chains,
Ann. Math. Stat., 37:1554–1563, 1966.

[3] J. C. Bezdec
Pattern Recognition with Fuzzy Objective Function Algorithms,
Plenum Press, New York, 1981.

[4] R. Brunelli, D. Falavigna
Person Identification Using Multiple Cues,
IEEE Trans. Pattern Anal. Mach. Intell., 17(10): 955–966, 1995.

[5] H. Bunke, T. Caelli
Hidden Markov Models: Applications in Computer Vision,
World Scientific, River Edge, New Jersey, 2001.

[6] C. Burges
A Tutorial on Support Vector Machines for Pattern Recognition,
Data Mining and Knowledge Discovery, 2:121–167, 1998.

[7] H. P. Chan, K. Doi, S. Galhotra, C. J. Vyborny, H. MacMahon, P. M. Jokich
Image Feature Analysis and Computer-aided Diagnosis in Digital Radiography. I. Automated Detection of Microcalcifications in Mammography,
Med. Physics, 14:538–548, 1987.

[8] R. O. Duda, P. E. Hart
Pattern Classification and Scene Analysis,
Wiley, New York, 1973.

[9] J. Fierrez-Aguilar, J. Ortega-Garcia, J. Gonzalez-Rodriguez, J. Bigun
Discriminative Multimodal Biometric Authentication based on Quality Measures,
Pattern Recognition, 38(5):777–779, 2005.

[10] M. Figueiredo, A. K. Jain
Unsupervised Learning of Finite Mixture Models,
IEEE Trans. Pattern Anal. Mach. Intell. 24 (3), 381–396, 2002.

[11] G. D. Forney
The Viterbi Algorithm,
Proc. IEEE 61, 268–278, 1973.

[12] P. Gader, M. Mystkowski, Y. Zhao
Landmine Detection with Ground Penetrating Radar using Hidden Markov Models,
IEEE Trans. Geosci. Remote Sens., 39:1231–1244, 2001.

[13] A. Jain, K. Nandakumar, A. Ross
Score Normalization in Multimodal Biometric Systems,
Pattern Recognition, 38(12):2270–2285, 2005.

[14] R. M. Kil, I. Koo
Optimization of a Network with Gaussian Kernel Functions Based on the Estimation of Error Confidence Intervals,
Proc. of IJCNN 2001, 3:1762–176, 2001.

[15] L. I. Kuncheva
A Theoretical Study on Six Classifier Fusion Strategies,
IEEE Trans. Pattern Anal. Mach. Intell., 24(2):281–286, 2002.

[16] L. I. Kuncheva
Switching between Selection and Fusion in Combining Classifiers: an Experiment,
IEEE Trans. on Systems, Man, and Cybernetics, Part B, 32(2):146–156, 2002.

[17] Z. Liu, E. Blasch, Z. Xue, R. Langaniere, and W. Wu
Objective Assessment of Multiresolution Image Fusion Algorithms for Context Enhancement in Night Vision: A Comparative Survey,
IEEE Trans. Pattern Anal. Mach. Intell., 34(1):94–109, 2012.

[18] J. Lu, K. N. Plataniotis, A. N. Venetsanopoulos
Face Recognition Using LDA-Based Algorithms,
IEEE Trans. on Neural Networks, 14(1):195–200, 2003.

[19] M. Mohamed, P. Gader
Generalized Hidden Markov Models part 2: Applications to Handwritten Word Recognition,
IEEE Trans. Fuzzy Systems, 8:186–194, 2000.

[20] K. Murphy
Machine Learning: a Probabilistic Perspective,
MIT Press, Cambridge, Massachusetts, 2012.

[21] K. Nandakumar, Y. Chen, S. C. Dass, A. K. Jain
Likelihood ratio-based biometric score fusion,
IEEE Trans. Pattern Anal. Mach. Intell., 30(2): 342–347, 2008.

[22] National Institute of Standards and Technology
NIST Biometric Scores Set—release 1,
http://www.itl.nist.gov/iad/894.03/biometricscores, 2004.

[23] S. Prabhakar, A. K. Jain
Decision-level Fusion in Fingerprint Verification,
Pattern Recognition, 35(4):861–874, 2002.

[24] L. Rabiner
Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,
Proc. IEEE 77, 257–286, 1989.

[25] S. Raudys
Statistical and Neural Classifiers: an Integrated Approach to Design,
Springer, London, 2001.

[26] A. Ross and A. Jain
Information Fusion in Biometrics,
Pattern Recognition Letters, 24:2115–2125, 2003.

[27] K.-A. Toh, X. Jiang, and W.-Y. Yau
Exploiting Global and Local Decisions for Multimodal Biometrics Verification,
IEEE Trans. on Signal Processing, 52(10):3059–3072, 2004.

[28] B. Ulery, A. R. Hicklin, C. Watson, W. Fellner, P. Hallinan
Studies of Biometric Fusion,
NIST Interagency Report, 2006.

