June 2016 Volume 11 Number 1 ISSN 1557-6418

JAIF 16

Journal of Advances in Information Fusion

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

Regular Papers Page

Partitioned Update Kalman Filter 3
Matti Raitoharju, Tampere University of Technology, Finland
Robert Piché, Tampere University of Technology, Finland
Juha Ala-Luhtala, Tampere University of Technology, Finland
Simo Ali-Löytty, Tampere University of Technology, Finland

Sigma-Point Filtering and Smoothing Based Parameter Estimation in Nonlinear Dynamic Systems 15
Juho Kokkala, Aalto University, Finland
Arno Solin, Aalto University, Finland
Simo Särkkä, Aalto University, Finland

On the relation between Gaussian process quadratures and sigma-point methods 31
Simo Särkkä, Aalto University, Finland
Jouni Hartikainen, Rocsole Ltd., Finland
Lennart Svensson, Chalmers University of Technology, Sweden
Fredrik Sandblom, Volvo Group, Sweden

Nonlinear Kalman Filters Explained: A Tutorial on Moment Computations and Sigma Point Methods 47
Michael Roth, Linköping University, Sweden
Gustaf Hendeby, Linköping University, Sweden
Fredrik Gustafsson, Linköping University, Sweden

The Smart Sampling Kalman Filter with Symmetric Samples 71
Jannik Steinbring, Karlsruhe Institute of Technology (KIT), Germany
Martin Pander, Karlsruhe Institute of Technology (KIT), Germany
Uwe D. Hanebeck, Karlsruhe Institute of Technology (KIT), Germany

Sigma-Point Set Rotation for Derivative-Free Filters in Target Tracking Applications 91
Jindrich Duník, University of West Bohemia, Czech Republic
Ondrej Straka, University of West Bohemia, Czech Republic
Miroslav Šimandl, University of West Bohemia, Czech Republic
Erik Blasch, Air Force Research Lab, USA

Volumes 1–10 Index 111

From the Editor-In-Chief

Guest Editorial: Nonlinear Derivative-Free Filters: Theory and Applications

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:

June 2016

Jindrich Duník

Ondrej Straka

Guest Editorial: Foreword to the Special Issue on Nonlinear Derivative-Free Filters: Theory and Applications

State estimation, or filtering, is a focal point for navigation, positioning, and tracking systems. State estimation also plays a crucial role in various areas and applications, where knowledge of the state is required for a (multistep) prediction, control, fault detection, or generally for decision making.

The special issue is focused on the design of local, i.e., nonlinear Kalman-filter-based, derivative-free filters for nonlinear stochastic dynamic time-varying systems with an emphasis on the theoretical advances in the field of study. In particular, the focus of the issue is laid on both the design of novel algorithms for state and parameter estimation and the in-depth analysis and further development of existing methods.

The special issue consists of six papers all from renowned research groups systematically working in the area of the state estimation. Some of the papers are significantly extended versions of the papers presented at the 17th International Conference on Information Fusion (FUSION 2014) within the special session "Advanced Sigma-Point Filters: Analysis, Sigma-Point Set Design, and Applications".

The opening paper of this special issue, the paper entitled "Partitioned Update Kalman Filter" by M. Raitoharju, R. Piché, J. Ala-Luhtala, and S. Ali-Löytty, is devoted to the design of the state estimation algorithm inherently measuring or assessing the nonlinearity of the function in the measurement equation of the state-space model at the actual estimator working point. Depending on the computed measure of nonlinearity, the particular elements of the measurement vector are processed starting from those having the lowest measure, i.e., from those with the lowest expected approximation error.

The second paper entitled "Sigma-Point Filtering based Parameter Estimation in Nonlinear Dynamic Systems" by J. Kokkala, A. Solin, and S. Särkkä, deals with the estimation of unknown parameters in nonlinear state-space models based on the maximum marginal

likelihood. The marginalized likelihood is computed using two approximate methods, namely a direct likelihood method and the expectation-maximisation algorithm, evaluated by means of approximate nonlinear filtering and smoothing techniques. The considered techniques take advantage of a higher-order unscented transformation and the Gauss-Hermite integration rule.

The third paper entitled "On the Relation between Gaussian Process Quadratures and Sigma-point Methods" by S. Särkkä, J. Hartikainen, L. Svensson, and F. Sandblom, is dedicated to a numerical evaluation of the Gaussian weighted integrals based on Gaussian process regression methods, i.e., Gaussian process quadratures. Special attention is paid to the interpretation of derivative-free sigma-point approximate methods for the evaluation of integrals typically appearing in the local filter design (for conditional moment computation) as a special case of Gaussian process quadratures. As a consequence, a set of novel local methods for nonlinear state estimation is proposed.

The fourth paper entitled "Nonlinear Kalman Filters Explained: A Tutorial on Moment Computations and Sigma Point Methods" by M. Roth, G. Hendeby, and F. Gustafsson, is concerned with an in-depth and widespread survey of nonlinear local estimation algorithms published so far. The algorithms are introduced in a unified framework and particular approximation techniques, both derivative-free and with an explicit derivative, used for integral evaluation are analysed with respect to the structure and properties of the considered integrals and nonlinear functions.

The fifth paper entitled "The Smart Sampling Kalman Filter with Symmetric Samples" by J. Steinbring, M. Pander, and U. D. Hanebeck, is aimed at a design of a novel local filtering algorithm suitable even for high-dimensional tasks. The algorithm is based on the deterministic symmetric equally-weighted sample set, which is used by an integration rule computing the conditional moments. The set is computed prior to the estimation experiment by an optimization procedure, thus not affecting the computational complexity of the filter.

The last, the sixth, paper entitled "Sigma-Point Set Rotation for Derivative-Free Filters in Target Tracking Applications" by J. Duník, O. Straka, M. Šimandl, and E. Blasch, is focused on a thorough analysis of the impact of user-defined parameters on the estimation performance of the derivative-free local filters. In particular, the sigma-point (or sample) set rotation determined either by a selected covariance matrix factorisation technique or by an additional sigma-point set rotation is treated. Recommendations for optimal and sub-optimal set rotations are provided and justified.

Overall, the special issue covers numerous perspectives of local nonlinear state estimation and represents an excellent overview of the state-of-the-art and current trends of the research in the area. We hope that you will find the special issue attractive, educational, and motivating and enjoy reading the papers. Finally, we would like to thank the Journal of Advances in Information Fusion (JAIF) Editorial Board for the possibility to prepare this special issue and the Editor-in-Chief, Prof. Uwe D. Hanebeck, for his encouragement and continuous support. We also greatly appreciate the authors and the reviewers for their outstanding effort and contribution to the special issue.

Jindrich Duník, Ondrej Straka
Guest Associate Editors

Department of Cybernetics
Faculty of Applied Sciences
University of West Bohemia
Pilsen, Czech Republic

Partitioned Update Kalman Filter

MATTI RAITOHARJU
ROBERT PICHÉ
JUHA ALA-LUHTALA
SIMO ALI-LÖYTTY

In this paper we present a new Kalman filter extension for state update called Partitioned Update Kalman Filter (PUKF). PUKF updates the state using multidimensional measurements in parts. PUKF evaluates the nonlinearity of the measurement function within a Gaussian prior by comparing the effect of the 2nd order term on the Gaussian measurement noise. A linear transformation is applied to measurements to minimize the nonlinearity of a part of the measurement. The measurement update is then applied using only the part of the measurement that has low nonlinearity and the process is then repeated for the updated state using the remaining part of the transformed measurement until the whole measurement has been used. PUKF does the linearizations numerically and no analytical differentiation is required. Results show that when the measurement geometry allows effective partitioning, the proposed algorithm improves estimation accuracy and produces accurate covariance estimates.

Manuscript received March 9, 2015; released for publication November 25, 2015.

Refereeing of this contribution was handled by Ondrej Straka.

Authors' addresses: M. Raitoharju and R. Piché, Department of Automation Science and Engineering, Tampere University of Technology (e-mail: {matti.raitoharju, robert.piche}@tut.fi). J. Ala-Luhtala and S. Ali-Löytty, Department of Mathematics, Tampere University of Technology (e-mail: {juha.ala-luhtala, simo.ali-loytty}@tut.fi).

The authors declare that they have no competing interests. Juha Ala-Luhtala received financial support from the Tampere University of Technology Doctoral Programme in Engineering and Natural Sciences. The simulations were carried out using the computing resources of CSC–IT Center for Science.

I. INTRODUCTION

Bayesian filtering algorithms are used to compute the estimate of an n-dimensional state x. In a general discrete-time model the state evolves according to a state transition equation

$$x_t = \tilde{f}_t(x_{t-1}, \varepsilon_t^x), \tag{1}$$

where $\tilde{f}_t$ is the state transition function at time index t and ε_t^x is the state transition noise. The state estimate is updated using measurements that are modeled as

$$y_t = \tilde{h}_t(x_t, \varepsilon_t^y), \tag{2}$$

where $\tilde{h}_t$ is a measurement function and ε_t^y is the measurement noise. If the measurement and state transition are linear, noises are additive, white and normal distributed, and the prior state (x_0) is normal distributed, the Kalman update can be used to compute the posterior. If these requirements are not fulfilled, usually an approximate estimation method has to be used. In this work, we concentrate on situations where the noises are additive and Gaussian so that (1–2) take the form

$$x_t = f_t(x_{t-1}) + \varepsilon_t^x \tag{3}$$

$$y_t = h_t(x_t) + \varepsilon_t^y, \tag{4}$$

where $\varepsilon_t^x \sim \mathrm{N}(0, W_t)$, W_t is the state transition noise covariance, $\varepsilon_t^y \sim \mathrm{N}(0, R_t)$, and R_t is the measurement noise covariance.

There are two main approaches for computing an approximation of the posterior distribution:

1) Approximate probabilities using point masses (e.g. grid and particle filters)
2) Approximate probabilities by Gaussians (e.g. Kalman filter extensions)

In the first approach one problem is how to choose a good number of point masses. The first approach also often requires more computational resources than the second approach. A drawback of the second approach is that the state distribution is assumed normal and unimodal, which makes the estimate inaccurate when the true posterior is not normal. Gaussian Mixture Filters (GMFs) (a.k.a. Gaussian sum filters) can be considered as a hybrid approach that use multiple normal distributions to estimate the probability distributions and can approximate any probability density function (pdf) [1]. GMFs have the same kind of problems as the algorithms using point masses in choosing a good number of components. The algorithm that will be proposed in this paper uses the second approach and so we will concentrate on it.

The algorithms that are based on Gaussian approximations usually extend the Kalman filter update to nonlinear measurements (there are also other options, see for example [2], [3]). The Extended Kalman Filter (EKF) is a commonly used algorithm for estimation with nonlinear measurement models [4]. EKF is

based on the first order Taylor linearization of the measurement function at the mean of prior. In the Second Order Extended Kalman Filter (EKF2) the linearization takes also the second order expansion terms into account [4]. In contrast to EKF, in EKF2 the prior covariance also affects the linearization. Both EKF and EKF2 require analytical computation of the Jacobian matrix and EKF2 requires also the computation of Hessian matrices of the measurement function. In [5] a 2nd order Central Difference Filter (CDF), which can be interpreted as a derivative-free numerical approximation of EKF2, was presented. The most commonly used Kalman filter extension that does not require analytical differentiation is probably the Unscented Kalman Filter (UKF) [6]. The Gaussian approximations in UKF are based on the propagation of "sigma points" through the nonlinear functions. Cubature Kalman Filters (CKFs) are similar algorithms, but they have different theory in the background [7]. All these methods do the update as a single operation.

Some algorithms do multiple linearizations to improve the estimate. In [8] the posterior is computed using multiple EKF updates that use different linearization points. In the Iterated Extended Kalman Filter (IEKF), the EKF update is computed in the prior mean and then the new mean is used as the new linearization point [9]. This can be done several times. A similar update can be done also with other Kalman type filters [10]. The Recursive Update Filter (RUF) updates the prior with measurement with reduced weight several times [11]. In every update the linearization point is used from the posterior of the last reduced weight update. GMFs can also be considered to be filters that do the linearization multiple times, once for each Gaussian component, and any Kalman filter extension can be used for the update.

In this paper we present Partitioned Update Kalman Filter (PUKF) that updates the state also in several steps. PUKF first computes the nonlinearity of measurement models. The nonlinearity measure is based on comparing the covariance of the 2nd order term covariance of the Gaussian measurement noise. Computation of this nonlinearity measure requires the same matrices as the EKF2 update and for this we use the 2nd order CDF [5], which is a derivative free version of the EKF2.

PUKF applies a linear transformation to the measurement function to make a new measurement function that has linearly independent measurement noise for measurement elements; the smallest nonlinearity corresponding to a measurement element is minimized first, then the second smallest nonlinearity etc. After the transformation, the update is done using only measurement elements that have smaller nonlinearity than a set threshold value or using the measurement element with the smallest nonlinearity. After the partial measurement update the covariance has become smaller or remained the same and the linearization errors for remaining measurements may have also became smaller. The remaining measurements' nonlinearity is re-evaluated using the partially updated state, the remaining measurements are transformed and a new partial update is applied until the whole measurement is applied. This process is shown in Figure 1. The use of only some dimensions of the measurements to get a new prior and the optimization of measurement nonlinearities differentiates PUKF from other Kalman filter extensions.

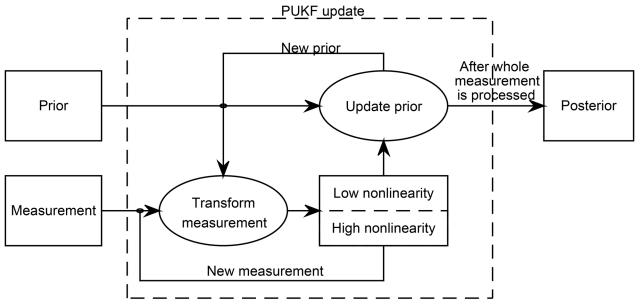


Fig. 1. Process diagram of the PUKF

The article is structured as follows: In Section II a numerical method for approximate EKF2 update is presented. The main algorithm is presented in Section III. The accuracy and reliability of the proposed algorithm is compared with other Kalman filter extensions and PFs in Section IV. Section V concludes the article.

II. EKF2 AND ITS NUMERICAL UPDATE USING 2ND ORDER CDF

Kalman filter extensions, like all Bayesian filters, can be computed in two stages: prediction and update. For the state transition model (3) the state is propagated in EKF2 using equations [9]:

$$\mu_t^- = f_t(\mu_{t-1}^+) + \tfrac{1}{2}\xi_t^f \tag{5}$$

$$P_t^- = J^f P_{t-1}^+ J^{f^T} + \tfrac{1}{2}\Xi_t^f + W_t, \tag{6}$$

where μ_t^- is the predicted mean at time t, μ_{t-1}^+ is the posterior mean of the previous time step, J^f is the Jacobian of the state transition function evaluated at μ_{t-1}^+, P_t^- is the predicted covariance, P_{t-1}^+ is the posterior covariance of the previous time step and ξ_t^h and Ξ_t^h are defined as

$$\xi_{t[i]}^f = \operatorname{tr} P_{t-1}^+ H_i^f \tag{7}$$

$$\Xi_{t[i,j]}^f = \operatorname{tr} P_{t-1}^+ H_i^f P_{t-1}^+ H_j^f, \tag{8}$$

where H_i^f is the Hessian of the ith element of the state transition function evaluated at μ_{t-1}^+. To simplify the notation we do not further show the time indices.

The update equations of EKF2 for the measurement model (4) are [9]

$$y^- = h(\mu^-) + \tfrac{1}{2}\xi^h \tag{9}$$

$$S = J^h P^- J^{h^T} + \tfrac{1}{2}\Xi^h + R \tag{10}$$

$$K = P^- J^{h^T} S^{-1} \tag{11}$$

$$\mu^+ = \mu^- + K(y - y^-) \tag{12}$$

$$P^+ = P^- - KSK^T, \tag{13}$$

where J^h is the Jacobian of the measurement function, K is the Kalman gain, S is the innovation covariance, and ξ^h and Ξ^h are defined as

$$\xi^h_{[i]} = \operatorname{tr} P^- H^h_i \tag{14}$$

$$\Xi^h_{[i,j]} = \operatorname{tr} P^- H^h_i P^- H^h_j, \tag{15}$$

where H^h_i is the Hessian matrix of the ith component of the measurement function. Eqns (9–13) can be turned into the EKF update using $\xi^h = 0$ and $\Xi^h = 0$.

If the measurement model is linear, the trace terms in EKF2 are zero and the update is the optimal update of the Kalman filter. When the measurement function is a second order polynomial the EKF2 update is not optimal as the distributions are no longer Gaussian, but the mean (9) and innovation covariance (10) are correct.

In this paper we use a numerical algorithm to compute an EKF2 like update. To derive this algorithm, we start with the formulas of the 2nd-order CDF from [5]. Let $\sqrt{P^-}$ be a matrix such that

$$\sqrt{P^-}\sqrt{P^-}^T = P^-. \tag{16}$$

In our implementation this matrix square root is computed using Cholesky decomposition.

Next we define matrices M and Q that are used for computing the numerical EKF2 update. We use notation $\Delta_i = \gamma\sqrt{P^-}_{[:,i]}$, where $\sqrt{P^-}_{[:,i]}$ is the ith column of matrix $\sqrt{P^-}$ and γ is an algorithm parameter that defines the spread of the function evaluations. Matrix M, whose elements are

$$\begin{aligned} M_{[:,i]} &= \left[J^h\sqrt{P^-}\right]_{[:,i]} \\ &\approx \gamma^{-1}\frac{h(\mu^- + \Delta_i) - h(\mu^- - \Delta_i)}{2}, \end{aligned} \tag{17}$$

is needed for the terms with Jacobian. The matrices $Q_k \approx \sqrt{P^-}H^{h_k}\sqrt{P^-}^T$ are needed to compute terms with Hessians. Elements of Q_k are

$$\begin{aligned} Q_{k_{[i,i]}} &= \gamma^{-2}[h_{[k]}(\mu^- + \Delta_i) + h_{[k]}(\mu^- - \Delta_i) - 2h_{[k]}(\mu^-)] \\ Q_{k_{[i,j]}} &= \gamma^{-2}[h_{[k]}(\mu^- + \Delta_i + \Delta_j) - h_{[k]}(\mu^- + \Delta_i) \\ &\quad - h_{[k]}(\mu^- + \Delta_j) + h_{[k]}(\mu^-)], \quad i \neq j. \end{aligned} \tag{18}$$

The EKF2 update can be approximated with these by doing the following substitutions:

$$\xi^h_i = \operatorname{tr} P^- H^h_i \approx \operatorname{tr} Q_i \qquad \text{in (9)} \tag{19}$$

$$J^h P^- J^{h^T} \approx MM^T \qquad \text{in (10)} \tag{20}$$

$$P^- J^{h^T} \approx \sqrt{P^-}M^T \qquad \text{in (14)} \tag{21}$$

$$\Xi^h_{[i,j]} = \operatorname{tr} P^- H^h_i P^- H^h_j \approx \operatorname{tr} Q_i Q_j. \qquad \text{in (15).} \tag{22}$$

The prediction step can be approximated by computing M^f (17) and Q^f (18) matrices using the state transition function instead of the measurement function and doing the following substitutions:

$$J^h P^+_{t-1} J^{h^T} \approx M^f M^{f^T} \qquad \text{in (6)} \tag{23}$$

$$\operatorname{tr} P^- H^f_i \approx \operatorname{tr} Q^f_i \qquad \text{in (7)} \tag{24}$$

$$\operatorname{tr} P^- H^f_i P^- H^f_j \approx \operatorname{tr} Q^f_i Q^f_j \qquad \text{in (8).} \tag{25}$$

In [12], an update algorithm similar to numerical EKF2 is proposed that uses only the diagonal elements of Q matrices. They state that $\gamma = \sqrt{3}$ for Gaussian distributions is optimal because it preserves the fourth moment and so we use this γ value in our algorithm.

III. PARTITIONED UPDATE KALMAN FILTER

When the measurement function is linear and the measurement noise covariance is block diagonal, the Kalman update produces identical results whether measurements are applied one block at a time or all at once. In our approach we try to find as linear as possible part of the measurement and use this part to update the state estimate to reduce approximation errors in the remaining measurement updates. When the measurement noise covariance R is not diagonal a linear transformation (decorrelation) is applied to transform the measurement so that the transformed measurement has diagonal covariance [13]. In PUKF, we choose this decorrelation so that the nonlinearity of the least nonlinear measurement element is minimized. The prior is updated using the least nonlinear part of the decorrelated measurements. After the partial update the process is repeated for the remaining dimensions of the transformed measurement.

For measuring the amount of nonlinearity we compare the trace term Ξ^h with the covariance of the measurement noise:

$$\begin{aligned} \eta &= \operatorname{tr}\sum_{i=1}^{d}\sum_{k=1}^{d} R^{-1}_{[k,i]} P^- H^h_i P^- H^h_k \\ &= \operatorname{tr}\sum_{i=1}^{d}\sum_{k=1}^{d} R^{-1}_{[k,i]}\Xi^h_{[i,k]} \end{aligned} \tag{26}$$

This nonlinearity measure is a local approximation of the nonlinearity and is developed from the measure presented in [9], [14]. In [15] it was compared with other nonlinearity measures and it was shown to be a good indication of how accurately state can be updated with a nonlinear measurement model using a Kalman filter extension. When the measurement model is linear the nonlinearity measure is $\eta = 0$.

The matrix Ξ^h depends on the nonlinearity of the measurement function and contributes to the innovation covariance (10) similarly, but multiplied with $\frac{1}{2}$, as the Gaussian measurement noise R. The measure (26) compares the ratio of Gaussian covariance R and non-Gaussian covariance Ξ^h. Figure 2 shows how the pdf

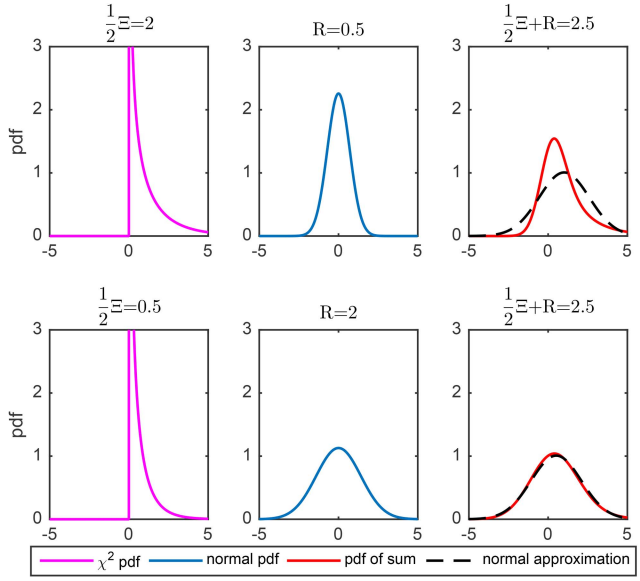


Fig. 2. Probability density functions of sums of independent χ^2 and normal random variables with different variances

of the sum of independent normal and χ^2 distributed random variables is closer to normal when $R > \Xi$ than when $R < \Xi$. The χ^2 distribution is chosen in the example, because a normal distributed variable squared is χ^2 distributed and in the second order polynomial approximations the squared term is the nonlinear part.

The nonlinearity measure (26) can be approximated numerically using the substitution (22). Numerical computation of a similar nonlinearity measure was proposed in [16], but the algorithm presented in Section II does the nonlinearity computation with fewer measurement function evaluations.

Multiplying (4) by an invertible square matrix D gives a transformed measurement model

$$Dy = Dh(x) + D\varepsilon^y. \tag{27}$$

We use the following notations for the transformed measurement model: $\hat{y} = Dy$, $\hat{h}(x) = Dh(x)$, $\hat{R} = DRD^T$, and $\hat{\varepsilon}^y = D\varepsilon^y \sim \mathrm{N}(0, \hat{R})$. We will show that D can be chosen so that

$$\hat{R} = I \quad \text{and} \quad \operatorname{tr} P^- \hat{H}_i^h P^- \hat{H}_k^h = 0, \quad i \neq k, \tag{28}$$

where $\hat{H}_i^h$ and $\hat{H}_k^h$ denote the Hessians of the ith and kth element of $\hat{h}(x)$.

In [15], it was shown that when a measurement model is transformed so that $\hat{R} = I$ the nonlinearity measure (26) is equal to the nonlinearity measure of the transformed measurements

$$\begin{aligned}\eta &= \operatorname{tr} \sum_{i=1}^{d} \sum_{k=1}^{d} R_{[k,i]}^{-1} P^- H_i^h P^- H_k^h \\ &= \operatorname{tr} \sum_{i=1}^{d} \sum_{k=1}^{d} \hat{R}_{[k,i]}^{-1} P^- \hat{H}_i^h P^- \hat{H}_k^h = \operatorname{tr} \sum_{i=1}^{d} P^- \hat{H}_i^h P^- \hat{H}_i^h.\end{aligned} \tag{29}$$

Because the cross terms do not affect to the amount of nonlinearity we can extract the nonlinearity caused by individual elements of the transformed measurements

$$\eta_i = \operatorname{tr} P^- \hat{H}_i^h P^- \hat{H}_i^h \tag{30}$$

and the total nonlinearity is

$$\eta = \sum_{i=1}^{d} \eta_i. \tag{31}$$

In Appendix A it is shown that

$$\hat{\Xi}_{[i,j]}^h = [D\Xi^h D^T]_{[i,j]} \approx \operatorname{tr} P^- \hat{H}_i^h P^- \hat{H}_j^h. \tag{32}$$

In this case the measurement related error terms of the transformed measurement $\hat{R}$ and $\hat{\Xi}^h$ are diagonal. This makes the measurements independent and allows the update of the state one element at a time.

In PUKF nonlinearities are minimized in such a way that η_1 (30) is as small as possible. Then η_2 is minimized such that η_1 does not change, and η_3 so that η_1 and η_2 do not change etc. The decorrelation transformation D that does the desired nonlinearity minimization can be computed by first computing a matrix square root (16) of the measurement noise covariance

$$\sqrt{R}\sqrt{R}^T = R \tag{33}$$

and then an eigendecomposition of $\sqrt{R}^{-1} \Xi^h \sqrt{R}^{-T}$

$$U \Lambda U^T = \sqrt{R}^{-1} \Xi^h \sqrt{R}^{-T}. \tag{34}$$

We assume that the eigenvalues in the diagonal matrix Λ are sorted in ascending order. The transformation matrix is now

$$D = U^T \sqrt{R}^{-1}. \tag{35}$$

A proof that this transformation minimizes the nonlinearity measures is given in Appendix B. After transforming the measurement model with this matrix, the measurement noise covariance is $\hat{R} = I$ and $\hat{\Xi}^h = \Lambda$.

After the measurement model is decorrelated (multiplied with D), the parts of measurement model that have low nonlinearity ($\Lambda_{[i,i]} \leq \eta_{\text{threshold}}$) are used in the update (Section II). If there is no such part then the most linear element of the measurement model is used to update the state. Then the same process is repeated for the remaining transformed measurement model until the whole measurement is processed.

In summary the PUKF update is:

1) Transform the measurement model using D (35)
2) Update the prior using only the least nonlinear measurement elements of the transformed measurement
3) If there are measurement elements left, use them as new measurement and use the updated state as a new prior and return to step 1

The detailed PUKF algorithm is presented in Algorithm 1 and a Matlab implementation is available online [17].

ALGORITHM 1 *Algorithm for doing the measurement update in PUKF*

input : Prior state: μ–mean P–covariance
Measurement model: y–value, $h(\cdot)$–function, R–covariance
$\eta_{\text{threshold}}$–nonlinearity limit,
γ–measurement function evaluation spread (default $\gamma = \sqrt{3}$)
output: Updated state: μ–mean, P–covariance

Compute $\sqrt{R}$ (33)
$d \leftarrow$ measurement dimension
while $d > 0$ **do**
  Compute $\sqrt{P}$ (16)
  Compute M and Q_i, $1 \le i \le d$ (17–18)
  Compute ξ^h and Ξ^h (19) and (22)
  Compute U and Λ (34)
  $D \leftarrow U^T \sqrt{R}^{-1}$
  Choose largest k so that $\Lambda_{[i,i]} \le \eta_{\text{threshold}}$, $i \le k \wedge \Lambda_{[j,j]} > \eta_{\text{threshold}}$, $j > k$
  if $k == 0$ **then**
    $k \leftarrow 1$
  end
  // Compute partial EKF2 update
    $y^- \leftarrow D_{[1:k,:]}[h(\mu) + \frac{1}{2}\xi^h]$
    $S \leftarrow D_{[1:k,:]} M M^T D^T_{[1:k,:]} + \frac{1}{2}\Lambda_{[1:k,1:k]} + I$
    $K \leftarrow \sqrt{P} M^T D^T_{[1:k,:]} S^{-1}$
    $\mu \leftarrow \mu + K(D_{[1:k,:]} y - y^-)$
    $P \leftarrow P - KSK^T$
  // Update remaining measurement
    $y \leftarrow D_{[k+1:d,:]} y$
    $h(x) \leftarrow D_{[k+1:d,:]} h(x)$
    $\sqrt{R} \leftarrow I$ Updated measurement noise covariance is an identity matrix due to decorrelation
    $d \leftarrow d - k$ Updated measurement dimension
end

The amount of nonlinearity (26) for independent measurements is equal to the sum of the nonlinearities for each of the measurements. The quantity $\eta_{\text{threshold}}$ is compared separately to independent transformed measurements elements and, thus, we propose to use same $\eta_{\text{threshold}}$ regardless of the measurement dimension. As a rule of thumb the nonlinearity threshold can be set to $\eta_{\text{threshold}} = 1$, which is equal to the threshold proposed for one dimensional measurements in [9].

Figure 3 shows how PUKF treats a two-dimensional second order polynomial measurement function

$$y = \begin{bmatrix} x^2 - 2x - 4 \\ -x^2 + \frac{3}{2} \end{bmatrix} + \varepsilon, \tag{36}$$

where $\varepsilon \sim \mathrm{N}(0, I)$. The prior has mean 1 and covariance 1. The nonlinearity of each measurement is 4 and the total nonlinearity is 8. Then $D = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ and the

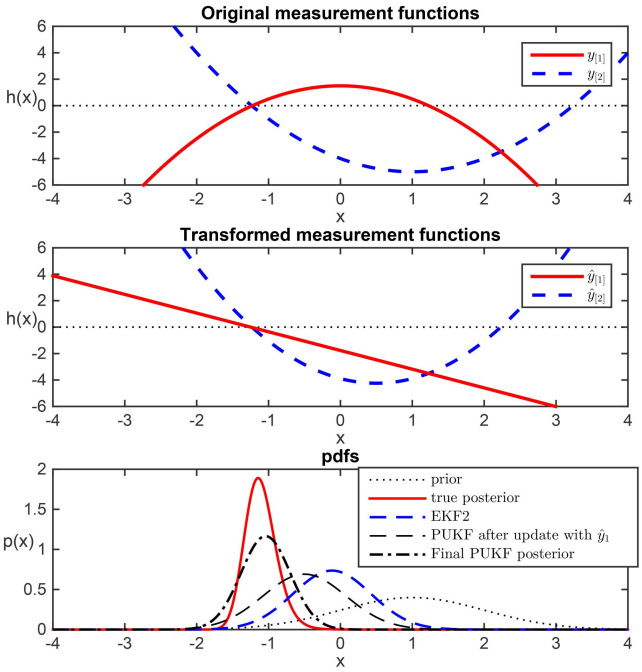


Fig. 3. Transforming second order polynomial measurements to minimize nonlinearity of $\hat{y}_1$ and posterior comparison of PUKF and EKF2

transformed measurement model has a linear term and a polynomial term

$$\hat{y} = \sqrt{2} \begin{bmatrix} -x - \frac{5}{4} \\ x^2 - x - \frac{11}{4} \end{bmatrix} + \hat{\varepsilon}, \tag{37}$$

where $\hat{\varepsilon} \sim \mathrm{N}(0, I)$. After transformation the first element of the measurement function is linear and $\eta_1 = 0$ and all the nonlinearity is associated with the second element $\eta_2 = 8$. In PUKF the linear measurement function is applied first and the partially updated state has mean $-\frac{1}{2}$ and covariance $\frac{1}{3}$. The polynomial measurement function is applied using this partially updated state. The amount of nonlinearity for the second order polynomial has decreased from 8 to $\frac{8}{9}$. EKF2 applies both measurements at once and the posterior estimate is the same for the original and transformed measurement models as shown in Appendix A. When comparing to the true posterior, which is computed using a dense grid, the posterior estimate of PUKF is significantly more accurate than the EKF2 posterior estimate.

IV. TESTS

We compare the proposed PUKF with other Kalman filter extensions and a PF in three different test scenarios. The PUKF was tested with 4 different values for $\eta_{\text{threshold}}$. When $\eta_{\text{threshold}} = \infty$ the whole measurement is applied at once and the algorithm is a numerical EKF2. When $\eta_{\text{threshold}} < 0$ measurement elements are processed one at a time and when $\eta_{\text{threshold}} = 0$ all linear measurement elements are first processed together and then nonlinear measurement elements one by one. Due to numerical roundoff errors it is better to use a small positive

$\eta_{\text{threshold}}$ to achieve this kind of behaviour. In our tests we use values $\{-\infty, 0.1, 1, \infty\}$ for $\eta_{\text{threshold}}$.

EKF and EKF2 are implemented as explained in Section II with analytical Jacobians and Hessians. RUF is implemented according to [11] with 3 and 10 steps. IEKF uses 10 iterations. For UKF the values for sigma point parameters are $\alpha = 10^{-3}$, $\kappa = 0$, $\beta = 2$. All Kalman filter extensions are programmed in Matlab with similar levels of code optimizations, but the runtimes should still be considered to be only indicative.

For reference we computed estimates with a bootstrap particle filter that does systematic resampling at every time step [18] using various numbers of particles and with a PF that uses EKF for computing the proposal distribution [19] with 10 particles.

In every test scenario the state transition model is linear time-invariant $x_t = J^f x_{t-1} + \varepsilon^x$, where $\varepsilon^x \sim \mathrm{N}(0, W)$. Thus, the prediction step (5)–(6) can be computed analytically and all Kalman filter extensions in tests use the analytical prediction.

The first test scenario is an aritificial example chosen to show the maximal potential of PUKF. The measurement model used is

$$h(x) = \begin{bmatrix} 2x_{[1]} + x_{[2]} + x_{[3]} + \frac{1}{2}x_{[1]}^2 + \frac{1}{2}x_{[2]}^2 + \frac{1}{2}x_{[3]}^2 \\ x_{[1]} + 2x_{[2]} + x_{[3]} + \frac{1}{2}x_{[1]}^2 + \frac{1}{2}x_{[2]}^2 + \frac{1}{2}x_{[3]}^2 \\ x_{[1]} + x_{[2]} + 2x_{[3]} + \frac{1}{2}x_{[1]}^2 + \frac{1}{2}x_{[2]}^2 + \frac{1}{2}x_{[3]}^2 \\ x_{[1]} + x_{[2]} + x_{[3]} + x_{[1]}^2 + \frac{1}{2}x_{[2]}^2 + \frac{1}{2}x_{[3]}^2 \\ x_{[1]} + x_{[2]} + x_{[3]} + \frac{1}{2}x_{[1]}^2 + x_{[2]}^2 + \frac{1}{2}x_{[3]}^2 \\ x_{[1]} + x_{[2]} + x_{[3]} + \frac{1}{2}x_{[1]}^2 + \frac{1}{2}x_{[2]}^2 + x_{[3]}^2 \end{bmatrix} + \varepsilon^y, \tag{38}$$

where $\varepsilon^y \sim \mathrm{N}(0, 8I + \mathbf{1})$ and $\mathbf{1}$ is a matrix of ones. This model is a linear transformation of

$$\hat{h}(x) = \begin{bmatrix} x_{[1]} \\ x_{[2]} \\ x_{[3]} \\ \frac{1}{2}x_{[1]}^2 \\ \frac{1}{2}x_{[2]}^2 \\ \frac{1}{2}x_{[3]}^2 \end{bmatrix} + \hat{\varepsilon}^y, \tag{39}$$

where $\hat{\varepsilon}^y \sim \mathrm{N}(0, I)$. The first three elements of (39) are linear and PUKF with $\eta_{\text{threshold}} \in \{0.1, 1\}$ uses the three linear measurement functions first to update the state. In this test scenario the prior mean is at the origin, the prior and state transition noise covariances are both $16I$, and the state transition matrix is an identity matrix.

Results for positioning with measurement model (38) are presented in Figure 4. The markers in the upper plot show the 5%, 25%, 50%, 75% and 95% quantiles of mean errors for each method. The quantiles are computed from 1000 runs consisting of 10 steps each. To show the quantiles better a logarithmic scale for error is used. PUKF ($\eta_{\text{threshold}} < \infty$) is the most accurate of

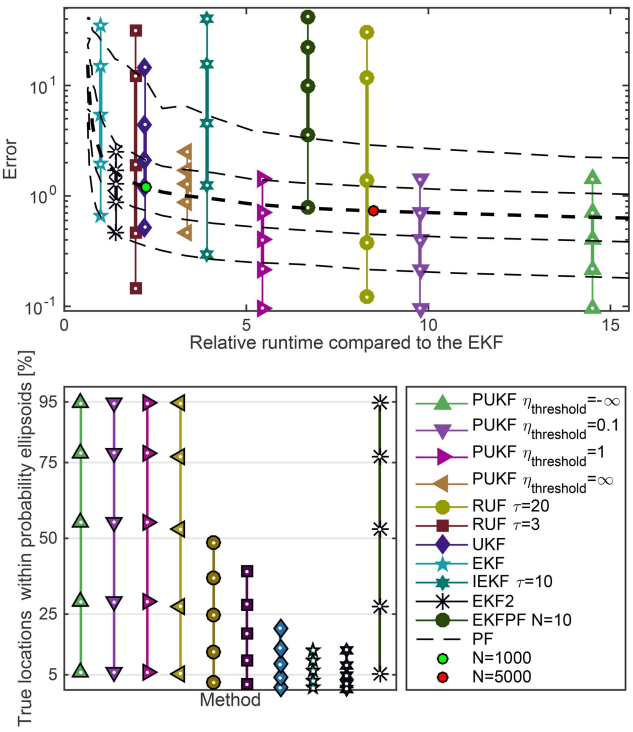


Fig. 4. Accuracy of different Kalman filter extensions in estimation with second order polynomial measurement model (38). In the top figure markers show the 5%, 25%, 50%, 75% and 95% quantiles of errors for each method for every estimated step. The errors are computed as the norm of the difference of the true and estimated mean. In the bottom figure the markers show how often the true state was within estimated error ellipsoids containing 5%, 25%, 50%, 75% and 95% of the probability mass.

the Kalman filter extensions by a large margin. When $\eta_{\text{threshold}} = \infty$ the whole measurement is processed at once and the result is the same as with EKF2, as expected. In this test scenario the PUKF performs clearly the best and methods that use EKF linearizations have very large errors. PUKF also outperforms PF with similar runtime.

In the bottom plot the accuracy of covariance estimates of different Kalman filter extensions are compared. For this plot we compute how often the true state is within the 5%, 25%, 50%, 75% and 95% ellipsoids of the Gaussian posterior. That is, a true location is within the p ellipsoid when

$$\chi_n^2((\mu - x_{\text{true}})^T P^{-1}(\mu - x_{\text{true}})) < p, \tag{40}$$

where x_{true} is the true state, μ and P are the posterior mean and covariance computed by the filter, and χ_n^2 is the cumulative density function of the chi-squared distribution with n degrees of freedom. The filter's error estimate is reliable when markers are close to the p values (dotted lines in the Figure). From the figure it is evident that PUKF and EKF2 have the most reliable error estimates and all other methods have too small covariance matrices.

The EKFPF did not perform wery well. This is probably caused by the inconsistency of EKF estimates

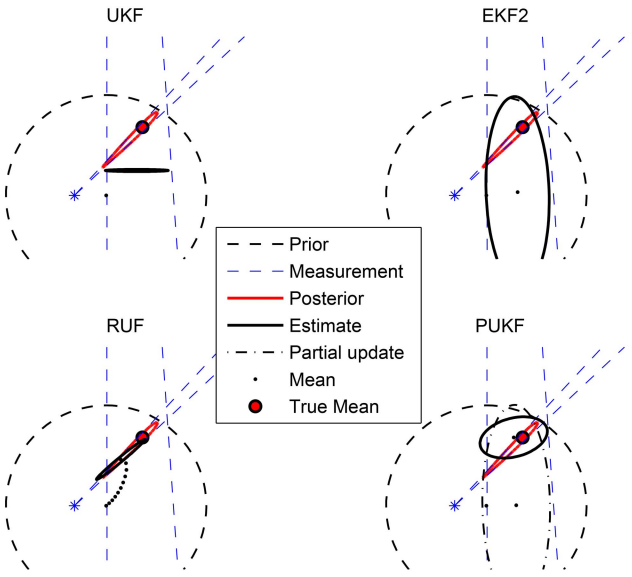


Fig. 5. Example situation of bearings positioning

that were used as the proposal distribution. We tested EKFPF also with 1000 particles. The estimation accuracy was similar to that obtained with a bootrstrap PF with 1000 particles, but the algorithm was much slower than other algorithms.

In our second test scenario the planar location of a target is estimated using bearing measurements. When the target is close to the sensor the measurement model is nonlinear, but when the target is far away the measurement becomes almost linear. The measurement model is

$$y = \text{atan2}(x_{[2]} - r_{[2]}, x_{[1]} - r_{[1]}) + \varepsilon^y, \qquad (41)$$

where atan2 is the four quadrant inverse tangent, r is the sensor location, and measurement noises are zero mean independent, with standard deviation of 2°. We choose the branch of atan2 so that evaluated values are as close as possible to the realized measurement value. In the test scenario two bearings measurements are used, one from a sensor close to the prior and the second from a sensor far away.

A representative initial state update using UKF, EKF2, RUF and PUKF is shown in Figure 5. This example is chosen so that the differences between estimates of different filters is clearly visible. The red line encloses the same probability mass of the true posterior as the $1 \cdot \sigma$ ellipses (black lines) of the Gaussian approximations computed with different Kalman filter extensions. The measurement from the distant sensor is almost linear within the prior and UKF uses it correctly, but the linearization of the estimate from the nearby sensor is not good and the resulting posterior is very narrow (EKF would be similar). In the EKF2 update the second order term of the measurement model from the nearby sensor is so large that EKF2 almost completely ignores that measurement and the prior is updated using only the measurement from the distant sensor. The iterative

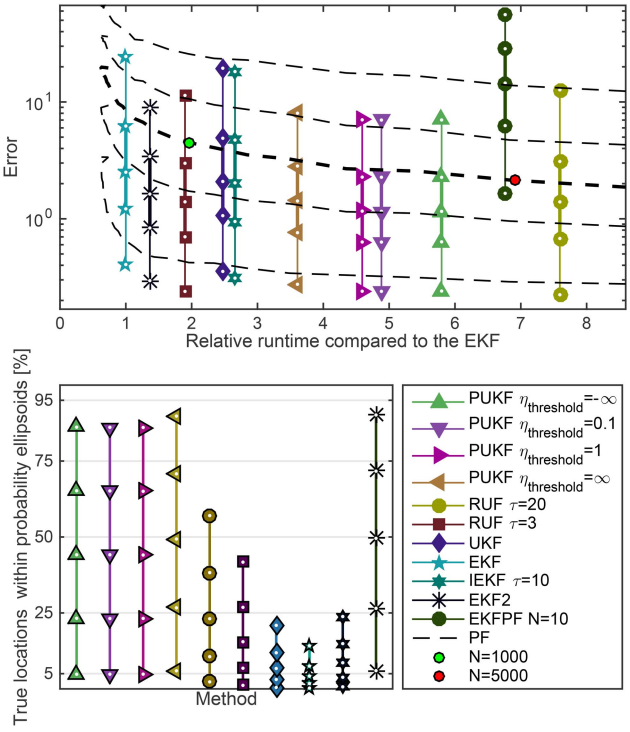


Fig. 6. Accuracy of different filters in bearings only tracking

update of RUF results in an estimate with small covariance that has similar shape as the true covariance. The mean of the true posterior is not inside the one-sigma ellipses of the RUF estimate and the mean is too close to the nearby sensor.

The first transformed measurement used by PUKF is almost the same as the measurement from the distant sensor and the estimate after the first partial update is similar to the EKF2 estimate. Because the estimate updated with the first measurement is further away from the nearby sensor the linearization of the second measurement is better and the posterior estimate is closer to the true posterior than with EKF2. The covariance estimate produced by PUKF is more conservative than the RUF of UKF covariances.

Figure 6 shows the statistics for this scenario. For this Figure the scenario was ran 1000 times using the same sensor locations and 10 step estimation with a 4-dimensional state model containing 2 position and 2 velocity dimensions. The prior has zero mean and covariance $10I$. The state transition function is

$$f(x) = \begin{bmatrix} I & I \\ \mathbf{0} & I \end{bmatrix} x + \varepsilon^x, \qquad (42)$$

where

$$\varepsilon^x \sim \text{N}\left(0, \begin{bmatrix} \frac{1}{300}I & \frac{1}{200}I \\ \frac{1}{200}I & \frac{1}{100}I \end{bmatrix}\right). \qquad (43)$$

Figure 6 shows that the PUKF provides the best accuracy. Interestingly RUF with 3 iterations has better accuracy than with 20 iterations. From the plot that shows the accuracy of the error estimates we can see

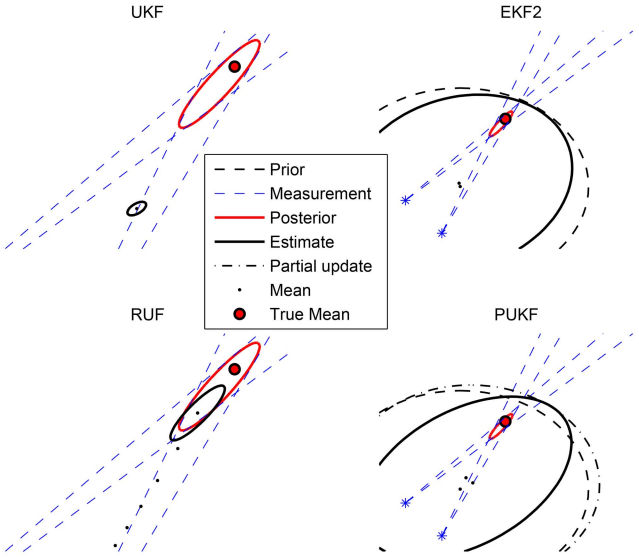


Fig. 7. Example of first update in bearings only tracking

that the PUKF and EKF2 have the best error estimates. Other methods have too optimistic covariance estimates. In this test scenario the PF did not manage to get good estimates with similar runtimes.

In the third test scenario we consider bearings only tracking with sensors close to each other. Otherwise the measurement model is the same as in the previous scenario The prior is as in previous test scenario. The state transition function is also (42) but the state transition noise is higher:

$$\varepsilon^x \sim \mathrm{N}\left(0, \begin{bmatrix} \frac{1}{3}I & \frac{1}{2}I \\ \frac{1}{2}I & I \end{bmatrix}\right). \tag{44}$$

The initial state and representative first updates are shown in Figure 7. In this Figure UKF and RUF estimates have very small covariances and so the plots are magnified. The UKF estimate mean is closer to the true mean than EKF2 and PUKF estimates, but the covariance of the estimate is very small. RUF has a better estimate than UKF, but the estimate is biased towards the sensor locations. Because both sensors are nearby and have large second order terms EKF2 and PUKF estimates do not differ much.

Results for estimating 10 step tracks 1000 times are shown in Figure 8. In this case the RUF has the best accuracy. In PUKF there is only very small differences whether all of the measurement are used at once or a nonlinearity threshold is used. This means that in this measurement geometry the partitioned update does not improve accuracy. EKF2 has better covariance estimates than the numerical update PUKF even though it has larger errors. The covariance estimates produced by RUF were again too small. In this test the PF has better accuracy than the Kalman filter extensions.