[29] L. Wiskott, J. M. Fellous, N. Krüger, C. von der Malsburg
Face Recognition by Elastic Bunch Graph Matching,
IEEE Trans. Pattern Anal. Mach. Intell., 19 (7):775–779, 1997.

[30] S. Xie, S. Shan, X. Chen, and J. Chen
Fusing Local Patterns of Gabor Magnitude and Phase for Face Recognition,
IEEE Trans. Image Process., 19(5):1349–1361, 2010.

[31] Y. Zheng
A Hidden Markov Model for Multimodal Biometrics Score Fusion,
Proc. SPIE 8064, 80640D, 2011.

[32] Y. Zheng
A Novel Orientation Code for Face Recognition,
Proc. SPIE 8056, 805606, 2011.

[33] Y. Zheng and E. Blasch
Score Fusion and Decision Fusion for the Performance Improvement of Face Recognition,
Int'l Conf. on Info Fusion, 2013.

[34] Y. Zheng and E. Blasch
The Advantages of Stereo Vision in a Face Recognition System,
Proc. SPIE 9091, 2014.

Yufeng Zheng received his Ph.D. degree in Optical Engineering/Image Processing from the Tianjin University (Tianjin, China) in 1997. He is presently with the Alcorn State University (Mississippi, USA) as an associate professor. Dr. Zheng serves as a program director of the Computer Networking and Information Technology Program, and a director of the Pattern Recognition and Image Analysis Lab. He is the principle investigator of three federal research grants in night vision enhancement, and in multispectral face recognition. So far Dr. Zheng holds two patents in glaucoma classification and face recognition, and has published one book, six book chapters, and more than 70 peer-reviewed papers. His research interests include pattern recognition, biologically inspired image analysis, biometrics, information fusion, and computer-aided diagnosis. Dr. Zheng is a Cisco Certified Network Professional (CCNP), a senior member of SPIE, a member of IEEE, Computer Society & Signal Processing, as well as a technical reviewer.

Erik Blasch received his B.S. in mechanical engineering from the Massachusetts Institute of Technology in 1992 and M.S. degrees in mechanical engineering ('94), health science ('95), and industrial engineering (human factors) ('95) from Georgia Tech and attended the University of Wisconsin for a M.D./Ph.D. in mechanical engineering/neurosciences until being called to active duty in 1996 to the United States Air Force. He completed an M.B.A. ('98), M.S.E.E. ('98), M.S. econ ('99), M.S./Ph.D. psychology (ABD), and a Ph.D. in electrical engineering from Wright State University and is a graduate of Air War College. From 2000–2010, Dr. Blasch was the information fusion evaluation tech lead for the Air Force Research Laboratory (AFRL) Sensors Directorate—COMprehensive Performance Assessment of Sensor Exploitation (COMPASE) Center, adjunct professor with Wright State University, and a reserve officer with Air Office of Scientific Research. From 2010–2012. Dr. Blasch was an exchange scientist to Defence R&D Canada at Valcartier, Quebec in the Future Command and Control (C2) Concepts group. He is currently with the AFRL Information Directorate. He compiled over 30 top ten finishes as part of robotic teams in international contests, received the 2009 IEEE Russ Bioengineering Award, and the 2014 *Joseph Mignogna Data Fusion Award* from the U.S. Department of Defense Joint Directors of Laboratories Data Fusion Group. He is a past President of the International Society of Information Fusion (ISIF), a member of the IEEE Aerospace and Electronics Systems Society (AESS) Board of Governors, AIAA Associate Fellow, and a SPIE Fellow. His research interests include target tracking, information/sensor/image fusion, pattern recognition, and biologically-inspired applications.

INTERNATIONAL SOCIETY OF INFORMATION FUSION

ISIF Website: http://www.isif.org

2014 BOARD OF DIRECTORS*

2012–2014	2013–2015	2014–2016
Darin T. Dunham	Jean Dezert	Sten F. Andler
Fredrik Gustafsson	Gee-Wah Ng	Murat Efe
Lance M. Kaplan	Anne-Laure Jousselme	Lyudmila Mihaylova

*Board of Directors are elected by the members of ISIF for a three year term.

PAST PRESIDENTS

Wolfgang Koch, 2013	Erik Blasch, 2007	Yaakov Bar-Shalom, 2002
Roy Streit, 2012	Pierre Valin, 2006	Pramod Varshney, 2001
Joachim Biermann, 2011	W. Dale Blair, 2005	Yaakov Bar-Shalom, 2000
Stefano Coraluppi, 2010	Chee Chong, 2004	Jim Llinas, 1999
Elisa Shahbazian, 2009	Xiao-Rong Li, 2003	Jim Llinas, 1998
Darko Musicki, 2008		

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.

Journal of Advances in Information Fusion Volume 9 Number 2 December 2014 ISSN 1557-6418