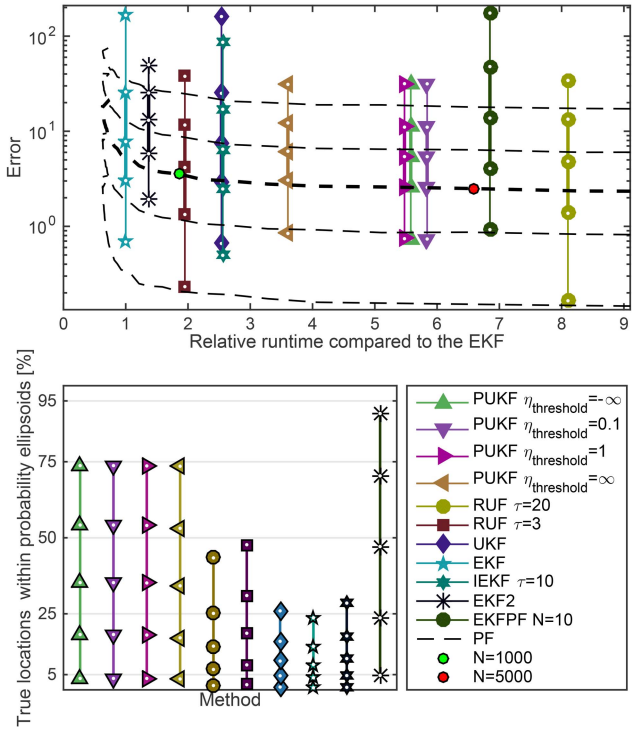


Fig. 8. Results for bearings only tracking with sensors close to each other

To further evaluate the accuracy of the estimates, we compare Kullback-Leiber (KL) divergences of estimates. The KL divergence is defined as

$$\int \ln\left(\frac{p(x)}{q(x)}\right) p(x)dx, \tag{45}$$

where $p(x)$ is the pdf of the true distribution and $q(x)$ is the pdf of the approximate distribution [20]. We computed the KL divergence for position dimensions. The true pdf is approximated using a 50×50 grid. The probability for each grid is computed as the sum of particle weights of a PF particles within each cell. For this we used 10^6 particles. Table I shows the median Kullback-Leibler divergences for each method in the two bearings measurement test scenarios. PUKF has the smallest KL divergence in both test scenarios.

V. CONCLUSIONS AND FUTURE WORK

In this paper we presented a new extension of the Kalman filter: the Partitioned Update Kalman Filter (PUKF). The proposed filter evaluates the nonlinearity of a multidimensional measurement and transforms the measurement model so that some dimensions of the measurement model have as low nonlinearity as possible. PUKF does the update of the state using the measurement in parts, so that the parts with the smallest amounts of nonlinearity are processed first. The proposed algorithm improves estimation results when measurements are such that the partial update reduces the nonlinearity of the remaining part. According to the simulated tests the PUKF improves the estimates when

TABLE I
Median Kullback-Leibler divergences of position dimensions in the two bearings only tests

Method	PUKF $\eta_{\text{threshold}} = -\infty$	PUKF $\eta_{\text{threshold}} = 0.1$	PUKF $\eta_{\text{threshold}} = 1$	PUKF $\eta_{\text{threshold}} = \infty$	RUF $\tau = 20$	RUF $\tau = 3$	UKF	EKF	IEKF $\tau = 10$	EKF2
First bearings test	0.62	0.62	0.63	1.07	0.99	1.70	5.20	14.42	6.69	1.16
Second bearings test	2.14	2.14	2.14	2.33	3.56	2.53	9.36	10.97	8.84	2.60

measurements can be transformed so that an informative linear part of the measurement can be extracted.

In many practical situations the almost linear part could be extracted manually. For example, Global Positioning System (GPS) measurements are almost linear and they could be applied before other measurements. The proposed algorithm does the separation automatically and when using the numerical algorithm for computing the prediction and update analytical differentiation is not required.

In our tests the estimated covariances produced by EKF2 and PUKF were the most accurate. In [11] it was claimed that RUF produces more accurate error estimates than EKF2. Their results were based on comparing $3 \cdot \sigma$ errors in 1D estimation. In this comparison 92% of samples should be within the $3 \cdot \sigma$ range. For their results they had only 100 samples and from the resulting figure it is hard to see how many samples exactly are within the range, but for EKF2 most of the points are within the range and some are outside.

In our tests, among other Kalman filter extensions RUF had good accuracy, but it provided too small covariance matrices. In future it could be interesting to extend RUF [11] to use EKF2-like statistical second order linearization and then combine it with the proposed algorithm.

Another use case for PUKF would be merging it with the Binomial Gaussian mixture filter [21]. This filter decorrelates measurements and uses nonlinearity measure (30) as an indication of whether the measurement model is so nonlinear that the prior component should be split. By decorrelating measurements with the algorithm proposed in this paper and doing the partial updates for the most linear components first, unnecessary splits could be avoided.

APPENDIX A INVARIANCE OF EKF AND EKF2 TO A LINEAR TRANSFORMATION OF THE MEASUREMENT MODEL

The second order Taylor polynomial approximation of the measurement function is

$$h(x) = h(\mu^-) + J^h(x-\mu^-) + \frac{1}{2}\begin{bmatrix} (x-\mu^-)^T H^{h_1}(x-\mu^-) \\ (x-\mu^-)^T H^{h_2}(x-\mu^-) \\ \vdots \\ (x-\mu^-)^T H^{h_d}(x-\mu^-) \end{bmatrix} + \varepsilon^y \quad (46)$$

where Jacobian J^h and Hessians H^h are evaluated at prior mean, ε^y is the measurement function noise.

In the linear transformation the measurement function (46) is multiplied by D. The second order approximation is

$$\hat{h}(x) = Dh(x) = Dh(\mu^-) + DJ^h(x-\mu^-) + \frac{1}{2}D\begin{bmatrix} (x-\mu^-)^T H_1^h(x-\mu^-) \\ (x-\mu^-)^T H_2^h(x-\mu^-) \\ \vdots \\ (x-\mu^-)^T H_n^h(x-\mu^-) \end{bmatrix} + D\varepsilon^y \quad (47)$$

The transformed Jacobian is

$$\hat{J}^h = DJ^h \quad (48)$$

and ith transformed Hessian is

$$\hat{H}_i^h = \sum_{k=1}^n D_{[i,k]} H_k^h. \quad (49)$$

The terms ξ^h and Ξ^h are

$$\begin{aligned} \hat{\xi}^h &= \begin{bmatrix} \operatorname{tr} P^- \hat{H}_1^h \\ \operatorname{tr} P^- \hat{H}_2^h \\ \vdots \\ \operatorname{tr} P^- \hat{H}_n^h \end{bmatrix} = \begin{bmatrix} \operatorname{tr} P^- \sum_{k=1}^n D_{[1,k]} H_k^h \\ \operatorname{tr} P^- \sum_{k=1}^n D_{[2,k]} H_k^h \\ \vdots \\ \operatorname{tr} P^- \sum_{k=1}^n D_{[n,k]} H_k^h \end{bmatrix} \\ &= D\begin{bmatrix} \operatorname{tr} P^- H_1^h \\ \operatorname{tr} P^- H_2^h \\ \vdots \\ \operatorname{tr} P^- H_n^h \end{bmatrix} = D\xi^h \end{aligned} \quad (50)$$

$$\begin{aligned} \hat{\Xi}_{[i,j]}^h &= \operatorname{tr} P^- \hat{H}_i^h P^- \hat{H}_j^h \\ &= \operatorname{tr} P^- \left(\sum_{k=1}^n D_{[i,k]} H_k^h\right) P^- \left(\sum_{l=1}^n D_{[i,l]} H_l^h\right) \\ &= \sum_{k=1}^n \sum_{l=1}^n D_{[i,k]} D_{[j,l]} \operatorname{tr} P^- H_k^h P^- H_l^h \\ \Rightarrow \hat{\Xi}^h &= D\Xi^h D^T \end{aligned} \quad (51)$$

For EKF update these terms are replaced with zero matrices.

Now using these transformed quantities in the EKF2 update equations (9–13) gives

$$\hat{y}^- = \hat{h}(\mu^-) + \tfrac{1}{2}\hat{\xi}^h = D(h(\mu^-) + \tfrac{1}{2}\xi^h) \quad (52)$$

$$\hat{S} = DJ^h P^- J^{h^T} D^T + \tfrac{1}{2} D\Xi^h D^T + DRD^T \quad (53)$$
$$= DSD^T$$
$$\hat{K} = P^- \hat{J}^{h^T} \hat{S}^{-1}$$
$$= P^- J^{h^T} D^T D^{-T} S^{-1} D^{-1} = P^- J^{h^T} S^{-1} D^{-1} \quad (54)$$
$$= KD^{-1}$$
$$\hat{\mu}^+ = \mu^- + \hat{K}[Dy - Dh(\mu^-) - \tfrac{1}{2} D\xi^h]$$
$$= \mu^- + K(y - J^h \mu^- - \tfrac{1}{2}\xi^h) \quad (55)$$
$$= \mu^+$$
$$\hat{P}^+ = P^- - \hat{K}\hat{S}\hat{K}^T$$
$$= P^- - KD^{-1} DSD^T (KD^{-1})^T = P^- - KSK^T$$
$$= P^+, \quad (56)$$

which shows that the posterior is the same as with the non-transformed measurements.

APPENDIX B PROOF THAT THE NONLINEARITIES ARE MINIMIZED

Let Ξ^h be a diagonal matrix containing nonlinearity values ordered ascending on the diagonal and let measurement noise covariance matrix be identity matrix $R = I$. We will show that the smallest diagonal element of Ξ^h is as small as possible under a linear transformation that preserves $R = I$ and further that the second smallest diagonal element is as small as possible, when the smallest is as small as possible etc.

If the measurement model is transformed by multiplying it with V, the transformed variables are $\hat{\Xi}^h = V\Xi^h V^T$ and $R = VIV^T = VV^T$. Because we want to have $R = I$, V has to be unitary. The ith diagonal element of the transformed matrix is $v_i^T \Xi^h v_i = \sum_{j=1}^d v_{i,[j]}^2 \Xi_{[j,j]}^h$, where v_i is the ith column of V. Because V is unitary, we have $\sum_{j=1}^d v_{i,[j]}^2 = 1$ and the ith diagonal element of the transformed matrix $\hat{\Xi}^h$ is

$$\sum_{j=1}^d v_{i,[j]}^2 \Xi_{[j,j]}^h \geq \sum_{j=1}^d v_{i,[j]}^2 \min_j \{\Xi_{[j,j]}^h\} = \min_j \{\Xi_{[j,j]}^h\}. \quad (57)$$

Thus, the new diagonal element cannot be smaller than the smallest diagonal element of Ξ^h.

If the smallest element is in the first element of the diagonal the possible transformation for the second smallest element is

$$\hat{\Xi}^h = \begin{bmatrix} 1 & 0^T \\ 0 & V \end{bmatrix} \Xi^h \begin{bmatrix} 1 & 0^T \\ 0 & V^T \end{bmatrix}. \quad (58)$$

With the same reasoning as given already the second diagonal has to be already the smallest possible. Inductively this applies to all diagonal elements.

REFERENCES

[1] H. W. Sorenson and D. L. Alspach
Recursive Bayesian estimation using Gaussian sums,
Automatica, vol. 7, no. 4, pp. 465–479, 1971. doi: 10.1016/0005-1098(71)90097-5.

[2] J. H. Kotecha and P. Djuric
Gaussian particle filtering,
IEEE Transactions on Signal Processing, vol. 51, no. 10, pp. 2592–2601, Oct 2003. doi: 10.1109/TSP.2003.816758.

[3] J. Steinbring and U. Hanebeck
Progressive Gaussian filtering using explicit likelihoods,
in *17th International Conference on Information Fusion (FUSION)*, July 2014, pp. 1–8.

[4] A. Gelb
Applied Optimal Estimation.
MIT Press, 1974.

[5] K. Ito and K. Xiong
Gaussian filters for nonlinear filtering problems,
IEEE Transactions on Automatic Control, vol. 45, no. 5, pp. 910–927, May 2000. doi: 10.1109/9.855552.

[6] S. J. Julier and J. K. Uhlmann
A New Extension of the Kalman Filter to Nonlinear Systems,
in *International Symposium Aerospace/Defense Sensing, Simulation and Controls*, vol. 3068, 1997. doi: 10.1117/12.280797 pp. 182–193.

[7] I. Arasaratnam and S. Haykin
Cubature Kalman filters,
IEEE Transactions on Automatic Control, vol. 54, no. 6, pp. 1254–1269, June 2009. doi: 10.1109/TAC.2009.2019800.

[8] S. Dmitriev and L. Šimelevič
A generalized Kalman filter with repeated linearization and its use in navigation over geophysical fields,
Avtomatika i Telemekhanika, no. 4, pp. 50–55, 1978.

[9] A. H. Jazwinski
Stochastic Processes and Filtering Theory,
ser. Mathematics in Science and Engineering. Academic Press, 1970, vol. 64.

[10] A. Garcia-Fernandez, L. Svensson, and M. Morelande
Iterated statistical linear regression for Bayesian updates,
in *17th International Conference on Information Fusion (FUSION)*, July 2014, pp. 1–8.

[11] R. Zanetti
Recursive update filtering for nonlinear estimation,
IEEE Transactions on Automatic Control, vol. 57, no. 6, pp. 1481–1490, June 2012. doi: 10.1109/TAC.2011.2178334.

[12] M. Nørgaard, N. K. Poulsen, and O. Ravn
New developments in state estimation for nonlinear systems,
Automatica, vol. 36, no. 11, pp. 1627–1638, Nov. 2000. doi: 10.1016/S0005-1098(00)00089-3.

[13] R. F. Stengel
Stochastic optimal control,
Theory and Application, pp. 1–638, 1986.

[14] S. Ali-Löytty
Box Gaussian mixture filter,
IEEE Transactions on Automatic Control, vol. 55, no. 9, pp. 2165–2169, September 2010. doi: 10.1109/TAC.2010.2051486.

[15] M. Raitoharju
"Linear models and approximations in personal positioning,"
Ph.D. dissertation, Tampere University of Technology, 2014. [Online]. Available: http://urn.fi/URN:ISBN:978-952-15-3421-8.

[16] M. Raitoharju and S. Ali-Löytty
An adaptive derivative free method for Bayesian posterior approximation,
IEEE Signal Processing Letters, vol. 19, no. 2, pp. 87–90, Feb 2012. doi: 10.1109/LSP.2011.2179800.

[17] M. Raitoharju
"Matlab implementation of the partitioned update Kalman filter,"
http://www.mathworks.com/matlabcentral/fileexchange/51838-partitioned-update-kalman-filter, 2015, [Online; accessed 30-June-2015].

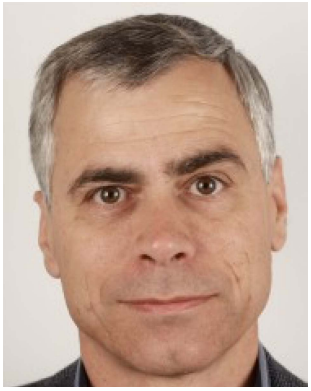
[18] J. Carpenter, P. Clifford, and P. Fearnhead
Improved particle filter for nonlinear problems,
IEE Proceedings—Radar, Sonar and Navigation, vol. 146, pp. 2–7, February 1999. doi: 10.1049/ip-rsn:19990255.

[19] J. F. de Freitas, M. Niranjan, A. H. Gee, and A. Doucet
Sequential Monte Carlo methods to train neural network models,
Neural computation, vol. 12, no. 4, pp. 955–993, 2000.

[20] S. Kullback and R. A. Leibler
On information and sufficiency,
Ann. Math. Statist., vol. 22, no. 1, pp. 79–86, 1951.

[21] M. Raitoharju, S. Ali-Löytty, and R. Piché
Binomial Gaussian mixture filter,
EURASIP Journal on Advances in Signal Processing, vol. 2015, no. 1, p. 36, 2015. doi: 10.1186/s13634-015-0221-2.

Matti Raitoharju received M.Sc. and Ph.D. degrees in Mathematics in Tampere University of Technology, Finland, in 2009 and 2014 respectively. He works as a postdoctoral researcher in the Department of Automation Science and Engineering at Tampere University of Technology. His research interests include estimation algorithms and mathematical modelling.



Robert Piché received the Ph.D. degree in civil engineering in 1986 from the University of Waterloo, Canada. Since 2004 he is professor at Tampere University of Technology, Finland. His scientific interests include mathematical and statistical modelling, systems theory, and applications in positioning, computational finance, and mechanics.

Juha Ala-Luhtala received the M.Sc. (Tech.) degree in information technology from Tampere University of Technology in 2011. He is currently working towards finishing his Ph.D. degree in mathematics at the same university. His research interests include Bayesian inference in stochastic state-space models, and applications in positioning and navigation.

Simo Ali-Löytty is a University Lecturer at Tampere University of Technology. He received his M.Sc. degree in 2004 and his Ph.D. degree in 2009 from Tampere University of Technology. His research interests are mathematical modelling and industrial mathematics especially Bayesian filters in personal positioning.

Sigma-Point Filtering and Smoothing Based Parameter Estimation in Nonlinear Dynamic Systems

JUHO KOKKALA
ARNO SOLIN
SIMO SÄRKKÄ

We consider approximate maximum likelihood parameter estimation in nonlinear state-space models. We discuss both direct optimization of the likelihood and expectation-maximization (EM). For EM, we also give closed-form expressions for the maximization step in a class of models that are linear in parameters and have additive noise. To obtain approximations to the filtering and smoothing distributions needed in the likelihood-maximization methods, we focus on using Gaussian filtering and smoothing algorithms that employ sigma-points to approximate the required integrals. We discuss different sigma-point schemes based on the third, fifth, seventh, and ninth order unscented transforms and the Gauss-Hermite quadrature rule. We compare the performance of the methods in two simulated experiments: a univariate nonlinear growth model as well as tracking of a maneuvering target. In the experiments, we also compare against approximate likelihood estimates obtained by particle filtering and extended Kalman filtering based methods. The experiments suggest that the higher-order unscented transforms may in some cases provide more accurate estimates.

Manuscript received March 31, 2015; revised August 31, 2015; released for publication November 9, 2015.

Refereeing of this contribution was handled by Jindrich Dunik.

Authors' addresses: Aalto University, Espoo, Finland. (e-mail: juho.kokkala@aalto.fi, arno.solin@aalto.fi, simo.sarkka@aalto.fi).

This work was supported by grants from the Academy of Finland (266940, 273475) and by the Emil Aaltonen foundation.

The authors declare no conflict-of-interest.

I. INTRODUCTION

This paper is an extended version of our article [24] where we considered parameter estimation in state-space models using expectation-maximization (EM) algorithms based on sigma-point and particle smoothers. In this paper, we extend our interest from EM algorithms to so called direct maximum likelihood based parameter estimation methods, where instead of using the EM algorithm, the marginal likelihood of the parameters is directly approximated using nonlinear filtering methods. In particular, we focus our interest to sigma-point filters which use high-order unscented Kalman filters and Gauss-Hermite Kalman filters to approximate the likelihood surface.

We consider state-space models of the following form:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}, \boldsymbol{\theta}) + \mathbf{q}_{k-1}, \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, \boldsymbol{\theta}) + \mathbf{r}_k, \end{aligned} \tag{1}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is the discrete-time state sequence with an initial distribution $\mathbf{x}_0 \sim \mathbb{N}(\mathbf{x}_0 \mid \mathbf{m}_0(\boldsymbol{\theta}), \mathbf{P}_0(\boldsymbol{\theta}))$, $\mathbf{y}_k \in \mathbb{R}^d$ is the measurement sequence, $\mathbf{q}_k \sim \mathbb{N}(\mathbf{0}, \mathbf{Q}(\boldsymbol{\theta}))$ is the Gaussian process noise sequence, $\mathbf{r}_k \sim \mathbb{N}(\mathbf{0}, \mathbf{R}(\boldsymbol{\theta}))$ is the Gaussian measurement error sequence, and $\boldsymbol{\theta} \in \mathbb{R}^m$ is a static parameter vector. Typically, one is interested in computing the posterior distribution of the state $\mathbf{x}_k$ given measurements up to time k, $p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_k)$, known as the filtering problem, or computing the posterior distribution of the state $\mathbf{x}_k$ given all measurements, $p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$, where $k \leq T$, known as the smoothing problem. In the general case, analytical expressions do not exist and we have to resort to approximative algorithms such as the sigma-point methods. See, for example, [32] for a general overview of Bayesian filtering and smoothing.

While many filtering and smoothing algorithms are formulated assuming fixed static parameters $\boldsymbol{\theta}$, in practice optimal values for these parameters are generally unknown. Therefore, methods for estimating the parameters from the data are desired. In this paper, we concentrate on maximum-likelihood methods, where the parameters are selected by maximizing the marginal likelihood, or equivalently the logarithm of the marginal likelihood, that is

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \log p(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}). \tag{2}$$

In linear systems with additive Gaussian noise, the likelihood can be evaluated using the Kalman filter [17, 21]. Many optimization algorithms utilize also the gradient of the log-likelihood. The gradient can be evaluated by so-called sensitivity equations, a recursion that is obtained by differentiating the Kalman filter recursion [15]. Alternatively, due to Fisher's identity, the gradient may be evaluated by differentiating an auxiliary function that can be computed during the smoothing

pass [28, 35]. Instead of directly optimizing the likelihood, the expectation-maximization (EM) algorithm [9] can be used to optimize parameters. The EM algorithm consists of iterating the expectation (E) step where a bound of the log-likelihood is computed using the current parameter estimates, and the maximization (M) step where the bound is maximized with respect to the parameters. The evaluation of the bound in the E-step is obtained by solving the smoothing problem. See [36] for a discussion of applying the EM algorithm in state-space models. Note that in the linear-Gaussian case both gradient evaluation methods as well as the EM algorithm in principle converge to the same solution, namely, the parameter value that maximizes the log-likelihood.

In this paper, our interest lies in estimating the static parameters by maximum-likelihood estimation in the case of nonlinear state-space models with additive Gaussian noise, that is model (1). Formally, the marginal likelihood can be computed by marginalizing out the states from the joint distribution of the measurements and states using nonlinear filtering equations and the prediction error decomposition (see, e.g., [5, 32]), leading to similar methods as in the linear-Gaussian case. However, since the state variables $\mathbf{x}$ cannot in general be marginalized out analytically, one needs to employ approximative methods. In the so called direct likelihood methods, the likelihood is approximated directly using approximative nonlinear filtering methods (see, e.g., [5, 22, 29, 32, 37]) and its maximum is found via nonlinear optimization. Similarly, the expectation-maximization (EM) algorithm can be employed, but the E-step cannot be solved exactly. Instead, the E-step is approximated with nonlinear smoothing algorithms (see, e.g., [12, 24, 31, 34, 41]).

The aim of this paper is to extend the results of our paper [24] by showing how high-order (i.e., third, fifth, seventh, and ninth order) unscented transforms and Gauss-Hermite integration based sigma-point methods can be used for approximate direct likelihood and EM-based parameter estimation in nonlinear state-space models. For EM, we also give closed-form expressions for the maximization step in a class of models that are linear in parameters and have additive noise. We compare the unscented transform and Gauss-Hermite based sigma-point methods to linearization-based extended Kalman filter algorithms and Monte Carlo based particle filtering algorithms. We also provide an algorithm for computing the gradients required by the gradient-based optimization methods. Although we focus on maximum likelihood estimation, the provided algorithms can be easily extended to computation of maximum a posteriori estimates by including a prior distribution to the objective function.

The remainder of the paper is organized as follows. In Section II, we present the sigma-point filters and smoothers based on assumed Gaussian density filtering and smoothing. In Section III, we discuss parameter estimation based on direct likelihood maximization and EM. Numerical experiments are presented in Section IV. Finally, Section V presents concluding discussion.

II. SIGMA-POINT FILTERING AND SMOOTHING

Under our interpretation, sigma-point filtering and smoothing is derived by assuming Gaussian approximations for the state distributions, which enables the use of a Kalman filter like filtering recursion and a Rauch-Tung-Striebel backward pass for the smoothing distributions. The Gaussian filtering and smoothing equations contain expectations over Gaussian distributions which cannot be generally evaluated in closed form. The sigma-points arise from approximating these Gaussian integrals by weighted sums determined by some cubature (multi-dimensional quadrature) formula. Hence, we interpret the different sigma-point methods as incarnations of different integral approximations.

In the following, we first present the assumed Gaussian density filtering and smoothing framework. Then, we discuss various different cubature rules for approximating the Gaussian integrals. Finally, we show how the cubature rules are applied to the assumed Gaussian density filtering and smoothing framework to obtain the filtering and smoothing equations explicitly in the sigma-point form.

A. General Gaussian Filtering and Smoothing

Assumed density Gaussian filtering (see [16, 32, 33]) is based on assuming that the filtering distributions are approximately Gaussian, that is, assuming means $\mathbf{m}_{k|k}$ and covariances $\mathbf{P}_{k|k}$ such that

$$p(\mathbf{x}_k \mid \mathbf{y}_{1:k}) \approx \mathbb{N}(\mathbf{x}_k \mid \mathbf{m}_{k|k}, \mathbf{P}_{k|k}) \tag{3}$$

as well as means $\mathbf{m}_{k|k+1}$ and covariances $\mathbf{P}_{k|k+1}$ such that

$$p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:k}) \approx \mathbb{N}(\mathbf{x}_{k+1} \mid \mathbf{m}_{k|k+1}, \mathbf{P}_{k|k+1}). \tag{4}$$

The filtering equations of the resulting Gaussian filter [16, 43] consist of a *prediction step* and an *update step*. In the prediction step, we compute the state mean and covariance of the distribution $p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1})$ using the Gaussian approximation for $p(\mathbf{x}_{k-1} \mid \mathbf{y}_{1:k-1})$. The resulting equations are

$$\begin{aligned}
\mathbf{m}_{k|k-1} &= \mathbb{E}[\mathbf{f}(\mathbf{x}_{k-1})], \\
\mathbf{P}_{k|k-1} &= \mathbb{E}[(\mathbf{f}(\mathbf{x}_{k-1}) - \mathbf{m}_{k|k-1}) \\
&\qquad \times (\mathbf{f}(\mathbf{x}_{k-1}) - \mathbf{m}_{k|k-1})^{\mathrm{T}}] + \mathbf{Q},
\end{aligned} \tag{5}$$

where the expectations are taken with respect to the distribution $\mathbf{x}_{k-1} \sim \mathbb{N}(\mathbf{m}_{k-1|k-1}, \mathbf{P}_{k-1|k-1})$.

In the corresponding update step, we assume a Gaussian density $p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}) = \mathbb{N}(\mathbf{x}_k \mid \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1})$ and

compute the state mean and covariance for the distribution $p(\mathbf{x}_k \mid \mathbf{y}_{1:k})$. The resulting equations are

$$\begin{aligned}
\boldsymbol{\mu}_k &= \mathbb{E}[\mathbf{h}(\mathbf{x}_k)], \\
\mathbf{S}_k &= \mathbb{E}[(\mathbf{h}(\mathbf{x}_k) - \boldsymbol{\mu}_k)(\mathbf{h}(\mathbf{x}_k) - \boldsymbol{\mu}_k)^{\mathrm{T}}] + \mathbf{R}, \\
\mathbf{C}_k &= \mathbb{E}[(\mathbf{x}_k - \mathbf{m}_{k|k-1})(\mathbf{h}(\mathbf{x}_k) - \boldsymbol{\mu}_k)^{\mathrm{T}}], \\
\mathbf{K}_k &= \mathbf{C}_k \mathbf{S}_k^{-1}, \\
\mathbf{m}_{k|k} &= \mathbf{m}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \boldsymbol{\mu}_k), \\
\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^{\mathrm{T}},
\end{aligned} \tag{6}$$

where $\boldsymbol{\mu}_k$ and $\mathbf{S}_k$ are the expectation and variance, respectively, of the measurement $\mathbf{y}_k$ and $\mathbf{C}_k$ is the covariance of the state $\mathbf{x}_k$ and the measurement $\mathbf{y}_k$. These expectations are taken with respect to the distribution $\mathbf{x}_k \sim \mathbb{N}(\mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1})$.

The smoothing distributions $p(\mathbf{x}_k \mid \mathbf{y}_{1:T})$ are obtained from a backward pass, that is, starting from $k = T$ and iterating backwards in time. On each step, the smoothing density of $\mathbf{x}_{k+1}$ is assumed to be Gaussian: $p(\mathbf{x}_{k+1} \mid \mathbf{y}_{1:T}) = \mathbb{N}(\mathbf{x}_{k+1} \mid \mathbf{m}_{k+1|T}, \mathbf{P}_{k+1|T})$. The mean and covariance for $p(\mathbf{x}_k \mid \mathbf{y}_{1:T})$ are then computed from the previous Gaussian smoothing density and the Gaussian filtering densities using the Rauch-Tung-Striebel backward pass [13, 30] as follows [33]:

$$\begin{aligned}
\mathbf{m}_{k+1|k} &= \mathbb{E}[\mathbf{f}(\mathbf{x}_k)], \\
\mathbf{P}_{k+1|k} &= \mathbb{E}[(\mathbf{f}(\mathbf{x}_k) - \mathbf{m}_{k+1|k}) \\
&\qquad \times (\mathbf{f}(\mathbf{x}_k) - \mathbf{m}_{k+1|k})^{\mathrm{T}}] + \mathbf{Q}, \\
\mathbf{D}_{k+1} &= \mathbb{E}[(\mathbf{x}_k - \mathbf{m}_{k|k})(\mathbf{f}(\mathbf{x}_k) - \mathbf{m}_{k+1|k})^{\mathrm{T}}], \\
\mathbf{G}_k &= \mathbf{D}_{k+1}[\mathbf{P}_{k+1|k}]^{-1}, \\
\mathbf{m}_{k|T} &= \mathbf{m}_{k|k} + \mathbf{G}_k(\mathbf{m}_{k+1|T} - \mathbf{m}_{k+1|k}), \\
\mathbf{P}_{k|T} &= \mathbf{P}_{k|k} + \mathbf{G}_k(\mathbf{P}_{k+1|T} - \mathbf{P}_{k+1|k})\mathbf{G}_k^{\mathrm{T}},
\end{aligned} \tag{7}$$

where $\mathbf{G}_k$ is known as the smoother gain and the expectations are taken with respect to the distribution $\mathbf{x}_k \sim \mathbb{N}(\mathbf{m}_{k|k}, \mathbf{P}_{k|k})$. The pairwise joint smoothing distributions $p(\mathbf{x}_k, \mathbf{x}_{k-1} \mid \mathbf{y}_{1:T})$ are also of interest since they are used in the expectation-maximization algorithm (see Section III-B). Gaussian approximations for these distributions are obtained as a by-product of the smoothing backward pass results as follows (see, e.g., [32, p. 189]).

$$\begin{aligned}
&p(\mathbf{x}_k, \mathbf{x}_{k-1} \mid \mathbf{y}_{1:T}) \approx \\
&\quad \mathbb{N}\left(\begin{pmatrix} \mathbf{x}_k \\ \mathbf{x}_{k-1} \end{pmatrix} \middle| \begin{pmatrix} \mathbf{m}_{k|T} \\ \mathbf{m}_{k-1|T} \end{pmatrix}, \begin{pmatrix} \mathbf{P}_{k|T} & \mathbf{P}_{k|T}\mathbf{G}_{k-1}^{\mathrm{T}} \\ \mathbf{G}_{k-1}\mathbf{P}_{k|T} & \mathbf{P}_{k-1|T} \end{pmatrix} \right).
\end{aligned} \tag{8}$$

B. Approximating the Gaussian Integrals

As we saw in the previous section, during the evaluation of the prediction and update steps of the Gaussian filter and smoother, we need to solve a set of Gaussian integrals on each step. These integrals are of the following form:

$$\mathbb{E}[\mathbf{g}(\mathbf{x})] = \int_{\mathbb{R}^n} \mathbf{g}(\mathbf{x}) \mathbb{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P}) d\mathbf{x}, \tag{9}$$

where $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^d$ is the integrand and the weighting function $\mathbb{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P})$ is a multi-dimensional Gaussian density with mean $\mathbf{m}$ and covariance matrix $\mathbf{P}$. In this paper, these integrals are computed by using multi-dimensional generalizations of Gaussian quadratures—also referred to as Gaussian *cubatures* [6]. They give approximations of the form

$$\mathbb{E}[\mathbf{g}(\mathbf{x})] \approx \sum_i w_i \mathbf{g}(\mathbf{x}_i), \tag{10}$$

where the weights w_i and sigma-points $\mathbf{x}_i$ are functions of the mean $\mathbf{m}$ and covariance $\mathbf{P}$ of the Gaussian weighting function. The sigma-points are positioned as follows:

$$\mathbf{x}_i = \mathbf{m} + \mathbf{L}\boldsymbol{\xi}_i, \tag{11}$$

where $\boldsymbol{\xi}_i$ are method specific unit sigma-points, and $\mathbf{L}$ is a matrix square-root factor such that $\mathbf{P} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$ (e.g., the Cholesky decomposition of $\mathbf{P}$). The differences in the methods come from different choices of weights and unit sigma-points.

In the following we briefly introduce a number of schemes for choosing the weights and sigma-points. The difference between these schemes stems from a trade-off between the number of sigma-points (required function evaluations) versus precision in the approximation. The degree of approximation is quantified by the highest polynomial order, p, for which the method is exact.

Unscented transform. The *unscented transform* (UT, [19, 20]) uses a set of $2n + 1$ cubature points located in the center and on the surface of an n-sphere. The radius and the weights can be controlled using a set of parameters. The cubature points are given by:

$$\begin{aligned}
\boldsymbol{\xi}_0 &= \mathbf{0}, \\
\boldsymbol{\xi}_i &= \begin{cases} \sqrt{\lambda + n}\,\mathbf{e}_i, & i = 1, \ldots, n, \\ -\sqrt{\lambda + n}\,\mathbf{e}_{i-n}, & i = n+1, \ldots, 2n, \end{cases}
\end{aligned}$$

where $\mathbf{e}_i$ denotes a unit vector to the direction of coordinate axis i, and the weights are defined as follows:

$$\begin{aligned}
w^{(0)} &= \begin{cases} \dfrac{\lambda}{n + \lambda}, & \text{for mean terms,} \\ \dfrac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta), & \text{for covariance terms,} \end{cases} \\
w^{(i)} &= \frac{1}{2(n + \lambda)}, \qquad i = 1, \ldots, 2n,
\end{aligned}$$

where $\lambda = \alpha^2(n + \kappa) - n$ and α, β, and κ are parameters of the method.

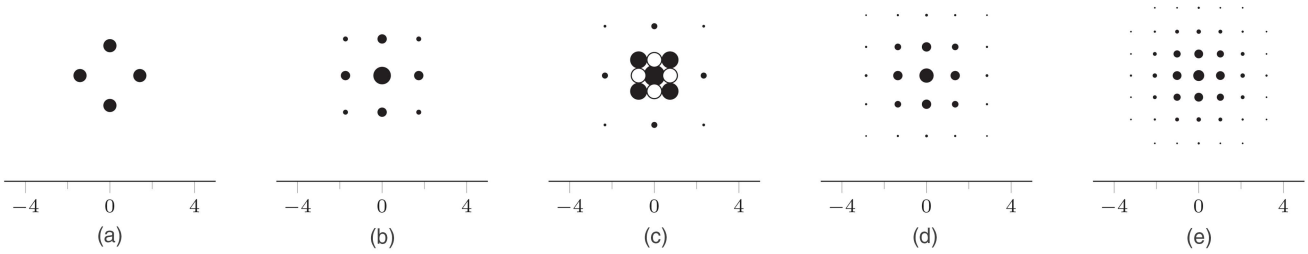


Fig. 1. Unit sigma-points in two dimensions for each of the methods. The absolute value of the weights are indicated by the point size, positive weights being black, negative weights white. (a) The symmetric cubature rule of order $p = 3$ with 4 points. (b) The symmetric cubature rule of order $p = 5$ with 9 points. (c) The symmetric cubature rule of order $p = 7$ with 17 points. (d) The symmetric cubature rule of order $p = 9$ with 25 points. (e) For comparison, the Gauss-Hermite (order $p = 9$) sigma-points (81 points, many of which with very small weights) are also shown.

Symmetric, 3rd order. A widely applicable sigma-point scheme is constructed by setting the unscented transform parameters to $\alpha = \pm 1$, $\beta = 0$, and $\kappa = 0$ [43]. This is also known as the 3rd order symmetric spherical-radial cubature method (CKF, [3]; see [38] for the explicit connection). This method utilizes a scaled and rotated set of $2n$ points, which are selected to be at the intersections of an n-sphere and the coordinate axes:

$$\boldsymbol{\xi}_i = \begin{cases} \sqrt{n}\mathbf{e}_i, & i = 1,2,\ldots,n, \\ -\sqrt{n}\mathbf{e}_{i-n}, & i = n+1,\ldots,2n. \end{cases}$$

The weights are defined as $w_i = 1/(2n)$ for $i = 1,2,\ldots, 2n$. The number of evaluation points is a linear function of the state dimension. The corresponding sigma-point filter is referred to as UKF 3.

Symmetric, 5th order. Building upon the work of McNamee and Stenger [26], it is possible to find explicit fully symmetric integration formulas of higher order than three. These integration schemes are exact for symmetric polynomials up to a given order p. For order $p = 5$, the number of required sigma-points is $2n^2 + 1$. The corresponding sigma-point filter is referred to as UKF 5.

Symmetric, 7th order. For order $p = 7$, the number of required sigma-points is $\frac{1}{3}(4n^3 + 8n + 3)$, meaning that they scale cubicly with the number of state dimensions. The corresponding sigma-point filter is referred to as UKF 7.

Symmetric, 9th order. For order $p = 9$, the number of required sigma-points is $\frac{1}{3}(2n^4 - 4n^3 + 22n^2 - 8n + 3)$. The corresponding sigma-point filter is referred to as UKF 9. If required, even higher order methods can be constructed in the spirit of [26].

Gauss-Hermite. The n-dimensional Gauss-Hermite quadrature method forms the sigma-points as a Cartesian product of the one-dimensional Gauss-Hermite quadratures, and the weights are simply products of the one-dimensional weights [6, 16, 43]. The disadvantage of this method is that with a pth order GH approximation (exact for polynomials up to order p), the required number of evaluation points is p^n, the number growing exponentially with state dimension n. The corresponding filter is referred to as GHKF.

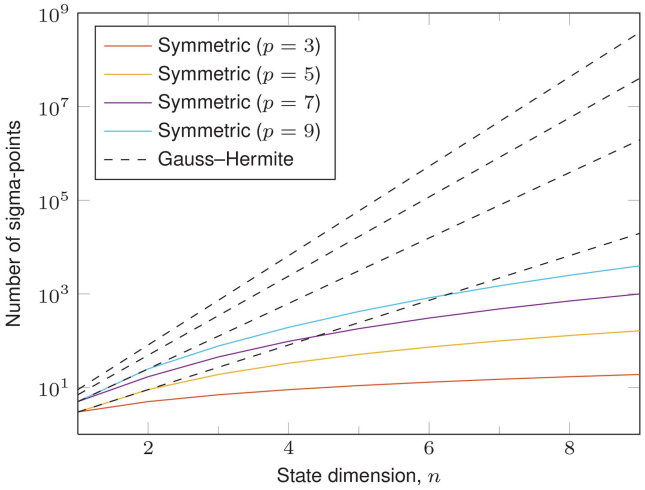


Fig. 2. Scaling of the number of sigma-points for each of the symmetric methods (solid lines). The required number of sigma-points for the Gauss-Hermite cubature of corresponding order p is visualized by the dashed lines.

The exact formulas for the higher-order methods become lengthy and have been omitted here for brevity (see, [25, 26, 43], for implementation details and discussion). Figure 1 gives a pictorial example of how the points and weights are placed in two dimensions ($n = 2$) for each of the methods. Note that even though the 5th and 9th order methods do not have negative weights when $n = 2$, they have negative weights in other dimensions.

For higher state dimensions, Figure 2 shows how the number of required points scale in each of the schemes. The exponentially growing number of evaluation points for Gauss-Hermite is apparent in Figure 2. In the UKFs, the number of evaluation points grow polynomially. McNamee and Stenger provide the following bound for the number of evaluation points for the fully symmetric integration formulas of arbitrary degree $p = 2k + 1$ in n-space: $\mathrm{O}((2n)^k/k!)$. Note that while in this paper we focus on the higher-order methods based on McNamee and Stenger, alternative cubature rules have also been suggested (see [7, 18]).

C. Sigma-Point Filtering and Smoothing

The following sigma-point filtering and smoothing equations are obtained by selecting a cubature rule, for example, one of the rules discussed in Section II-B and substituting it in place of the expectations in the Gaussian filtering and smoothing equations (Sec. II-A, Eqs. 5–7).

In the following equations, we denote the lower triangular matrix square-root (Cholesky) factor of a covariance matrix $\mathbf{P}$ by $\mathbf{L}$ so that for example $\mathbf{P}_{k|k-1} = \mathbf{L}_{k|k-1}\mathbf{L}_{k|k-1}^{\mathrm{T}}$. The prediction step is

$$\begin{aligned}
\mathbf{m}_{k|k-1} &= \sum_i w_i \mathbf{f}(\mathbf{m}_{k-1|k-1} + \mathbf{L}_{k-1|k-1}\boldsymbol{\xi}_i), \\
\mathbf{P}_{k|k-1} &= \sum_i \{w_i (\mathbf{f}(\mathbf{m}_{k-1|k-1} + \mathbf{L}_{k-1|k-1}\boldsymbol{\xi}_i) - \mathbf{m}_{k|k-1}) \\
&\quad \times (\mathbf{f}(\mathbf{m}_{k-1|k-1} + \mathbf{L}_{k-1|k-1}\boldsymbol{\xi}_i) - \mathbf{m}_{k|k-1})^{\mathrm{T}}\} + \mathbf{Q}
\end{aligned} \tag{12}$$

and the update step is

$$\begin{aligned}
\boldsymbol{\mu}_k &= \sum_i w_i \mathbf{h}(\mathbf{m}_{k|k-1} + \mathbf{L}_{k|k-1}\boldsymbol{\xi}_i), \\
\mathbf{S}_k &= \sum_i \{w_i (\mathbf{h}(\mathbf{m}_{k|k-1} + \mathbf{L}_{k|k-1}\boldsymbol{\xi}_i) - \boldsymbol{\mu}_k) \\
&\quad \times (\mathbf{h}(\mathbf{m}_{k|k-1} + \mathbf{L}_{k|k-1}\boldsymbol{\xi}_i) - \mu_k)^{\mathrm{T}}\} + \mathbf{R}, \\
\mathbf{C}_k &= \sum_i \{w_i \mathbf{L}_{k|k-1}\boldsymbol{\xi}_i \\
&\quad \times (\mathbf{h}(\mathbf{m}_{k|k-1} + \mathbf{L}_{k|k-1}\boldsymbol{\xi}_i) - \boldsymbol{\mu}_k)^{\mathrm{T}}\}, \\
\mathbf{K}_k &= \mathbf{C}_k \mathbf{S}_k^{-1}, \\
\mathbf{m}_{k|k} &= \mathbf{m}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \boldsymbol{\mu}_k), \\
\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^{\mathrm{T}}.
\end{aligned} \tag{13}$$

The Rauch-Tung-Striebel smoother equations are

$$\begin{aligned}
\mathbf{m}_{k+1|k} &= \sum_i \{w_i \mathbf{f}(\mathbf{m}_{k|k} + \mathbf{L}_{k|k}\boldsymbol{\xi}_i)\}, \\
\mathbf{P}_{k+1|k} &= \sum_i \{w_i (\mathbf{f}(\mathbf{m}_{k|k} + \mathbf{L}_{k|k}\boldsymbol{\xi}_i) - \mathbf{m}_{k+1|k}) \\
&\quad \times (\mathbf{f}(\mathbf{m}_{k|k} + \mathbf{L}_{k|k}\boldsymbol{\xi}_i) - \mathbf{m}_{k+1|k})^{\mathrm{T}}\} + \mathbf{Q}, \\
\mathbf{D}_{k+1} &= \sum_i \{w_i \mathbf{L}_{k|k}\boldsymbol{\xi}_j (\mathbf{f}(\mathbf{m}_{k|k} + \mathbf{L}_{k|k}\boldsymbol{\xi}_i) - \mathbf{m}_{k+1|k})\}, \\
\mathbf{G}_k &= \mathbf{D}_{k+1}[\mathbf{P}_{k+1|k}]^{-1}, \\
\mathbf{m}_{k|T} &= \mathbf{m}_{k|k} + \mathbf{G}_k(\mathbf{m}_{k+1|T} - \mathbf{m}_{k+1|k}), \\
\mathbf{P}_{k|T} &= \mathbf{P}_{k|k} + \mathbf{G}_k(\mathbf{P}_{k+1|T} - \mathbf{P}_{k+1|k})\mathbf{G}_k^{\mathrm{T}}.
\end{aligned} \tag{14}$$

To evaluate expectations with respect to the pairwise smoothing distributions (Eq. 8), the required $2n$-dimensional sigma-points need to be generated separately as they are not used in the smoother pass. The sigma-points used for the pairwise smoothing distributions are of the form

$$\begin{pmatrix} \mathbf{x}_k^{(i)} \\ \mathbf{x}_{k-1}^{(i)} \end{pmatrix} = \begin{pmatrix} \mathbf{m}_{k|T}^{(i)} \\ \mathbf{m}_{k-1|T}^{(i)} \end{pmatrix} + \sqrt{\begin{pmatrix} \mathbf{P}_{k|T} & \mathbf{P}_{k|T}\mathbf{G}_{k-1}^{\mathrm{T}} \\ \mathbf{G}_{k-1}\mathbf{P}_{k|T} & \mathbf{P}_{k-1|T} \end{pmatrix}} \boldsymbol{\xi}_i^{(2n)}, \tag{15}$$

where $\boldsymbol{\xi}_i^{(2n)}$ are the $2n$-dimensional unit sigma-points. Then, expectation a function $\mathbf{f}(\mathbf{x}_k, \mathbf{x}_{k-1})$ is approximated as

$$\mathbb{E}(\mathbf{f}(\mathbf{x}_k, \mathbf{x}_{k-1}) \mid \mathbf{y}_{1:T}) = \sum_i w_i^{(2n)} \mathbf{f}(\mathbf{x}_k^{(i)}, \mathbf{x}_{k-1}^{(i)}), \tag{16}$$

where $w_i^{(2n)}$ are the corresponding weights of the $2n$-dimensional sigma-point scheme.

III. PARAMETER ESTIMATION

In this section, we consider methods for estimating the static parameters $\boldsymbol{\theta}$ of the state-space model (1). All methods discussed target the maximum likelihood solution, that is, aim to maximize $p(\mathbf{y}_{1:T} \mid \boldsymbol{\theta})$, or equivalently the log-likelihood:

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \log p(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}). \tag{17}$$

Since the state variables $\mathbf{x}_{0:T}$ cannot in general be marginalized in closed-form, approximative numeric methods are needed.

Here, we focus on three approaches where sigma-point filtering and smoothing is used to approximate the likelihood. First, we consider a so-called direct-likelihood approach, where the sigma-point algorithm is used to directly approximate the log-likelihood and its gradient, which are then used in numeric optimization algorithms such as conjugate-gradient optimization. Second, the expectation-maximization (EM) algorithm which is based on a lower bound for the log-likelihood and iterating optimization of parameters with respect to the lower bound and updating the lower bound with new parameters. The third approach is a modification of the direct-likelihood optimization where Fisher's identity is used to express the gradient of the log-likelihood using the same lower bound function that appears in the EM algorithm. Note that the third approach is otherwise similar to the first, but since it is based on the EM lower bound, we present the methods in this order. Each of these three approaches may be used in combination with any of the sigma-point rules discussed in Section II-B.

Note that all the algorithms presented in this section are easily extended to maximum a posteriori estimation since maximizing the posterior density is equivalent to maximizing the (unnormalized) log-posterior. That is, the sum of log-likelihood and log-prior:

$$\boldsymbol{\theta}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta}} [\log p(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta})]. \tag{18}$$

Since the log-prior is known, approximations of the unnormalized log-posterior as well as its gradient and

lower bounds are immediately obtained from the corresponding approximations for the log-likelihood.

A. Direct Likelihood Based Parameter Estimation

The marginal log-likelihood function can be formulated as the following sum

$$\mathcal{L}_T(\boldsymbol{\theta}) = \sum_{k=1}^{T} \log p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1}, \boldsymbol{\theta}). \tag{19}$$

Furthermore, the terms of the sum on the right hand side may in principle be evaluated by

$$p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1}, \boldsymbol{\theta}) = \int p(\mathbf{y}_k \mid \mathbf{x}_k, \boldsymbol{\theta}) p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}, \boldsymbol{\theta}) d\mathbf{x}_k, \tag{20}$$

that is, integrating the measurement model $p(\mathbf{y}_k \mid \mathbf{x}_k, \boldsymbol{\theta})$ over the predicted state distribution $p(\mathbf{x}_k \mid \mathbf{y}_{1:k-1}, \boldsymbol{\theta})$ which is computed during the Bayesian filtering recursion. In assumed density Gaussian filtering is used, we get the approximation

$$p(\mathbf{y}_k \mid \mathbf{y}_{1:k-1}, \boldsymbol{\theta}) \approx \mathbb{N}(\mathbf{y}_k \mid \boldsymbol{\mu}_k, \mathbf{S}_k), \tag{21}$$

whence the marginal log-likelihood expression in Equation (19) evaluates to

$$\begin{aligned} \mathcal{L}_T(\boldsymbol{\theta}) = \log p(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}) \approx -\frac{1}{2} \sum_{k=1}^{T} \log |2\pi \mathbf{S}_k| \\ - \frac{1}{2} \sum_{k=1}^{T} (\mathbf{y}_k - \boldsymbol{\mu}_k)^{\mathrm{T}} \mathbf{S}_k^{-1} (\mathbf{y}_k - \boldsymbol{\mu}_k), \end{aligned} \tag{22}$$

where the quantities $\boldsymbol{\mu}_k$ and $\mathbf{S}_k$ are evaluated during the filtering recursion, in the case of sigma-point methods by Equation (13).

To enable use of gradient-based optimization algorithms, we also need a method for evaluating the gradients of the marginal log-likelihood. This is based on the so-called sensitivity equations [14, 32] that are obtained by differentiating the filtering equations. Namely, the gradient of the log-likelihood is obtained by the recursion

$$\begin{aligned} \frac{\partial \mathcal{L}_k(\boldsymbol{\theta})}{\partial \theta_i} = \frac{\partial \mathcal{L}_{k-1}(\boldsymbol{\theta})}{\partial \theta_i} - \frac{1}{2} \mathrm{tr} \left(\mathbf{S}_k^{-1}(\boldsymbol{\theta}) \frac{\partial \mathbf{S}_k(\boldsymbol{\theta})}{\partial \theta_i} \right) \\ - \mathbf{v}_k^{\mathrm{T}}(\boldsymbol{\theta}) \mathbf{S}_k^{-1}(\boldsymbol{\theta}) \frac{\partial \mathbf{v}_k(\boldsymbol{\theta})}{\partial \theta_i} \\ + \frac{1}{2} \mathbf{v}_k^{\mathrm{T}}(\boldsymbol{\theta}) \mathbf{S}_k^{-1}(\boldsymbol{\theta}) \frac{\partial \mathbf{S}_k(\boldsymbol{\theta})}{\partial \theta_i} \mathbf{S}_k^{-1}(\boldsymbol{\theta}) \mathbf{v}_k(\theta), \end{aligned} \tag{23}$$

where $\mathbf{v}_k = \mathbf{y}_k - \boldsymbol{\mu}_k$. The derivatives $\partial \mathbf{S}_k(\boldsymbol{\theta}) / \partial \theta_i$ and $\partial \mathbf{v}_k(\boldsymbol{\theta}) / \partial \theta_i$ are computed along the filtering pass by the equations shown in Figure 3.

B. Expectation-Maximization Based Parameter Estimation

Expectation-maximization (EM), proposed by Dempster *et al.* [9] is an iterative algorithm for finding maximum likelihood parameter estimates in settings with some unobserved variables, such as the state variables $\mathbf{x}$ in the state-space context. The motivation is that the so-called full-data likelihood of the observed and unobserved variables is easier to compute, and a lower bound for the marginal likelihood of the observed variables may be obtained based on expected full-data log-likelihood. In the following, we present the EM algorithm following the formulation by Neal and Hinton [27] and the notation of Schön *et al.* [34].

The EM algorithm is based on the following lower bound of the log-likelihood:

$$\log p(\mathbf{y}_{1:T} \mid \boldsymbol{\theta}) \geq \int q(\mathbf{x}_{0:T}) \log \frac{p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \mid \boldsymbol{\theta})}{q(\mathbf{x}_{0:T})} d\mathbf{x}_{0:T}, \tag{24}$$

where q is an arbitrary probability density over the states $\mathbf{x}_{0:T}$. The idea is to iteratively maximize this lower bound with respect to q (holding $\boldsymbol{\theta}$ fixed) and with respect to $\boldsymbol{\theta}$ (holding q fixed). Furthermore, when $\boldsymbol{\theta} = \boldsymbol{\theta}^{(n)}$ is fixed, the maximum with respect to q is obtained by

$$q(\mathbf{x}_{0:T}) := p(\mathbf{x}_{0:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)}). \tag{25}$$

By substituting this into Equation (24), the bound becomes

$$\begin{aligned} \int p(\mathbf{x}_{0:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)}) \log \frac{p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \mid \boldsymbol{\theta})}{p(\mathbf{x}_{0:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)})} d\mathbf{x}_{0:T} \\ = \int p(\mathbf{x}_{0:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)}) \log p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \mid \boldsymbol{\theta}) d\mathbf{x}_{0:T} \\ - \int p(\mathbf{x}_{0:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)}) \log p(\mathbf{x}_{0:T} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)}) d\mathbf{x}_{0:T}. \end{aligned}$$

The latter term is independent of $\boldsymbol{\theta}$ and may thus be omitted when maximizing the lower bound with respect to $\boldsymbol{\theta}$. The first term is the conditional expectation of $\log p(\mathbf{y}_{1:T}, \mathbf{x}_{0:T} \mid \boldsymbol{\theta})$ conditional on $\boldsymbol{\theta}^{(n)}$ and $\mathbf{y}_{1:T}$. Thus, the step of maximizing the lower bound (Eq. 24) may be replaced by computing the following function:

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) = \mathbb{E}[\log p(\mathbf{x}_{0:T}, \mathbf{y}_{1:T} \mid \boldsymbol{\theta}) \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)}]. \tag{26}$$

The EM algorithm in its general form thus consists of initializing the parameters to $\boldsymbol{\theta}^{(0)}$ and for $n = 0, 1, \ldots$ iterating the following two steps:

- E-step: compute $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)})$.
- M-step: $\boldsymbol{\theta}^{(n+1)} \leftarrow \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)})$.

In state-space models, the $\mathcal{Q}$-function can be decomposed by employing the Markov property of the state sequence and the conditional independence of the measurements:

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) = \mathrm{I}_1(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) + \mathrm{I}_2(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) + \mathrm{I}_3(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}), \tag{27}$$

$$
\begin{aligned}
\frac{\partial \mathbf{m}_{k|k-1}}{\partial \theta_i} &= \sum_j \Big\{ w_j \Big[\mathbf{F}_{\mathbf{x}} \left(\mathbf{m}_{k-1|k-1} + \mathbf{L}_{k-1|k-1}\, \boldsymbol{\xi}_j,\ \boldsymbol{\theta} \right) \\
&\quad \times \left(\frac{\partial \mathbf{m}_{k-1|k-1}}{\partial \theta_i} + \frac{\partial \mathbf{L}_{k-1|k-1}}{\partial \theta_i}\, \boldsymbol{\xi}_j \right) \\
&\quad + \frac{\partial \mathbf{f}}{\partial \theta_i} \left(\mathbf{m}_{k-1|k-1} + \mathbf{L}_{k-1}\, \boldsymbol{\xi}_j,\ \boldsymbol{\theta} \right) \Big] \Big\}, \\
\frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} &= \sum_j \Big\{ w_j \Big[\mathbf{F}_{\mathbf{x}} \left(\mathbf{m}_{k-1} + \mathbf{L}_{k-1}\, \boldsymbol{\xi}_j, \boldsymbol{\theta} \right) \\
&\quad \times \left(\frac{\partial \mathbf{m}_{k-1|k-1}}{\partial \theta_i} + \frac{\partial \mathbf{L}_{k-1|k-1}}{\partial \theta_i}\, \boldsymbol{\xi}_j \right) \\
&\quad + \frac{\partial \mathbf{f}}{\partial \theta_i} \left(\mathbf{m}_{k-1|k-1} + \mathbf{L}_{k-1|k-1}\, \boldsymbol{\xi}_j, \boldsymbol{\theta} \right) - \frac{\partial \mathbf{m}_{k|k-1}}{\partial \theta_i} \Big] \\
&\quad \times \left[\mathbf{f} \left(\mathbf{m}_{k-1|k-1} + \mathbf{L}_{k-1|k-1}\, \boldsymbol{\xi}_j, \boldsymbol{\theta} \right) - \mathbf{m}_{k|k-1} \right]^{\mathsf{T}} \\
&\quad + \left[\mathbf{f} \left(\mathbf{m}_{k-1|k-1} + \mathbf{L}_{k-1|k-1}\, \boldsymbol{\xi}_j, \boldsymbol{\theta} \right) - \mathbf{m}_{k|k-1} \right] \\
&\quad \times \Big[\mathbf{F}_{\mathbf{x}} \left(\mathbf{m}_{k-1} + \mathbf{L}_{k-1}\, \boldsymbol{\xi}_j, \boldsymbol{\theta} \right) \\
&\quad \times \left(\frac{\partial \mathbf{m}_{k-1|k-1}}{\partial \theta_i} + \frac{\partial \mathbf{L}_{k-1|k-1}}{\partial \theta_i}\, \boldsymbol{\xi}_j \right) \\
&\quad + \frac{\partial \mathbf{f}}{\partial \theta_i} \left(\mathbf{m}_{k-1|k-1} + \mathbf{L}_{k-1|k-1}\, \boldsymbol{\xi}_j, \boldsymbol{\theta} \right) - \frac{\partial \mathbf{m}_{k|k-1}}{\partial \theta_i} \Big]^{\mathsf{T}} + \frac{\partial \mathbf{Q}}{\partial \theta_i} \Big\}, \\
\frac{\partial \boldsymbol{\mu}_k}{\partial \theta_i} &= \sum_j \Big\{ w_j \Big[\mathbf{H}_{\mathbf{x}} \left(\mathbf{m}_{k|k-1} + \mathbf{L}_{k|k-1}\, \boldsymbol{\xi}_j,\ \boldsymbol{\theta} \right) \\
&\quad + \frac{\partial \mathbf{h}}{\partial \theta_i} \left(\mathbf{m}_{k|k-1} + \mathbf{L}_{k|k-1}\, \boldsymbol{\xi}_j,\ \boldsymbol{\theta} \right) \Big] \Big\}, \\
\frac{\partial \mathbf{v}_k}{\partial \theta_i} &= -\frac{\partial \boldsymbol{\mu}_k}{\partial \theta_i}, \\
\frac{\partial \mathbf{S}_k}{\partial \theta_i} &= \sum_j \Big\{ w_j \Big[\mathbf{H}_{\mathbf{x}} \left(\mathbf{m}_{k|k-1} + \mathbf{L}_{k|k-1}\, \boldsymbol{\xi}_j,\ \boldsymbol{\theta} \right) \\
&\quad \times \left(\frac{\partial \mathbf{m}_{k|k-1}}{\partial \theta_i} + \frac{\partial \mathbf{L}_{k|k-1}}{\partial \theta_i} \right) \\
&\quad + \frac{\partial \mathbf{h}}{\partial \theta_i} \left(\mathbf{m}_{k|k-1} + \mathbf{L}_{k|k-1}\, \boldsymbol{\xi}_j,\ \boldsymbol{\theta} \right) - \frac{\partial \boldsymbol{\mu}_k}{\partial \theta_i} \Big] \\
&\quad \times \Big[\mathbf{h} \left(\mathbf{m}_{k|k-1} + \mathbf{L}_{k|k-1}\, \boldsymbol{\xi}_j,\ \boldsymbol{\theta} \right) - \boldsymbol{\mu}_k \Big]^{\mathsf{T}} \\
&\quad + \Big[\mathbf{h} \left(\mathbf{m}_{k|k-1} + \mathbf{L}_{k|k-1}\, \boldsymbol{\xi}_j,\ \boldsymbol{\theta} \right) - \boldsymbol{\mu}_k \Big] \\
&\quad \times \Big[\mathbf{H}_{\mathbf{x}} \left(\mathbf{m}_{k|k-1} + \mathbf{L}_{k|k-1}\, \boldsymbol{\xi}_j,\ \boldsymbol{\theta} \right) \\
&\quad + \frac{\partial \mathbf{h}}{\partial \theta_i} \left(\mathbf{m}_{k|k-1} + \mathbf{L}_{k|k-1}\, \boldsymbol{\xi}_j \right) - \frac{\partial \boldsymbol{\mu}_k}{\partial \theta_i} \Big]^{\mathsf{T}} \Big\} + \frac{\partial \mathbf{R}}{\partial \theta_i}, \\
\frac{\partial \mathbf{C}_k}{\partial \theta_i} &= \sum_j \Big\{ w_j \Big[\frac{\partial \mathbf{L}_{k|k-1}}{\partial \theta_i}\, \boldsymbol{\xi}_j \Big(\mathbf{h} \left(\mathbf{m}_{k|k-1} + \mathbf{L}_{k|k-1}\, \boldsymbol{\xi}_j, \boldsymbol{\theta} \right) - \mu_k \Big)^{\mathsf{T}} \\
&\quad + \mathbf{L}_{k|k-1}\, \boldsymbol{\xi}_j \Big[\mathbf{H}_{\mathbf{x}} \left(\mathbf{m}_{k|k-1} + \mathbf{L}_{k|k-1}\, \boldsymbol{\xi}_j,\ \boldsymbol{\theta} \right) \\
&\quad \times \left(\frac{\partial \mathbf{m}_{k|k-1}}{\partial \theta_i} + \frac{\partial \mathbf{L}_{k|k-1}}{\partial \theta_i}\, \boldsymbol{\xi}_j \right) \\
&\quad + \frac{\partial \mathbf{h}}{\partial \theta_i} \left(\mathbf{m}_{k|k-1} + \mathbf{L}_{k|k-1}\, \boldsymbol{\xi}_j,\ \boldsymbol{\theta} \right) - \frac{\partial \boldsymbol{\mu}_k}{\partial \theta_i} \Big]^{\mathsf{T}} \Big] \Big\}, \\
\frac{\partial \mathbf{K}_k}{\partial \theta_i} &= \frac{\partial \mathbf{C}_k}{\partial \theta_i}\, \mathbf{S}_k^{-1} - \mathbf{C}_k\, \mathbf{S}_k^{-1}\, \frac{\partial \mathbf{S}_k}{\partial \theta_i}\, \mathbf{S}_k^{-1}, \\
\frac{\partial \mathbf{m}_{k|k}}{\partial \theta_i} &= \frac{\partial \mathbf{m}_{k|k-1}}{\partial \theta_i} + \frac{\partial \mathbf{K}_k}{\partial \theta_i}\, \mathbf{v}_k + \mathbf{K}_k\, \frac{\partial \mathbf{v}_k}{\partial \theta_i}, \\
\frac{\partial \mathbf{P}_{k|k}}{\partial \theta_i} &= \frac{\partial \mathbf{P}_{k|k-1}}{\partial \theta_i} - \frac{\partial \mathbf{K}_k}{\partial \theta_i}\, \mathbf{S}_k\, \mathbf{K}_k^{\mathsf{T}} - \mathbf{K}_k\, \frac{\partial \mathbf{S}_k}{\partial \theta_i}\, \mathbf{K}_k^{\mathsf{T}} - \mathbf{K}_k\, \mathbf{S}_k\, \frac{\partial \mathbf{K}_k^{\mathsf{T}}}{\partial \theta_i}.
\end{aligned}
$$

Fig. 3. Recursion for computing the derivatives of the prediction and update steps. $\mathbf{F}_{\mathbf{x}}$ is the Jacobian of $\mathbf{f}(\mathbf{x},\boldsymbol{\theta})$ as a function of $\mathbf{x}$ and $\mathbf{H}_{\mathbf{x}}$ is the Jacobian of $\mathbf{h}(\mathbf{x},\boldsymbol{\theta})$ as a function of $\mathbf{x}$. Algorithms for computing the derivatives of the Cholesky factors $\mathbf{L}$ such that $\mathbf{P} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$ are omitted here (see [32]).

where the terms are

$$\mathrm{I}_1(\boldsymbol{\theta},\boldsymbol{\theta}^{(n)}) = \mathbb{E}[\log p(\mathbf{x}_0 \mid \boldsymbol{\theta}) \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)}], \tag{28}$$

$$\mathrm{I}_2(\boldsymbol{\theta},\boldsymbol{\theta}^{(n)}) = \sum_{k=1}^{T} \mathbb{E}[\log p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \boldsymbol{\theta}) \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)}], \tag{29}$$

$$\mathrm{I}_3(\boldsymbol{\theta},\boldsymbol{\theta}^{(n)}) = \sum_{k=1}^{T} \mathbb{E}[\log p(\mathbf{y}_k \mid \mathbf{x}_k, \boldsymbol{\theta}) \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)}]. \tag{30}$$

To evaluate this expression, one needs the smoothing distributions $p(\mathbf{x}_t \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)})$ and the joint smoothing distributions of consecutive states $p(\mathbf{x}_k, \mathbf{x}_{k+1} \mid \mathbf{y}_{1:T}, \boldsymbol{\theta}^{(n)})$. Sigma-point approximations to the EM algorithm are then obtained by replacing the expectations over the smoothing distributions by their sigma-point smoother approximations. The Gaussian smoother approximation for $\mathcal{Q}$ is

$$
\begin{aligned}
&\mathcal{Q}(\boldsymbol{\theta},\boldsymbol{\theta}^{(n)}) \approx \\
&\quad - \frac{1}{2}\log|2\pi\mathbf{P}_0| - \frac{T}{2}\log|2\pi\mathbf{Q}| - \frac{T}{2}\log|2\pi\mathbf{R}| \\
&\quad - \frac{1}{2}\mathrm{tr}\{\mathbf{P}_0^{-1}[\mathbf{P}_{0|T} + (\mathbf{m}_{0|T} - \mathbf{m}_0)(\mathbf{m}_{0|T} - \mathbf{m}_0)^{\mathrm{T}}]\} \\
&\quad - \frac{1}{2}\sum_{k=1}^{T} \mathrm{tr}\{\mathbf{Q}^{-1}\mathbb{E}[(\mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1}))(\mathbf{x}_k - \mathbf{f}(\mathbf{x}_{k-1}))^{\mathrm{T}} \mid \mathbf{y}_{1:T}]\} \\
&\quad - \frac{1}{2}\sum_{k=1}^{T} \mathrm{tr}\{\mathbf{R}^{-1}\mathbb{E}[(\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k))(\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k))^{\mathrm{T}} \mid \mathbf{y}_{1:T}]\},
\end{aligned}
\tag{31}
$$

where $\mathbf{P}_0, \mathbf{Q}, \mathbf{R}$ and the model functions $\mathbf{f}(\cdot), \mathbf{h}(\cdot)$ depend on the parameters $\boldsymbol{\theta}$. The smoothing distribution means and covariances $\mathbf{m}_{k|T}, \mathbf{P}_{k|T}$ are obtained during the smoothing backward pass. The expectations over the smoothing distribution in the latter two terms are evaluated by using the sigma-point approximations for Gaussian integrals as follows. The second expectation depends only on the smoothing distribution $\mathbb{N}(\mathbf{x}_k \mid \mathbf{m}_{k|T}, \mathbf{P}_{k|T})$ and is computed as follows:

$$
\begin{aligned}
&\mathbb{E}[(\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k)(\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k)^{\mathrm{T}} \mid \mathbf{y}_{1:T}] \\
&\qquad \approx \sum_i w_i (\mathbf{y}_k - \mathbf{h}(\mathbf{m}_{k|T}) + \mathbf{L}_{k|T}\boldsymbol{\xi}_i) \\
&\qquad\quad \times (\mathbf{y}_k - \mathbf{h}(\mathbf{m}_{k|T}) + \mathbf{L}_{k|T}\boldsymbol{\xi}_i)^{\mathrm{T}}.
\end{aligned}
\tag{32}
$$

The first expectation depends on the pairwise joint smoothing distribution $p(\mathbf{x}_k, \mathbf{x}_{k-1} \mid \mathbf{y}_{1:T})$ (cf. Sec. II-A, Eq. 8). Thus, to evaluate it we need to use $2n$-dimensional sigma-points as discussed in Section II-C, Equation 15.

In general, maximizing $\mathcal{Q}$ in the M-step requires the use numerical optimization, for example, using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [11]. However, using numerical optimization inside EM is quite cumbersome, because with the same effort we

could numerically optimize the approximate likelihood directly. Hence the benefit of EM is in the situation when the optimization can be performed in closed form. This kind of special case is the class of models where the parameters appear linearly although the model itself might be nonlinear.

In the following, we present closed-form solutions for the special case where the model functions are linear combinations of the parameters where the parameters appear as coefficients of the linear combinations and/or the covariances. That is, we consider models that can be represented as follows:

$$\mathbf{x}_k = \mathbf{A}\tilde{\mathbf{f}}(\mathbf{x}_{k-1}) + \mathbf{q}_k, \tag{33}$$

$$\mathbf{y}_k = \mathbf{H}\tilde{\mathbf{h}}(\mathbf{x}_k) + \mathbf{r}_k, \tag{34}$$

where $\tilde{\mathbf{f}}(\cdot)$ and $\tilde{\mathbf{h}}(\cdot)$ are functions containing the nonlinearities and the parameters are a subset of $\{\mathbf{A}, \mathbf{H}, \mathbf{Q}, \mathbf{R}, \mathbf{m}_0, \mathbf{P}_0\}$.

For these models, the expression for $\mathcal{Q}$ can be written as

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) = \\ &-\frac{1}{2}\log|2\pi\mathbf{P}_0| - \frac{T}{2}\log|2\pi\mathbf{Q}| - \frac{T}{2}\log|2\pi\mathbf{R}| \\ &-\frac{1}{2}\mathrm{tr}\{\mathbf{P}_0^{-1}[\mathbf{P}_{0|T} + (\mathbf{m}_{0|T} - \mathbf{m}_0)(\mathbf{m}_{0|T} - \mathbf{m}_0)^{\mathrm{T}}]\} \\ &-\frac{T}{2}\mathrm{tr}\{\mathbf{Q}^{-1}[\boldsymbol{\Sigma} - \mathbf{C}\mathbf{A}^{\mathrm{T}} - \mathbf{A}\mathbf{C}^{\mathrm{T}} + \mathbf{A}\boldsymbol{\Phi}\mathbf{A}^{\mathrm{T}}]\} \\ &-\frac{T}{2}\mathrm{tr}\{\mathbf{R}^{-1}[\mathbf{D} - \mathbf{B}\mathbf{H}^{\mathrm{T}} - \mathbf{H}\mathbf{B}^{\mathrm{T}} + \mathbf{H}\boldsymbol{\Theta}\mathbf{H}^{\mathrm{T}}]\},\end{aligned}$$

where the model parameters to be optimized are some subset of $\{\mathbf{A}, \mathbf{H}, \mathbf{Q}, \mathbf{R}\mathbf{m}_0, \mathbf{P}_0\}$ and $\boldsymbol{\Sigma}, \boldsymbol{\Phi}, \boldsymbol{\Theta}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ can be evaluated based on the latest E-step sigma-point smoother results as follows:

$$\boldsymbol{\Sigma} = \frac{1}{T}\sum_{k=1}^{T} \mathbf{P}_{k|T} + \mathbf{m}_{k|T}[\mathbf{m}_{k|T}]^{\mathrm{T}}, \tag{35}$$

$$\boldsymbol{\Phi} = \frac{1}{T}\sum_{k=1}^{T} \mathbb{E}[\tilde{\mathbf{f}}(\mathbf{x}_{k-1})\tilde{\mathbf{f}}^{\mathrm{T}}(\mathbf{x}_{k-1}) \mid \mathbf{y}_{1:T}], \tag{36}$$

$$\boldsymbol{\Theta} = \frac{1}{T}\sum_{k=1}^{T} \mathbb{E}[\tilde{\mathbf{h}}(\mathbf{x}_k)\tilde{\mathbf{h}}^{\mathrm{T}}(\mathbf{x}_k) \mid \mathbf{y}_{1:T}], \tag{37}$$

$$\mathbf{B} = \frac{1}{T}\sum_{k=1}^{T} \mathbf{y}_k\mathbb{E}[\tilde{\mathbf{h}}^{\mathrm{T}}(\mathbf{x}_k) \mid \mathbf{y}_{1:T}], \tag{38}$$

$$\mathbf{C} = \frac{1}{T}\sum_{k=1}^{T} \mathbb{E}[\mathbf{x}_k\tilde{\mathbf{f}}^{\mathrm{T}}(\mathbf{x}_{k-1}) \mid \mathbf{y}_{1:T}], \tag{39}$$

$$\mathbf{D} = \frac{1}{T}\sum_{k=1}^{T} \mathbf{y}_k\mathbf{y}_k^{\mathrm{T}}. \tag{40}$$

Using these values, the optimal parameters in the M-step, that is, the maximum points of the $\mathcal{Q}(\cdot, \boldsymbol{\theta}^{(n)})$-function are

- When $\boldsymbol{\theta} = \mathbf{A}$, we get
$$\mathbf{A}^* = \mathbf{C}\boldsymbol{\Phi}^{-1}.$$
- When $\boldsymbol{\theta} = \mathbf{H}$, we get
$$\mathbf{H}^* = \mathbf{B}\boldsymbol{\Theta}^{-1}.$$
- When $\boldsymbol{\theta} = \mathbf{Q}$, we get
$$\mathbf{Q}^* = \boldsymbol{\Sigma} - \mathbf{C}\mathbf{A}^{\mathrm{T}} - \mathbf{A}\mathbf{C}^{\mathrm{T}} + \mathbf{A}\boldsymbol{\Phi}\mathbf{A}^{\mathrm{T}}.$$
- When $\boldsymbol{\theta} = \mathbf{R}$, we get
$$\mathbf{R}^* = \mathbf{D} - \mathbf{H}\mathbf{B}^{\mathrm{T}} - \mathbf{B}\mathbf{H}^{\mathrm{T}} + \mathbf{H}\boldsymbol{\Theta}\mathbf{H}^{\mathrm{T}}.$$
- When $\boldsymbol{\theta} = \mathbf{m}_0$, we get
$$\mathbf{m}_0^* = \mathbf{m}_{0|T}. \tag{41}$$
- Finally, the maximum with respect to the initial covariance $\boldsymbol{\theta} = \mathbf{P}_0$ is
$$\mathbf{P}_0^* = \mathbf{P}_{0|T} + (\mathbf{m}_{0|T} - \mathbf{m}_0)(\mathbf{m}_{0|T} - \mathbf{m}_0)^{\mathrm{T}}.$$

C. Evaluating the Gradient Based on Fisher's Identity

The expected log-likelihood that appears in the EM algorithm may also be used as a basis of an alternative approach for evaluating the gradient in direct optimization. Based on Fisher's identity, the gradient of the marginal log-likelihood may be expressed as

$$\frac{\partial \mathcal{L}_T(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \left.\frac{\partial \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}^{(n)} = \boldsymbol{\theta}}, \tag{42}$$

where $\mathcal{Q}$ is the function defined in the EM algorithm (Eq. 27). When the $\mathcal{Q}$-function is approximated with sigma-point smoothers, we obtain an alternative approximation of the gradient of the marginal log-likelihood that may be used in place of the approximation derived in Section III-A. For linear state-space models, this approach was suggested by Segal and Weinstein [35] and later by Olsson *et al.* [28] who called the approach the 'easy gradient recipe.' See [5, 32] for discussions of the nonlinear case.

IV. EXPERIMENTS

In this section, we demonstrate the different sigma-point schemes and different parameter estimation algorithms with two example models. First, we use a one-dimensional model (the univariate nonstationary growth model, UNGM, [1, 23]) to illustrate the approximate likelihood curves obtained by different methods. Second, we compare the performance of different algorithms with simulated data in a problem of tracking a maneuvering target with bearings-only measurements. In this example, we focus on estimating the sensor variances and compare the variance estimates as well as the actual tracking error.

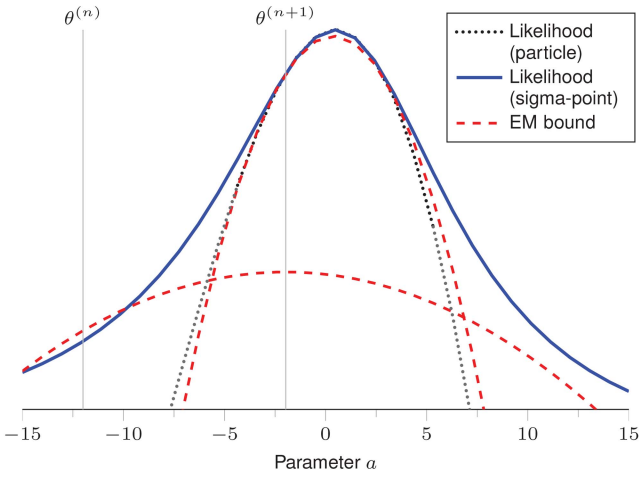


Fig. 4. Visualization of the one-step evolution of the EM algorithm for the univariate estimation of parameter a. The dotted line represents the particle filter log-likelihood estimate (extrapolated in the tails), while the solid line is the sigma-point filter log-likelihood approximation. The dashed lines correspond to the sigma-point EM bounds for iterations n and $n+1$.

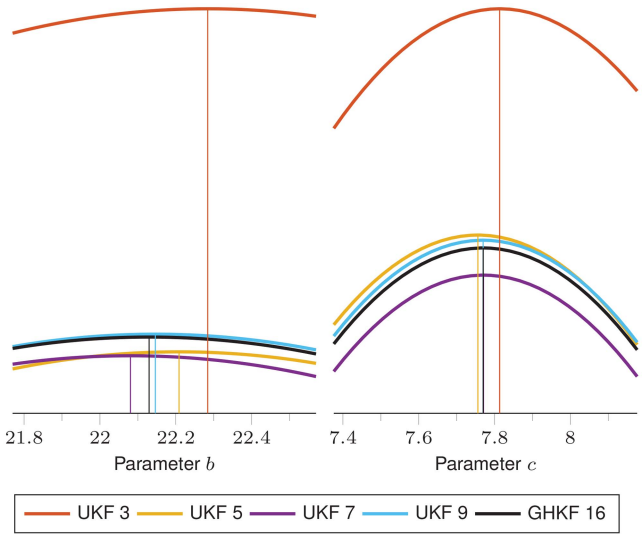


Fig. 5. Log-likelihood curves for parameters b and c evaluated by five different sigma-point methods. The vertical line indicates the location of the maximum.

Codes for replicating the experiments are available on the author homepages.

A. Simple Nonlinear Growth Model

We simulated a realization with $T = 100$ data from the following model:

$$x_{k+1} = ax_k + b\frac{x_k}{1+x_k^2} + c\cos(1.2k) + q_k, \qquad (43)$$

$$y_k = dx_k + r_k. \qquad (44)$$

with $a = 0.5$, $b = 25$, $c = 8$, $d = \sqrt{0.05}$, $q_k \sim \mathbb{N}(0,10)$, $r_k \sim \mathbb{N}(0,0.01)$, $x_0 \sim \mathbb{N}(0,0.01)$. This is the univariate nonstationary growth model [1, 23] except that we changed the measurement model to linear as the model with the typically used quadratic measurement model is known to be challenging for sigma-point algorithms [42].

First, we estimated the likelihood of parameter a, holding other parameters fixed at their ground-truth values. Likelihood curves obtained by direct likelihood estimation with various sigma-point rules as well as the EM lower bounds for two iterations are shown in Figure 4. For comparison, a likelihood estimate obtained by particle filtering (1000 particles and the optimal importance distribution) is also shown. The EM iterations seem to converge toward the maximum of the likelihood curve and the second EM bound is rather close to the particle filter likelihood estimate. The EM lower bounds are mostly below the sigma-point likelihood curve as expected, except that the first EM lower bound slightly exceeds the sigma-point likelihood approximation in the vicinity of the initial parameter. However, both the evaluation of $\mathcal{Q}$ and the sigma-point estimate of the likelihood are approximations.

Second, to compare the different sigma-point rules, we considered estimation of the parameter b with other parameters fixed and parameter c with other parameters fixed, using a grid of parameter values with close proximity to the maximum likelihood values. Namely, for b we used 32 evenly spaced points between 21.7698 and 22.5698 and for c we used 32 points between 7.376 and 8.176. These are shown in Figure 5. The estimate obtained by the Gauss-Hermite rule and the estimates obtained by the higher-order UKFs are rather close to each other while the estimate by the 3rd order UKF is farther in both parameters.

B. Coordinated-Turn Model

In this section, we compare the performance of the parameter estimation methods discussed in this article using a more practical example. The problem is tracking a target maneuvering according to the coordinated turn model [2, 4, 33, 38] with bearings-only sensor measurements. The state is 5-dimensional:

$$\mathbf{x} = (x_1 \quad x_2 \quad \dot{x}_1 \quad \dot{x}_2 \quad \omega)^{\mathrm{T}}, \qquad (45)$$

where (x_1, x_2) is the location of the target in 2-dimensional Cartesian coordinates, $(\dot{x}_1, \dot{x}_2)$ is the corresponding speed, and ω is the turn rate. The dynamic model is

$$\mathbf{x}_{k+1} = \begin{pmatrix} 1 & 0 & \frac{\sin(\omega_k \Delta t)}{\omega_k} & \frac{\cos(\omega_k \Delta t) - 1}{\omega_k} & 0 \\ 0 & 1 & -\frac{\cos(\omega_k \Delta t) - 1}{\omega_k} & \frac{\sin(\omega_k \Delta t)}{\omega_k} & 0 \\ 0 & 0 & \cos(\omega_k \Delta t) & -\sin(\omega_k \Delta t) & 0 \\ 0 & 0 & \sin(\omega_k \Delta t) & \cos(\omega_k \Delta t) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{x}_k + \mathbf{q}_k. \qquad (46)$$

The process noise is $\mathbf{q}_k \sim \mathbb{N}(\mathbf{0}, \mathbf{Q})$, where

$$\mathbf{Q} = \begin{pmatrix} q_c\Delta t^3/3 & 0 & q_c\Delta t^2/2 & 0 & 0 \\ 0 & q_c\Delta t^3/3 & 0 & q_c\Delta t^2/2 & 0 \\ q_c\Delta t^2/2 & 0 & q_c\Delta t & 0 & 0 \\ 0 & q_c\Delta t^2/2 & 0 & q_c\Delta t & 0 \\ 0 & 0 & 0 & 0 & q_\omega\Delta t \end{pmatrix}. \tag{47}$$

The measurements are angles from the sensors with additive Gaussian noise:

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{r}_k, \tag{48}$$

where the measurement noise is $\mathbf{r}_k \sim \mathbb{N}(\mathbf{0}, \mathbf{R})$. The covariance matrix $\mathbf{R}$ is naturally assumed diagonal, as the measurement errors of separate sensors should be independent. For each sensor i at location $\mathbf{s}_i$ the measurement is given by

$$h_i(\mathbf{x}_k) = \text{atan2}(x_{2,k} - s_{2,i}, x_{1,k} - s_{1,i}), \tag{49}$$

where atan2 is the four-quadrant inverse tangent. We focus on estimating the measurement noise variances while keeping other parameters fixed. That is, the sensor locations and dynamic model covariance are assumed to be known and the initial state distribution fixed.

The parameters of the process noise covariance were set to $q_c = 0.1$, $q_\omega = 0.1$ and the time step to $\Delta t = 0.01$. The ground-truth measurement noise covariance was $\mathbf{R} = \text{diag}(0.05^2, 0.1^2)$. The two sensors were located at $\mathbf{s}_1 = (-1, 0.5)$ and $\mathbf{s}_2 = (1, 1)$. The parameters of the initial distribution were $\mathbf{m}_0 = (2, 0, 0, 0, 0)^\mathrm{T}$ and $\mathbf{P}_0 = \text{diag}(0.5^2\ 0.5^2,\ 0.5^2,\ 0.5^2,\ 1^2)$. We simulated 100 different trajectories with $T = 50$ timesteps from this model.

To compare performance of the different sigma-point schemes, we performed direct maximum likelihood estimation of the sensor noise standard deviation of the first sensor, keeping the noise of the second sensor as well as other parameters fixed at their ground truth values. The sigma-point schemes used were UKF 3, UKF 5, and UKF 7, as well as GHKF 3, GHKF 5, and GHKF 7. The 9th order schemes were omitted since the number of sigma-points is already quite high as the state is 5-dimensional. In addition to the sigma-point methods, we also compared against maximum likelihood estimation based on the extended Kalman filter (EKF, see, e.g. [17]).

The optimization was performed with gradient-based optimization using the Matlab optimization toolbox.[1] Furthermore, we investigated how the estimation performance varies as a function of uncertainty of the target's initial location. This was done by using an additional parameter for the per-coordinate standard deviation ($\sigma \in (0, 0.5]$). The first two diagonal components of $\mathbf{P}_0$ were set to σ^2. Furthermore, the first two components of $\mathbf{m}_0$ were interpolated between the original $\mathbf{m}_0$ and the simulated $\mathbf{x}_0$ to keep the uncertainty of the initial location consistent with the prior.

[1] MATLAB version R2014b, the `fminunc` function, quasi-Newton algorithm, initialized with $\sqrt{\mathbf{R}_{1,1}} = 0.1$.

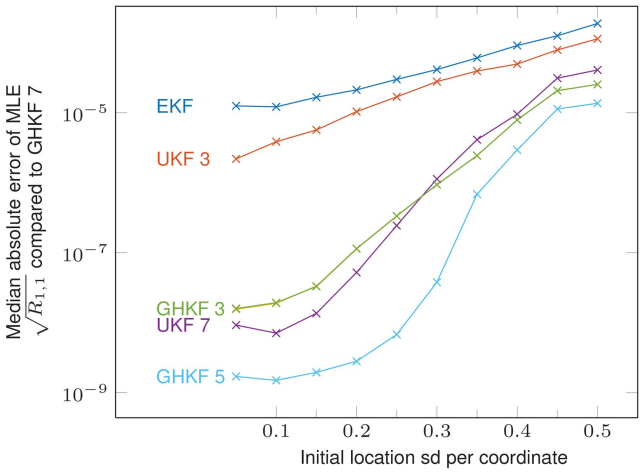


Fig. 6. Median absolute error of the parameter estimates compared to GHKF 7 (median taken over the 100 simulated trajectories) as a function of the initial location prior standard deviation. UKF 5 is essentially indistinguishable from GHKF 3.

Since GHKF 7 is the highest-order sigma-point scheme amongst those used in this experiment, we assume it is the most accurate and compare against it. Figure 6 shows comparison of median (over the 100 trajectories) absolute deviation of the MLE estimates obtained by the various filtering schemes compared to the ones obtained by GHKF 7. GHKF 5 is closest, while EKF and UKF 3 are farthest from the baseline. UKF 7, GHKF 3 and UKF 5 have similar performance. UKF 5 and GHKF 3 are essentially identical. This is explained by the observation that in 5 dimensions, all UKF 5 sigma-points are present in the GHKF 3 sigma-point set and the sum of the GHKF 3 weights of these points is 0.79. The contribution of the remaining sigma-points that have total 0.21 weight apparently has a negligible contribution at least with this model. In addition, we also look at track estimation errors using the final parameter estimates by each sigma-point scheme. Figure 7 shows the mean RMSE over the 100 trajectories, that is, for each simulated trajectory, we computed the smoother RMSE and then took the average.

To compare the two different gradient evaluation approaches, sensitivity equations (Section III-A) and the Fisher identity approach (Section III-C), we evaluated the derivative of the log-likelihood with respect to the standard deviation of the error of the first sensor, using UKF 3 and UKF 5 and both gradient evaluation approaches. The results are shown in Figure 8.

To measure the performance as a function of computational cost, we recorded the parameter values as well as the times used at each iteration of the optimization routines. In this experiment, the initial location standard deviation per coordinate was set to 0.5. Median

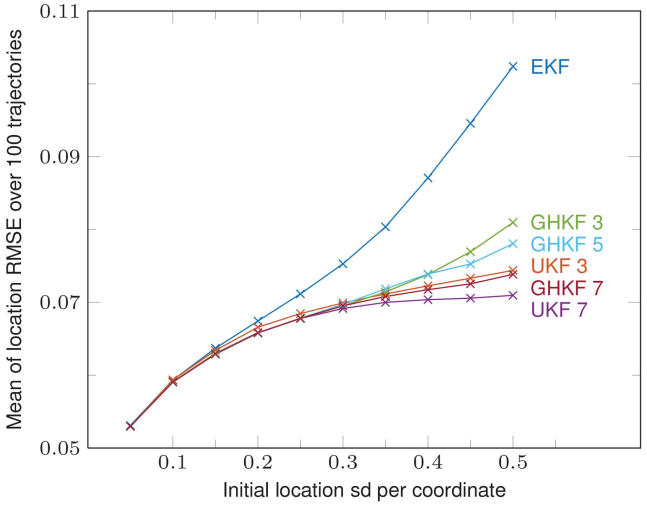


Fig. 7. Mean RMSE of smoothed location (mean taken over the 100 trajectories) using the MLE estimated noise variance of the first sensor. UKF 5 is essentially indistinguishable from GHKF 3.

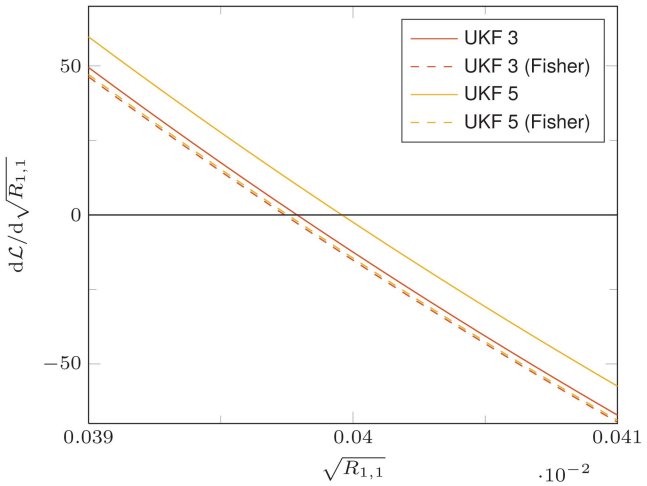


Fig. 8. Derivative of the log-likelihood with respect to the standard deviation of the error of the first sensor. Evaluated using UKF 3 and UKF 5 both with the sensitivity equation approach (Section III-A) and the Fisher identity based approach (Section III-C).

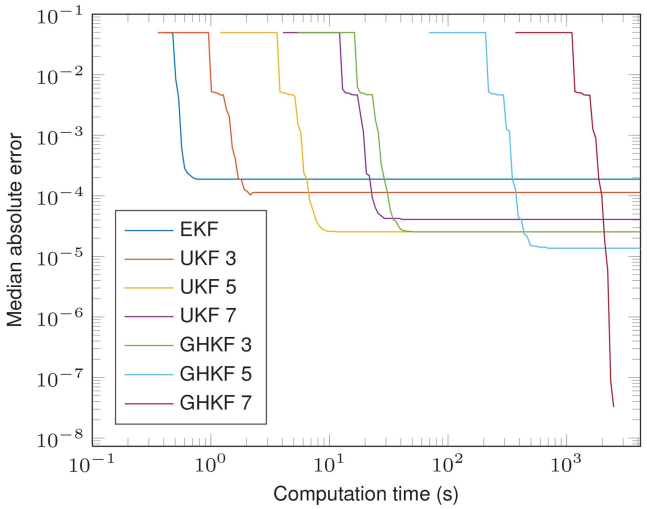


Fig. 9. Median absolute error of the parameter as a function of computation time during the optimization. Median taken over 100 datasets.

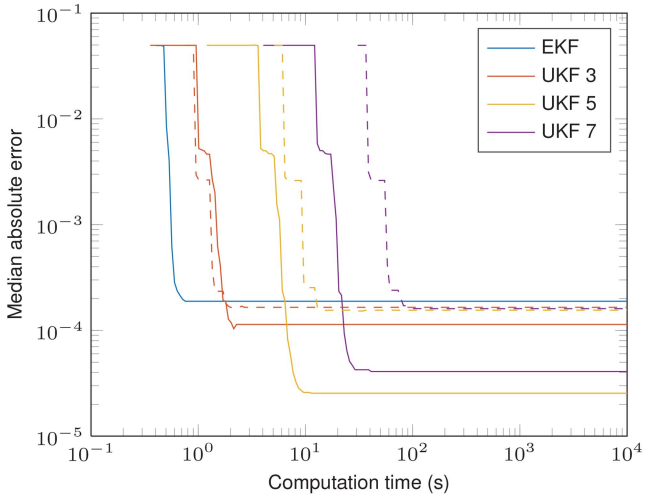


Fig. 10. Median absolute error of the parameter as a function of computation time during the optimization. Median taken over 100 datasets. The solid lines are gradient-based direct optimization, while the dashed lines show EM (run for 32 iterations) with the corresponding sigma-point schemes.

absolute error (compared to final GHKF 7 estimate, as a function of time) is shown in Figures 9 and 10. As one would expect, the higher-order schemes are more computationally demanding and GHKF is more computationally demanding than UKF, but eventually the higher-order GHKF schemes find better parameter estimates.

Figure 11 shows the evolution of EM parameter estimation for one simulated trajectory with $\sigma = 0.5$. The EM algorithm practically converges in a couple of steps with all three sigma-point schemes, and the final parameter estimates are rather close to each other and to the direct MLE estimates. Theoretically, the EM algorithm has linear convergence [9] although it is hard to say whether these convergence results extend to the case where the E-step is approximated using sigma-point smoothers.

V. CONCLUSION AND DISCUSSION

In this paper together with the complementing conference article [24], we have considered various probabilistic point estimation approaches for parameter estimation in nonlinear system identification. We discussed direct likelihood maximization as well as the expectation-maximization (EM) algorithm coupled with various filtering and smoothing algorithms, namely, sigma-point filters, particle filters, and extended Kalman filters as well as the corresponding smoothers. In this paper, we focused on the differences between different sigma-point filters based on unscented transforms of third, fifth, seventh and ninth orders, and the Gauss-Hermite cubature rules.

In diminishing order of computational complexity and theoretical exactness, the filtering methods would

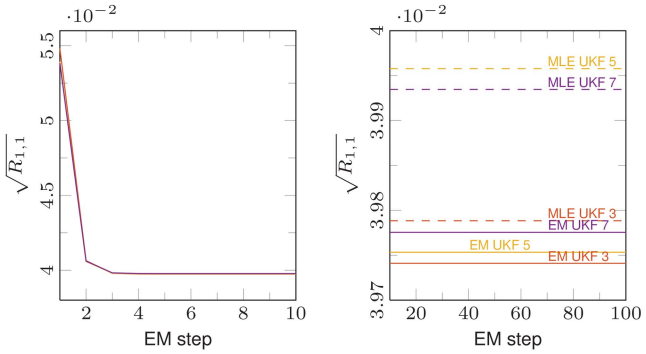


Fig. 11. The plot on the left shows the evolution of the parameter estimate during the first 10 EM steps using UKF 3, UKF 5, and UKF 7. All methods converge essentially at same speed to the same value. The plot on the right shows the evolution of the parameter estimates after the first 10 EM steps as well as the corresponding direct MLE estimates (the dashed lines).

rank as follows: particle filter, sigma-point filter, extended Kalman filter based direct likelihood approximation. In theory, particle filters converge to the exact filtering solution as the number of particles increases, while the other methods considered are based on assuming a Gaussian density and using the Kalman filter equations. However, especially in high-dimensional cases, the computational cost may prohibit the use of particle filters. In practice, the assumed density Gaussian filtering approach may have satisfactory performance if the nonlinearity is not too high. In principle, all sigma-point filters are based on assuming Gaussian density, and a higher-order cubature rule should lead to more accurate approximation of the Gaussian integrals at the cost of higher computational burden. However, typically in the literature (e.g. [2]) it has been claimed that when the Gaussian density approximation is already inaccurate, more accurate computation of the Gaussian integrals is not beneficial.

We also tested the methods in two simulated case studies. In the univariate nonstationary growth model, maximum likelihood estimates produced by different sigma-point schemes were similar. However, the estimates obtained by higher-order unscented schemes were closer to the Gauss-Hermite (order 16) baseline than the conventional 3rd order unscented transform. This suggests that the higher-order methods may indeed have some utility.

In the target tracking experiment, we compared the estimates of the noise standard deviation of one of the sensors as a function of prior uncertainty of the target's location using each of the sigma-point schemes. With higher prior uncertainty, there were more differences amongst the methods. This is reasonable because when there is less uncertainty in the model, all the methods obtain more accurate parameter estimates. Furthermore, the nonlinearity of the model has a stronger effect when the state variance is larger. Since no exact maximum likelihood estimate was available, we compared to the highest-order sigma-point scheme, namely, GHKF 7. Compared to that, the Gauss-Hermite schemes were closer than the unscented transform based schemes and higher-order schemes were closer than lower-order schemes. Thus, the results are consistent with an assumption that the higher-order sigma-point methods produce better approximations to the Gaussian filtering result. However, the magnitude of the differences observed in this experiment is not of practical relevance and the sample size is small. Thus, the experiment may be viewed only as suggestive.

We also measured the performance of the discussed optimization routines as a function of computational time. In direct gradient-based optimization, higher-degree algorithms are more time-consuming but eventually seem to obtain better parameter estimates. The EM algorithm with low-degree sigma-point schemes (UKF 3, UKF 5) was initially faster than the gradient-based optimization, but eventually the gradient-based optimization seems to obtain better values. The EM algorithm with UKF 7 sigma-points was more computationally demanding than the direct optimization with UKF 7. This is due to the fact that EM requires the $2n$-dimensional sigma-points for the smoothing distribution. This suggests that EM is not applicable in high-dimensional problems combined with high-degree sigma-point schemes. However, when interpreting these findings, it should be noted that we compared to the GHKF 7 estimate since the true maximum-likelihood estimate was not available.

The sigma-point integration schemes are derived by assuming exact integration results for polynomials of certain degrees. Thus, it should be noted that it is not guaranteed that a higher-order integration rule produces a more accurate results, even though it is accurate for higher-order polynomials. Furthermore, it is not guaranteed that a better approximation to the Gaussian filtering result produces a better approximation to the exact maximum likelihood result. On the other hand, there is no reason why in general a lower-order approximation to the Gaussian filtering integrals would produce more accurate approximations to the exact filtering results.

We also compared the actual tracking performance in terms of the smoother root mean square errors of the target locations using the smoother results obtained with the maximum likelihood parameter estimate of each sigma-point filter. There was no clear differences between the different sigma-point schemes in terms of the tracking error in this experiment. The tracking error of EKF increased more rapidly as a function of the initial location uncertainty, which demonstrates the local linearization nature of EKF.

In five dimensions, the UKF 5 sigma-point scheme approximates the GHKF 3 scheme in the sense that all UKF 5 sigma-points are GHKF 3 sigma-points as well, and more than half of total weight is contributed by these points. The target-tracking experiment demonstrated that these two schemes indeed produce almost equal results. However, most GHKF 3 sigma-points are

not used in the UKF 5 scheme, and thus evaluating an integral by UKF 5 requires considerably fewer function evaluations. These results suggest that there is no reason to use GHKF 3 in five dimensions as UKF 5 produces essentially same results with fewer computations. It is possible that similar computationally lighter close approximations exist for other Gauss-Hermite based sigma-point schemes as well.

In the target-tracking experiment, we also investigated how the performance of the EM algorithm varies with the sigma-point scheme used. As a function of EM iterations, the evolution of the parameter estimate was not affected by the choice of sigma-point scheme. However, the number of sigma-points in the UKF 3 rule is a small fraction of the number of sigma-points with the higher-order rules. Thus, measured by model function evaluations, EM with the UKF 3 rule converged faster. This suggests that even when the interest lies in obtaining as accurate parameter estimates as possible, a reasonable computational approach would be to first use EM with a low-order sigma-point scheme, such as UKF 3, to obtain a ballpark estimate. Then, if accuracy is desired, the initial estimate could be refined using a more accurate sigma-point scheme combined with a direct optimization algorithm.

In this paper, we considered only discrete-time state-space models. Different sigma-point schemes may also be used for continuous-discrete state-space models (see, e.g., [8] and references therein). We considered only fixed deterministic sigma-point schemes. An interesting future research topic could be to combine the recently proposed filters based on adapting or randomizing the sigma-points [10, 39, 40] with parameter estimation.

Finally, we attempt to conclude which of the methods considered here one should use in practice. Regarding the choice of sigma-point methods the higher-order unscented transform methods turned out to be quite good in the examples that we considered—but if the best possible accuracy is desired, then Gauss-Hermite methods need to be used. The EM algorithm is indeed useful in situations when the M-step optimization can be done in closed form—of which important special cases are the linear-in-parameters models considered here. However, for models that are not linear-in-parameters, EM might not be a good choice. For these models, it is thus beneficial to directly optimize the log-likelihood, and in that case we have the choice to evaluate the gradients either using the sensitivity equations or using the Fisher's identity. It turned out that the Fisher's identity is often computationally more demanding than the sensitivity equations, due to the requirement of a smoothing pass, which favors the use of the sensitivity equations for this purpose. Furthermore, the sensitivity equations give the exact gradients of the approximate likelihood whereas the Fisher's identity only gives approximate gradients of it. However, the Fisher's identity has the advantage of easy black-box implementation which can sometimes be seen as an advantage.

APPENDIX. M-STEP IN THE LINEAR-IN-PARAMETERS CASE

By substituting the linear-in-parameters model ($\mathbf{f}(\mathbf{x}) := \mathbf{A}\tilde{\mathbf{f}}(\mathbf{x})$, $\mathbf{h}(\mathbf{x}) := \mathbf{H}\tilde{\mathbf{h}}(\mathbf{x})$) into the general expression for $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)})$, we obtain

$$\begin{aligned}
&\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) \\
&\quad \approx -\frac{1}{2}\log|2\pi\mathbf{P}_0| - \frac{T}{2}\log|2\pi\mathbf{Q}| - \frac{T}{2}\log|2\pi\mathbf{R}| \\
&\qquad - \frac{1}{2}\mathrm{tr}\{\mathbf{P}_0^{-1}[\mathbf{P}_{0|T} + (\mathbf{m}_{0|T} - \mathbf{m}_0)(\mathbf{m}_{0|T} - \mathbf{m}_0)^{\mathrm{T}}]\} \\
&\qquad - \frac{1}{2}\sum_{k=1}^{T}\mathrm{tr}\{\mathbf{Q}^{-1}\mathbb{E}[(\mathbf{x}_k - \mathbf{A}\tilde{\mathbf{f}}(\mathbf{x}_{k-1})) \\
&\qquad \times (\mathbf{x}_k - \mathbf{A}\tilde{\mathbf{f}}(\mathbf{x}_{k-1}))^{\mathrm{T}} \mid \mathbf{y}_{1:T}]\} \\
&\qquad - \frac{1}{2}\sum_{k=1}^{T}\mathrm{tr}\{\mathbf{R}^{-1}\mathbb{E}[(\mathbf{y}_k - \mathbf{H}\tilde{\mathbf{h}}(\mathbf{x}_k)) \\
&\qquad \times (\mathbf{y}_k - \mathbf{H}\tilde{\mathbf{h}}(\mathbf{x}_k))^{\mathrm{T}} \mid \mathbf{y}_{1:T}]\}. \qquad (50)
\end{aligned}$$

Since trace is linear, the penultimate term can be written as

$$\begin{aligned}
&= -\frac{T}{2}\mathrm{tr}\{\mathbf{Q}^{-1}\frac{1}{T}\sum_{k=1}^{T}\mathbb{E}[(\mathbf{x}_k - \mathbf{A}\tilde{\mathbf{f}}(\mathbf{x}_{k-1})) \\
&\quad \times (\mathbf{x}_k - \mathbf{A}\tilde{\mathbf{f}}(\mathbf{x}_{k-1}))^{\mathrm{T}} \mid \mathbf{y}_{1:T}]\} \\
&= -\frac{T}{2}\mathrm{tr}\{\mathbf{Q}^{-1}\frac{1}{T}\sum_{k=1}^{T}\mathbb{E}[\mathbf{x}_k\mathbf{x}_k^{\mathrm{T}} - \mathbf{x}_k\tilde{\mathbf{f}}(\mathbf{x}_{k-1})^{\mathrm{T}}\mathbf{A}^{\mathrm{T}} \\
&\quad - \mathbf{A}\tilde{\mathbf{f}}(\mathbf{x}_{k-1})\mathbf{x}_k^{\mathrm{T}} + \mathbf{A}\tilde{\mathbf{f}}(\mathbf{x}_{k-1})\tilde{\mathbf{f}}(\mathbf{x}_{k-1})^{T}\mathbf{A}^{\mathrm{T}} \mid \mathbf{y}_{1:T}]\}, \qquad (51)
\end{aligned}$$

which due to linearity of expectation equals

$$\begin{aligned}
&= -\frac{T}{2}\mathrm{tr}\Big\{\mathbf{Q}^{-1}\Big[\frac{1}{T}\sum_{k=1}^{T}\mathbb{E}[\mathbf{x}_k\mathbf{x}_k^{\mathrm{T}} \mid \mathbf{y}_{1:T}] \\
&\quad - \frac{1}{T}\Big(\sum_{k=1}^{T}\mathbb{E}[\mathbf{x}_k\tilde{\mathbf{f}}(\mathbf{x}_{k-1})^{\mathrm{T}} \mid \mathbf{y}_{1:T}]\Big)\mathbf{A}^{\mathrm{T}} \\
&\quad - \mathbf{A}\frac{1}{T}\sum_{k=1}^{T}\mathbb{E}[\tilde{\mathbf{f}}(\mathbf{x}_{k-1})\mathbf{x}^{\mathrm{T}} \mid \mathbf{y}_{1:T}] \\
&\quad + \mathbf{A}\frac{1}{T}\sum_{k=1}^{T}(\mathbb{E}[\tilde{\mathbf{f}}(\mathbf{x}_{k-1})\tilde{\mathbf{f}}(\mathbf{x}_{k-1})^{\mathrm{T}} \mid \mathbf{y}_{1:T}])\mathbf{A}^{\mathrm{T}}\Big]\Big\}. \qquad (52)
\end{aligned}$$

Noting that $\mathbb{E}[\mathbf{x}_k\mathbf{x}_k^{\mathrm{T}} \mid \mathbf{y}_{1:T}] = \mathbf{m}_{k|T}\mathbf{m}_{k|T}^{\mathrm{T}} + \mathbf{P}_{k|T}$ and substituting in the notation introduced in Equations (35–40), we obtain

$$= -\frac{T}{2}\mathrm{tr}\{\mathbf{Q}^{-1}[\boldsymbol{\Sigma} - \mathbf{C}\mathbf{A}^{\mathrm{T}} - \mathbf{A}\mathbf{C}^{\mathrm{T}} + \mathbf{A}\boldsymbol{\Phi}\mathbf{A}^{\mathrm{T}}]\}. \qquad (53)$$

Similar calculation for the last term in Equation (50), noting that $\mathbb{E}[\mathbf{y}_k\mathbf{y}_k^{\mathrm{T}} \mid \mathbf{y}_{1:T}] = \mathbf{y}_k\mathbf{y}_k^{\mathrm{T}}$, gives

$$-\frac{T}{2}\mathrm{tr}\{\mathbf{R}^{-1}[\mathbf{D} - \mathbf{B}\mathbf{H}^{\mathrm{T}} - \mathbf{H}\mathbf{B}^{\mathrm{T}} + \mathbf{H}\boldsymbol{\Theta}\mathbf{H}\mathrm{T}]\}. \quad (54)$$

Substituting Equations (53) and (54) into Equation (50), we get

$$\begin{aligned}\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n)}) = \\ &-\frac{1}{2}\log|2\pi\mathbf{P}_0| - \frac{T}{2}\log|2\pi\mathbf{Q}| - \frac{T}{2}\log|2\pi\mathbf{R}| \\ &-\frac{1}{2}\mathrm{tr}\{\mathbf{P}_0^{-1}[\mathbf{P}_{0|T} + (\mathbf{m}_{0|T} - \mathbf{m}_0)(\mathbf{m}_{0|T} - \mathbf{m}_0)^{\mathrm{T}}]\} \\ &-\frac{T}{2}\mathrm{tr}\{\mathbf{Q}^{-1}[\boldsymbol{\Sigma} - \mathbf{C}\mathbf{A}^{\mathrm{T}} - \mathbf{A}\mathbf{C}^{\mathrm{T}} + \mathbf{A}\boldsymbol{\Phi}\mathbf{A}^{\mathrm{T}}]\} \\ &-\frac{T}{2}\mathrm{tr}\{\mathbf{R}^{-1}[\mathbf{D} - \mathbf{B}\mathbf{H}^{\mathrm{T}} - \mathbf{H}\mathbf{B}^{\mathrm{T}} + \mathbf{H}\boldsymbol{\Theta}\mathbf{H}^{\mathrm{T}}]\}.\end{aligned}$$

To maximize this with respect to the parameters $(\mathbf{m}_0, \mathbf{P}_0, \mathbf{A}, \mathbf{H}, \mathbf{Q}, \mathbf{R})$ we differentiate with respect to parameter in question and set the derivative to 0. For $\mathbf{Q}$:

$$\begin{aligned}\frac{d\mathcal{Q}}{d\mathbf{Q}} &= -\frac{T}{2}\frac{d}{d\mathbf{Q}}\log|2\pi\mathbf{Q}| \\ &\quad -\frac{T}{2}\frac{d}{d\mathbf{Q}}\mathrm{tr}\{\mathbf{Q}^{-1}[\boldsymbol{\Sigma} - \mathbf{C}\mathbf{A}^{\mathrm{T}} - \mathbf{A}\mathbf{C}^{\mathrm{T}} + \mathbf{A}\boldsymbol{\Phi}\mathbf{A}^{\mathrm{T}}]\} \\ &= -\frac{T}{2}\mathbf{Q}^{-1} \\ &\quad + \frac{T}{2}\mathbf{Q}^{-1}[\boldsymbol{\Sigma} - \mathbf{C}\mathbf{A}^{\mathrm{T}} - \mathbf{A}\mathbf{C}^{\mathrm{T}} + \mathbf{A}\boldsymbol{\Phi}\mathbf{A}^{\mathrm{T}}]\mathbf{Q}^{-1}. \quad (55)\end{aligned}$$

Setting the derivative equal to $\mathbf{0}$, we obtain the equation

$$\frac{T}{2}\mathbf{Q}^{-1} = \frac{T}{2}\mathbf{Q}^{-1}[\boldsymbol{\Sigma} - \mathbf{C}\mathbf{A}^{\mathrm{T}} - \mathbf{A}\mathbf{C}^{\mathrm{T}} + \mathbf{A}\boldsymbol{\Phi}\mathbf{A}^{\mathrm{T}}]\mathbf{Q}^{-1}. \quad (56)$$

Multiplying from right by $(2/T)\mathbf{Q}$ and from left by $\mathbf{Q}$ gives

$$\mathbf{Q} = \boldsymbol{\Sigma} - \mathbf{C}\mathbf{A}^{\mathrm{T}} - \mathbf{A}\mathbf{C}^{\mathrm{T}} + \mathbf{A}\boldsymbol{\Phi}\mathbf{A}^{\mathrm{T}}. \quad (57)$$

The derivations for the optimal solutions of $\mathbf{R}$ and $\mathbf{P}_0$ are similar. For $\mathbf{A}$:

$$\begin{aligned}\frac{d\mathcal{Q}}{d\mathbf{A}} &= -\frac{T}{2}\Big[-\frac{d}{d\mathbf{A}}\mathrm{tr}(\mathbf{Q}^{-1}\mathbf{C}\mathbf{A}^{\mathrm{T}}) \\ &\quad -\frac{d}{d\mathbf{A}}\mathrm{tr}(\mathbf{Q}^{-1}\mathbf{A}\mathbf{C}^{T}) + \frac{d}{d\mathbf{A}}\mathrm{tr}(\mathbf{A}\boldsymbol{\Phi}\mathbf{A}^{\mathrm{T}})\Big] \\ &= -\frac{T}{2}\mathbf{Q}^{-1}[2\mathbf{A}\boldsymbol{\Phi} - 2\mathbf{C}] \quad (58)\end{aligned}$$

Since $\mathbf{Q}^{-1}$ is nonsingular, the derivative is zero only if the last factor is zero. If $\boldsymbol{\Phi}$ is invertible, this in turn implies

$$\mathbf{A} = \mathbf{C}\boldsymbol{\Phi}^{-1}. \quad (59)$$

The derivations for the optimal solutions of $\mathbf{H}$ and $\mathbf{m}_0$ are similar.

If the parameter $\boldsymbol{\theta}$ is any subset of $\{\mathbf{A}, \mathbf{H}, \mathbf{Q}, \mathbf{R}, \mathbf{m}_0, \mathbf{P}_0\}$, it can be optimized by these closed-form expressions. First, note that $(\mathbf{A}, \mathbf{Q})$, $(\mathbf{H}, \mathbf{R})$ and $(\mathbf{m}_0, \mathbf{P}_0)$ are independent in the sense that, for example, the optimal $\mathbf{A}$ and $\mathbf{Q}$ do not depend on the other four parameters. Furthermore, the optimal $\mathbf{A}$ does not depend on $\mathbf{Q}$. Thus, $\mathbf{A}$ and $\mathbf{Q}$ can be jointly optimized by first solving the optimal $\mathbf{A}$ and then substituting that into the expression of optimal $\mathbf{Q}$. Similar reasoning works for $(\mathbf{H}, \mathbf{R})$ and $(\mathbf{m}_0, \mathbf{P}_0)$.

ACKNOWLEDGMENTS

This work was supported by grants from the Academy of Finland (266940, 273475) and by the Emil Aaltonen foundation. We acknowledge the computational resources provided by the Aalto Science-IT project.

REFERENCES

[1] M. L. Andrade Netto, L. Gimeno, and M. J. Mendes
A new spline algorithm for non-linear filtering of discrete time systems.
In *A Link Between Science and Application of Automatic Control*, pages 2123–2130, Helsinki, Finland, 1978.

[2] I. Arasaratnam and S. Haykin
Cubature Kalman filters.
IEEE Transactions on Automatic Control, 54(6):1254–1269, 2009.

[3] I. Arasaratnam, S. Haykin, and T. R. Hurd
Cubature Kalman filtering for continuous-discrete systems: Theory and simulations.
IEEE Transactions on Signal Processing, 58(10):4977–4993, 2010.

[4] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation.
Wiley Interscience, 2001.

[5] O. Cappé, E. Moulines, and T. Rydén
Inference in Hidden Markov Models.
Springer, 2005.

[6] R. Cools
Constructing cubature formulae: The science behind the art.
In *Acta Numerica*, volume 6, pages 1–54. 1997.

[7] D. Crouse
Basic tracking using nonlinear 3D monostatic and bistatic measurements.
IEEE Aerospace and Electronic Systems Magazine, 29(8):4–53, 2014.

[8] D. Crouse
Basic tracking using nonlinear continuous-time dynamic models [tutorial].
IEEE Aerospace and Electronic Systems Magazine, 30(2):4–41, 2015.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin
Maximum likelihood from incomplete data via the EM algorithm.
Journal of the Royal Statistical Society: Series B (Statistical Methodology), 39(1):1–38, 1977.

[10] J. Duník, O. Straka, and M. Šimandl
Stochastic integration filter.
IEEE Transactions on Automatic Control, 58(6):1561–1566, 2013.

[11] R. Fletcher
Practical Methods of Optimization.
John Wiley & Sons, 2nd edition, 1987.

[12] M. Gašperin and Đ. Juričić
Application of unscented transformation in nonlinear system identification.
In *Proceedings of the 18th IFAC World Congress*, volume 18, pages 4428–4433, 2011.

[13] A. Gelb, editor
Applied Optimal Estimation.
The MIT Press, 1974.

[14] R. Goodrich and P. E. Caines
Linear system identification from nonstationary cross-sectional data.
IEEE Transactions on Automatic Control, 24(3):403–411, 1979.

[15] N. K. Gupta and R. K. Mehra
Computational aspects of maximum likelihood estimation and reduction in sensitivity function calculations.
IEEE Transactions on Automatic Control, 19(6):774–783, 1974.

[16] K. Ito and K. Xiong
Gaussian filters for nonlinear filtering problems.
IEEE Transactions on Automatic Control, 45(5):910–927, 2000.

[17] A. H. Jazwinski
Stochastic Processes and Filtering Theory.
Academic Press, 1970.

[18] B. Jia, M. Xin, and Y. Cheng
High-degree cubature Kalman filter.
Automatica, 49(2):510–518, 2013.

[19] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte
A new approach for filtering nonlinear systems.
In *Proceedings of the American Control Conference*, volume 3, pages 1628–1632, Seattle, WA, 1995.

[20] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte
A new method for the nonlinear transformation of means and covariances in filters and estimators.
IEEE Transactions on Automatic Control, 45(3):477–482, 2000.

[21] R. E. Kalman
A new approach to linear filtering and prediction problems.
Transactions of the ASME, Journal of Basic Engineering, 82:35–45, March 1960.

[22] G. Kitagawa
Non-Gaussian state-space modeling of nonstationary time series.
Journal of the American Statistical Association, 82(400): 1032–1041, 1987.

[23] G. Kitagawa
Non-Gaussian state-space modeling of nonstationary time series: Rejoinder.
Journal of the American Statistical Association, 82(400): 1060–1063, 1987.

[24] J. Kokkala, A. Solin, and S. Särkkä
Expectation maximization based parameter estimation by sigma-point and particle smoothing.
In *Proceedings of the 17th International Conference on Information Fusion (FUSION)*, Salamanca, Spain, 2014.

[25] U. N. Lerner
Hybrid Bayesian networks for reasoning about complex systems.
PhD thesis, Stanford University, Stanford, CA, 2002.

[26] J. McNamee and F. Stenger
Construction of fully symmetric numerical integration formulas.
Numerische Mathematik, 10(4):327–344, 1967.

[27] R. M. Neal and G. E. Hinton
A view of the EM algorithm that justifies incremental, sparse, and other variants.
In *Learning in Graphical Models*, pages 355–368. Springer, 1998.

[28] R. K. Olsson, K. B. Petersen, and T. Lehn-Schiøler
State-space models: From the EM algorithm to a gradient approach.
Neural Computation, 19(4):1097–1111, 2007.

[29] G. Poyiadjis, A. Doucet, and S. S. Singh
Particle methods for optimal filter derivative: Application to parameter estimation.
In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 925–928, Philadelphia, PA, 2005.

[30] H. E. Rauch, F. Tung, and C. T. Striebel
Maximum likelihood estimates of linear dynamic systems.
AIAA Journal, 3(8):1445–1450, 1965.

[31] S. Roweis and Z. Ghahramani
Learning nonlinear dynamical systems using the expectation-maximization algorithm.
In S. Haykin, editor, *Kalman Filtering and Neural Networks*, chapter 6, pages 175–220. Wiley-Interscience, 2001.

[32] S. Särkkä
Bayesian Filtering and Smoothing, volume 3 of *Institute of Mathematical Statistics Textbooks*.
Cambridge University Press, 2013.

[33] S. Särkkä and J. Hartikainen
On Gaussian optimal smoothing of non-linear state space models.
IEEE Transactions on Automatic Control, 55(8):1938–1941, 2010.

[34] T. B. Schön, A. Wills, and B. Ninness
System identification of nonlinear state-space models.
Automatica, 47(1):39–49, 2011.

[35] M. Segal and E. Weinstein
A new method for evaluating the log-likelihood gradient (score) of linear dynamic systems.
IEEE Transactions on Automatic Control, 33(8):763–766, 1988.

[36] R. H. Shumway and D. S. Stoffer
An approach to time series smoothing and forecasting using the EM algorithm.
Journal of Time Series Analysis, 3(4):253–264, 1982.

[37] H. Singer
Parameter estimation of nonlinear stochastic differential equations: Simulated maximum likelihood versus extended Kalman filter and Itô-Taylor expansion.
Journal of Computational and Graphical Statistics, 11(4): 972–995, 2002.

[38] A. Solin
Cubature integration methods in non-linear Kalman filtering and smoothing.
Bachelor's thesis, Faculty of Information and Natural Sciences, Aalto University, Finland, 2010.

[39] O. Straka, J. Duník, M. Šimandl, and E. Blasch
Randomized unscented transform in state estimation of non-Gaussian systems: Algorithms and performance.
In *Proceedings of the 15th International Conference on Information Fusion (FUSION)*, Singapore, 2012.

[40] O. Straka, J. Duník, M. Šimandl, and E. Blasch
Comparison of adaptive and randomized unscented Kalman filter algorithms.
In *Proceedings of the 17th International Conference on Information Fusion (FUSION)*, Salamanca, Spain, 2014.

[41] V. Väänänen
Gaussian filtering and smoothing based parameter estimation in nonlinear models for sequential data.
Master's thesis, School of Electrical Engineering, Aalto University, Finland, 2012.

[42] Y. Wu, D. Hu, M. Wu, and X. Hu
Unscented Kalman filtering for additive noise case: Augmented vs. non-augmented.
In *Proceedings of the American Control Conference*, volume 6, pages 4051–4055, Portland, OR, 2005.

[43] Y. Wu, D. Hu, M. Wu, and X. Hu
A numerical-integration perspective on Gaussian filters.
IEEE Transactions on Signal Processing, 54(8):2910–2921, 2006.

Juho Kokkala received his M.Sc. (Tech.) degree from Aalto University, Finland, in 2010, majoring in Systems and Operations Research. He is currently a doctoral student at Aalto University and will defend his thesis in 2016. His research interests are Bayesian statistics, especially Bayesian methods for nonlinear state-space models.

Arno Solin received his M.Sc. (Tech.) degree (with distinction) in engineering physics and mathematics from Aalto University, Espoo, Finland, in 2012. He is scheduled to defend his Doctoral of Science (Tech.) degree in computational science at Aalto University in early 2016. He is currently a researcher at the Department of Neuroscience and Biomedical Engineering and the Department of Computer Science at Aalto University. His research interests are in probabilistic methods for spatio-temporal data with applications to machine learning, signal processing, and spatial statistics with applications in tracking and brain imaging.

Simo Särkkä received his M.Sc. degree in engineering physics and mathematics, and D.Sc. degree in electrical and communications engineering from Helsinki University of Technology, Espoo, Finland, in 2000 and 2006, respectively. Currently he is an Associate Professor and Academy Research Fellow with Aalto University, and Technical Advisor and Director of IndoorAtlas Ltd. His research interests are in multi-sensor data processing systems with applications in location sensing, health technology, machine learning, inverse problems, and brain imaging.

On the relation between Gaussian process quadratures and sigma-point methods

SIMO SÄRKKÄ
JOUNI HARTIKAINEN
LENNART SVENSSON
FREDRIK SANDBLOM

This article is concerned with Gaussian process quadratures, which are numerical integration methods based on Gaussian process regression methods, and sigma-point methods, which are used in advanced non-linear Kalman filtering and smoothing algorithms. We show that many sigma-point methods can be interpreted as Gaussian process quadrature based methods with suitably selected covariance functions. We show that this interpretation also extends to more general multivariate Gauss-Hermite integration methods and related spherical cubature rules. Additionally, we discuss different criteria for selecting the sigma-point locations: exactness of the integrals of multivariate polynomials up to a given order, minimum average error, and quasi-random point sets. The performance of the different methods is tested in numerical experiments.

Manuscript received November 10, 2015; released for publication November 26, 2015.

Refereeing of this contribution was handled by Ondrej Straka.

The first author is grateful to the Academy of Finland for financial support.

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

Authors' addresses: S. Särkkä, Aalto University, Rakentajanaukio 2 c, 02150 Espoo, Finland, (e-mail: simo.sarkka@aalto.fi). J. Hartikainen, Rocsole Ltd., 70150 Kuopio, Finland, (e-mail: jouni.hartikainen @rocsole.com). L. Svensson, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden, (e-mail: lennart.svensson@chalmers.se). F. Sandblom, Volvo Group Trucks Technology, SE-405 08 Gothenburg, Sweden, (e-mail: fredrik.sandblom@volvo.com).

1557-6418/16/$17.00

I. INTRODUCTION

Gaussian process quadratures [1]–[6] are methods to numerically compute integrals of the form

$$\mathcal{I}[\mathbf{g}] = \int \mathbf{g}(\mathbf{x}) w(\mathbf{x}) d\mathbf{x}, \tag{1}$$

where $\mathbf{g} : \mathbb{R}^n \mapsto \mathbb{R}^m$ is a non-linear integrand function and $w(\mathbf{x})$ is a given, typically positive, weight function such that $\int w(\mathbf{x}) d\mathbf{x} < \infty$. In Gaussian process quadratures the function $\mathbf{g}(\mathbf{x})$ is approximated with a Gaussian process regressor [7] and the integral is approximated with that of the Gaussian process regressor.

Sigma-point methods [8]–[16] can be seen [17] as methods that approximate the above integrals via

$$\int \mathbf{g}(\mathbf{x}) w(\mathbf{x}) d\mathbf{x} \approx \sum_i W_i \mathbf{g}(\mathbf{x}_i), \tag{2}$$

where W_i are some predefined weights and $\mathbf{x}_i$ are the sigma-points. Typically the sigma-points and weights are selected such that when $\mathbf{g}$ is a multivariate polynomial up to a certain order, the approximation is exact.

A particularly useful class of methods is obtained when the weight function is selected to be a multivariate Gaussian density $w(\mathbf{x}) = \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P})$. In the context of Gaussian process quadratures it then turns out that the integral of the Gaussian process regressor can be computed in closed form provided that the covariance function of the process is chosen to be a squared exponential [7], [18] (i.e., exponentiated quadratic). This kind of quadrature methods is also often referred to as Bayesian or Bayes-Hermite quadratures [2]. They are closely related to Gauss-Hermite quadratures in the sense that as Gaussian quadratures can be seen to form a polynomial approximation to the integrand via point-evaluations, Gaussian process quadratures use a Gaussian process regression approximation instead [1]–[3]. Because Gaussian process regressors can be used to approximate a much larger class of functions than polynomial approximations [7], they can be expected to perform much better also in numerical integration.

The selection of a Gaussian weight function is also particularly useful in non-linear filtering and smoothing, because the equations of non-linear Gaussian (Kalman) filters and smoothers [17], [19]–[22] consist of Gaussian integrals of the above form and linear operations on vectors and matrices. The selection of different weights and sigma-points leads to different brands of approximate filters and smoothers [17]. For example, the Gauss-Hermite quadrature and cubature based filters and smoothers [21]–[25] are based on explicit numerical integration of the Gaussian integrals. The unscented transform based methods as well as other sigma-point methods [8]–[16] can also be retrospectively interpreted to belong to the class of Gaussian numerical integration based methods [23]. Conversely, Gaussian type of quadrature or cubature based methods can also be interpreted to be special cases of sigma-point methods.

Furthermore, the classical Taylor series based methods [26] and Stirling's interpolation based methods [16], [27] can be seen as ways to approximate the integrand such that the Gaussian integral becomes tractable (cf. [17], [28]). The Fourier-Hermite series [29], Hermite polynomial [30] methods are also based on numerical approximation of the integrands. For more recent advances and applications of sigma-point methods in filtering and smoothing, see the articles [31]–[35] and the references therein.

The aim of this article is to provide a Gaussian process quadrature viewpoint to sigma-point methods and multivariate numerical integration methods for non-linear filtering and smoothing. The generalized viewpoint also leads to novel non-linear filtering and smoothing algorithms. We show that many sigma-point filtering and smoothing algorithms such as unscented Kalman filters and smoothers, cubature Kalman filters and smoothers, and Gauss-Hermite Kalman filters and smoothers can be seen as special cases of Gaussian process quadrature based methods with suitably chosen covariance functions. More generally, we show that many classical multivariate Gaussian quadrature methods, including Gauss-Hermite rules [36], and symmetric integration formulas [37] are special cases of the present methodology. We also discuss different criteria for selecting the sigma-point locations: exactness for multivariate polynomials up to a given order, minimum average error, and quasi-random point sets.

The combination of Gaussian process regressors with Bayesian filters has also been previously studied in [38] and [18]. In both of those works the idea is to use training data to form Gaussian process approximations to the dynamic and observation models. In [38], filtering in the resulting Gaussian process state-space model is done using approximate Bayesian filters such as unscented Kalman filters and particle filters, whereas in [18] the non-linear (Kalman) filtering and smoothing equations are computed via closed form formulas. The present approach and point of view is different, because we use Gaussian processes to approximate the integrals (quadratures) appearing in the filtering and smoothing equations. In practical point of view this roughly corresponds to locally retraining the Gaussian process regressor at every step using specifically designed training point locations.

The Gaussian process quadrature methodology used here can be seen to belong to a larger field of probabilistic numerics [39]–[41], where the underlying idea is to interpret numerical methods as instances of probabilistic inference. For example, numerical integration amounts to computing an estimate of the integral of a function given a finite number of function evaluations, whereas differential equation solvers estimate the ODE solution given a sequence of derivative evaluations, and optimization methods use local estimates of the target function to steer their iterations. Although in this article we only use probabilistic numerical integration, it is clear that probabilistic ODE solvers, optimization methods, and other probabilistic numerical methods would be useful in non-linear filtering and smoothing context as well.

This article is an extended version of the conference article [6], where we analyzed the use of Gaussian process quadratures in non-linear filtering and smoothing as well as their connection to the unscented transform and Gauss-Hermite quadratures. In this article, we deepen and sharpen the analysis of those connections and extend our analysis to a more general class of spherically symmetric integration rules. We also analyze different sigma-point selection schemes as well as provide more extensive set of numerical experiments.

II. BACKGROUND

A. Non-Linear Gaussian (Kalman) Filtering and Smoothing

Non-linear Gaussian (Kalman) filters and smoothers [17], [21]–[23] are methods that can be used to approximate the filtering distributions $p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_k)$ and smoothing distributions $p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_T)$ of non-linear state-space models of the form

$$\begin{aligned} \mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{q}_{k-1}, \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k) + \mathbf{r}_k, \end{aligned} \tag{3}$$

where, for $k = 1, 2, \ldots, T$, $\mathbf{x}_k \in \mathbb{R}^n$ are the hidden states, $\mathbf{y}_k \in \mathbb{R}^d$ are the measurements, and $\mathbf{q}_{k-1} \sim \mathrm{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ and $\mathbf{r}_k \sim \mathrm{N}(\mathbf{0}, \mathbf{R}_k)$ are the process and measurements noises, respectively. The non-linear function $\mathbf{f}(\cdot)$ is used to model the dynamics of the system and $\mathbf{h}(\cdot)$ models the mapping from the states to the measurements.

Non-linear Gaussian filters (see, e.g., [17], page 98) are general methods to produce Gaussian approximations to the filtering distributions:

$$p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_k) \approx \mathrm{N}(\mathbf{x}_k \mid \mathbf{m}_k, \mathbf{P}_k), \quad k = 1, 2, \ldots, T. \tag{4}$$

Non-linear Gaussian smoothers (see, e.g., [17], page 154) are the corresponding methods to produce approximations to the smoothing distributions:

$$p(\mathbf{x}_k \mid \mathbf{y}_1, \ldots, \mathbf{y}_T) \approx \mathrm{N}(\mathbf{x}_k \mid \mathbf{m}_k^s, \mathbf{P}_k^s), \quad k = 1, 2, \ldots, T. \tag{5}$$

Both Gaussian filters and smoothers can be easily generalized to state-space models with non-additive noises (see [17]), but here we only consider the additive noise case.

B. Gaussian Integration and Sigma-Point Methods

Sigma-point filtering and smoothing methods can generally be described as methods that approximate the Gaussian integrals in the Gaussian filtering and smoothing equations (and in the Gaussian moment matching transform) as

$$\int \mathbf{g}(\mathbf{x}) \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P}) d\mathbf{x} \approx \sum_i W_i \mathbf{g}(\mathbf{x}_i), \tag{6}$$

where W_i are some predefined weights and $\mathbf{x}_i$ are the sigma-points. Typically, the sigma-point methods use so called *stochastic decoupling* which refers to the idea that we do a change of variables

$$\int \mathbf{g}(\mathbf{x})\mathrm{N}(\mathbf{x} \mid \mathbf{m},\mathbf{P})d\mathbf{x} = \int \underbrace{\mathbf{g}(\mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi})}_{\tilde{\mathbf{g}}(\boldsymbol{\xi})}\mathrm{N}(\boldsymbol{\xi} \mid \mathbf{0},\mathbf{I})d\boldsymbol{\xi}, \quad (7)$$

where $\mathbf{P} = \sqrt{\mathbf{P}}\sqrt{\mathbf{P}}^{\mathrm{T}}$. This implies that we only need to design weights W_i and unit sigma-points $\boldsymbol{\xi}_i$ for integrating against unit Gaussian distributions:

$$\int \tilde{\mathbf{g}}(\boldsymbol{\xi})\mathrm{N}(\boldsymbol{\xi} \mid \mathbf{0},\mathbf{I})d\boldsymbol{\xi} \approx \sum_i W_i \tilde{\mathbf{g}}(\boldsymbol{\xi}_i), \quad (8)$$

thus leading to approximations of the form

$$\int \mathbf{g}(\mathbf{x})\mathrm{N}(\mathbf{x} \mid \mathbf{m},\mathbf{P})d\mathbf{x} \approx \sum_i W_i \mathbf{g}(\mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi}_i). \quad (9)$$

Different sigma-point methods correspond to different choices of weights W_i and unit sigma-points $\boldsymbol{\xi}_i$. For example, the canonical unscented transform [8] uses the following set of $2n+1$ weights (recall that n is the dimensionality of the state) and sigma-points:

$$W_0 = \frac{\kappa}{n+\kappa}, \quad W_i = \frac{1}{2(n+\kappa)}, \quad i = 1,\ldots,2n,$$
$$\boldsymbol{\xi}_i = \begin{cases} \mathbf{0}, & i = 0, \\ \sqrt{n+\kappa}\,\mathbf{e}_i, & i = 1,\ldots,n, \\ -\sqrt{n+\kappa}\,\mathbf{e}_{i-n}, & i = n+1,\ldots,2n. \end{cases} \quad (10)$$

where κ is a design parameter in the algorithm and $\mathbf{e}_i \in \mathbb{R}^n$ is the unit vector towards the direction of the ith coordinate axis.

Note that sigma-point methods sometimes use different weights for the integrals appearing in the mean and covariance computations of Gaussian filters and smoothers. However, here we will only concentrate on the methods that use the same weights for both in order to derive more direct connections between the methods. For example, the above unscented transform weights are just a special case of more general unscented transforms (see, e.g., [17]).

C. Gaussian Process Regression

Gaussian process quadrature [2], [3] is based on forming a Gaussian process (GP) regression [7] approximation to the integrand using pointwise evaluations and then integrating the approximation. In GP regression [7] the purpose is to predict the value of an unknown function

$$o = g(\mathbf{x}) \quad (11)$$

at a certain test point $(o^*,\mathbf{x}^*)$ based on a finite number of training samples $\mathcal{D} = \{(o_j,\mathbf{x}_j) : j = 1,\ldots,N\}$ observed from it. The difference to classical regression is that instead of postulating a parametric regression function $g_\theta(\mathbf{x};\boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^D$ are the parameters, in GP regression we put a Gaussian process prior with a given covariance function $K(\mathbf{x},\mathbf{x}')$ on the unknown functions $g_K(\mathbf{x})$.

In practice, the observations are often assumed to contain noise and hence a typical model setting is:

$$g_K \sim \mathrm{GP}(0,K(\mathbf{x},\mathbf{x}')),$$
$$o_j = g_K(\mathbf{x}_j) + \epsilon_j, \quad \epsilon_j \sim \mathrm{N}(0,\sigma^2), \quad (12)$$

where the first line above means that the random function g_K has a zero mean Gaussian process prior with the given covariance function $K(\mathbf{x},\mathbf{x}')$. A commonly used covariance function is the exponentiated quadratic (also called squared exponential) covariance function

$$K(\mathbf{x},\mathbf{x}') = s^2 \exp\left(-\frac{1}{2\ell^2}\|\mathbf{x}-\mathbf{x}'\|^2\right), \quad (13)$$

where $s,\ell > 0$ are parameters of the covariance function (see [7]).

The GP regression equations can be derived as follows. Assume that we want to estimate the value of the noise-free function $g(\mathbf{x})$ based on its Gaussian process approximation $g_K(\mathbf{x})$ at a test point $\mathbf{x}$ given the vector of observed values $\mathbf{o} = (o_1,\ldots,o_N)$. Due to the Gaussian process assumption we now get

$$\begin{pmatrix} \mathbf{o} \\ g_K(\mathbf{x}) \end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{K}+\sigma^2\mathbf{I} & \mathbf{k}(\mathbf{x}) \\ \mathbf{k}^{\mathrm{T}}(\mathbf{x}) & K(\mathbf{x},\mathbf{x}) \end{pmatrix}\right), \quad (14)$$

where $\mathbf{K} = [K(\mathbf{x}_i,\mathbf{x}_j)]$ is the joint covariance of observed points, $K(\mathbf{x},\mathbf{x})$ is the (co)variance of the test point, and $\mathbf{k}(\mathbf{x}) = [K(\mathbf{x},\mathbf{x}_i)]$ is the vector of cross covariances with the test point.

The Bayesian estimate of the unknown value of $g_K(\mathbf{x})$ is now given by its posterior mean, given the training data. Because everything is Gaussian, the posterior distribution is Gaussian and hence described by the posterior mean and (auto)covariance functions:

$$\mathrm{E}[g_K(\mathbf{x}) \mid \mathbf{o}] = \mathbf{k}^{\mathrm{T}}(\mathbf{x})(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{o}$$
$$\mathrm{Cov}[g_K(\mathbf{x}) \mid \mathbf{o}] = K(\mathbf{x},\mathbf{x}') - \mathbf{k}^{\mathrm{T}}(\mathbf{x})(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{k}(\mathbf{x}'). \quad (15)$$

These are the Gaussian process regression equations in their typical form [7], in the special case where g is scalar. The extension to multiple output dimensions is conceptually straightforward (see, e.g., [7], [42]), but construction of the covariance functions as well as the practical computational methods tends to be complicated [43], [44]. However, a typical easy approach to the multivariate case is to treat each of the dimensions independently.

D. Gaussian Process Quadrature

In Gaussian process quadrature [2], [3] the basic idea is to approximate the integral of a given function g against a weight function $w(\mathbf{x})$, that is,

$$\mathcal{I}[g] = \int g(\mathbf{x})w(\mathbf{x})d\mathbf{x}, \quad (16)$$

by evaluating the function g at a finite number of points and then by forming a Gaussian process approximation g_K to the function. The integral is then approximated by integrating the Gaussian process approximation (or its posterior mean) which is conditioned on the evaluation points instead of the function itself. Here we assume that g is scalar for simplicity as we can always take a vector function elementwise.

Gaussian process quadratures are related to a regression interpretation of classical Gaussian quadratures which means that we can interpret many of the classical methods as orthogonal polynomial approximations of the integrand evaluated at certain finite number of points [3]. The integral is then approximated by integrating the polynomial instead of the original function. However, the aim of Gaussian process quadrature is to get a good performance in average, whereas in classical polynomial quadratures the integration rule is designed to be exact for a limited class of (polynomial) functions. Still, these approaches are very much linked together [3].

Due to linearity of integration, the posterior mean of the integral of the Gaussian process regressor is given as

$$\mathrm{E}\left[\int g_K(\mathbf{x})w(\mathbf{x})d\mathbf{x} \mid \mathbf{o}\right] = \int \mathrm{E}[g_K(\mathbf{x}) \mid \mathbf{o}]w(\mathbf{x})d\mathbf{x}, \quad (17)$$

where the "training set" $\mathbf{o} = (g(\mathbf{x}_1), \ldots, g(\mathbf{x}_N))$ now contains the values of the function g evaluated at certain selected inputs.

The posterior variance of the integral can be evaluated in an analogous manner, and it is sometimes used to optimize the evaluation points of the function g_N [2]–[5]. The posterior covariance of the approximation is

$$\begin{aligned}&\mathrm{Var}\left[\int g_K(\mathbf{x})w(\mathbf{x})d\mathbf{x} \mid \mathbf{o}\right] \\ &\quad = \iint \mathrm{Cov}[g_K(\mathbf{x}) \mid \mathbf{o}]w(\mathbf{x})d\mathbf{x}\,w(\mathbf{x}')d\mathbf{x}'. \quad (18)\end{aligned}$$

That is, when we approximate the integral (16) with the posterior mean we have

$$\int g(\mathbf{x})w(\mathbf{x})d\mathbf{x} \approx \left[\int \mathbf{k}^{\mathrm{T}}(\mathbf{x})w(\mathbf{x})d\mathbf{x}\right](\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{o}. \quad (19)$$

The posterior variance of the (scalar) integral is

$$\begin{aligned}&\mathrm{Var}\left[\int g_K(\mathbf{x})w(\mathbf{x})d\mathbf{x} \mid \mathbf{o}\right] \\ &\quad = \iint K(\mathbf{x},\mathbf{x}')w(\mathbf{x})d\mathbf{x}\,w(\mathbf{x}')d\mathbf{x}' \\ &\qquad - \left[\int \mathbf{k}^{\mathrm{T}}(\mathbf{x})w(\mathbf{x})d\mathbf{x}\right](\mathbf{K} + \sigma^2\mathbf{I})^{-1}\left[\int \mathbf{k}(\mathbf{x}')w(\mathbf{x}')d\mathbf{x}'\right]. \quad (20)\end{aligned}$$

In this article we are specifically interested in the case of Gaussian weight function, which then reduces the integral appearing in the above expressions (19) and (20) to

$$\left[\int \mathbf{k}^{\mathrm{T}}(\mathbf{x})w(\mathbf{x})d\mathbf{x}\right]_i = \int K(\mathbf{x},\mathbf{x}_i)\mathrm{N}(\mathbf{x} \mid \mathbf{m},\mathbf{P})d\mathbf{x}. \quad (21)$$

It is now easy to see that when the covariance function is a squared exponential $K(\mathbf{x},\mathbf{x}_i) = s^2\exp(-(2\ell^2)^{-1}\|\mathbf{x} - \mathbf{x}_i\|^2)$, this integral can be computed in closed form by using the computation rules for Gaussian distributions. Furthermore if the covariance function is a multivariate polynomial, then these integrals are given by the moments of the Gaussian distributions, which are also available in closed form.

III. GAUSSIAN PROCESS QUADRATURES FOR SIGMA-POINT FILTERING AND SMOOTHING

In this section we start by showing how Gaussian process quadratures (GPQ) can be seen as sigma-point methods and then introduce the Gaussian process transform (GPT). The Gaussian process transform then enables us to construct GPQ-based non-linear filters and smoothers analogously to [17].

A. GPQ as a sigma-point method

In this section the aim is to show how Gaussian process quadratures (GPQ) can be seen as sigma-point methods.

LEMMA III.1 *(GPQ as a sigma-point method). The Gaussian process quadrature (or Bayes-Hermite/Bayesian quadrature) can be seen as a sigma-point-type of integral approximation*

$$\int \mathbf{g}(\mathbf{x})\mathrm{N}(\mathbf{x} \mid \mathbf{m},\mathbf{P})d\mathbf{x} \approx \sum_{i=1}^{N} W_i\mathbf{g}(\mathbf{x}_i), \quad (22)$$

where $\mathbf{x}_i = \mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi}_i$*, with the unit sigma-points* $\boldsymbol{\xi}_i$ *selected according to a predefined criterion, and the weights are determined by*

$$W_i = \left[\left(\int \mathbf{k}^{\mathrm{T}}(\boldsymbol{\xi})\mathrm{N}(\boldsymbol{\xi} \mid \mathbf{0},\mathbf{I})d\boldsymbol{\xi}\right)(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\right]_i, \quad (23)$$

where $\mathbf{K} = [K(\boldsymbol{\xi}_i,\boldsymbol{\xi}_j)]$ *is the matrix of unit sigma-point covariances and* $\mathbf{k}(\boldsymbol{\xi}) = [K(\boldsymbol{\xi},\boldsymbol{\xi}_i)]$ *is the vector of cross covariances. In principle, the choice of unit sigma-points above is completely free, but good choices of them are discussed in the following sections.*

PROOF Let us first use stochastic decoupling (7), which enables us to only consider unit-Gaussian integration formulas of the form (8). Because we can integrate vector functions element-by-element, without loss of generality we can assume that $g(\mathbf{x})$ is single-dimensional. Let us now model the function $\boldsymbol{\xi} \mapsto g(\mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi})$ as a Gaussian process g_K with a given covariance function $K(\boldsymbol{\xi},\boldsymbol{\xi}')$ and fix the training set for the GP regressor by selecting the points

$\boldsymbol{\xi}_i$, $i = 1,\ldots,N$, which also determines the corresponding points $\mathbf{x}_i = \mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi}_i$ such that the training set is $\mathbf{o} = (g(\mathbf{x}_1),\ldots,g(\mathbf{x}_N))$. The GP approximation to the integral now follows from (19):

$$\begin{aligned}\int g(\mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi})\mathrm{N}(\boldsymbol{\xi} \mid 0,\mathbf{I})d\boldsymbol{\xi} \\ \approx \left[\int \mathbf{k}^{\mathrm{T}}(\boldsymbol{\xi})\mathrm{N}(\boldsymbol{\xi} \mid 0,\mathbf{I})d\boldsymbol{\xi}\right](\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{o}, \quad (24)\end{aligned}$$

which when simplified and applied to all the dimensions of $\mathbf{g}$ gives the result.

Note that above we actually assume that the stochastically-decoupled-function $\boldsymbol{\xi} \mapsto g(\mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi})$ instead of the original integrand $g(\mathbf{x})$ has the given covariance function. The reason for this modeling choice is that it enables us to decouple the mean and covariance from the integration formula and hence it is computationally beneficial. This also makes the result invariant to affine transformations of the state and it also has a property that the variability of the functions corresponds to the scale of the problem. However, on the other hand, one might argue that it is the function $g(\mathbf{x})$ which we should actually model and using the stochastically-decoupled-function is "wrong."

REMARK III.1 (Variance of GPQ). From Equation (20) we get that the component-wise variances of the Gaussian process quadrature approximation can be expressed as

$$\begin{aligned}V_j = &\iint K(\boldsymbol{\xi},\boldsymbol{\xi}')\mathrm{N}(\boldsymbol{\xi} \mid \mathbf{0},\mathbf{I})d\boldsymbol{\xi}\,\mathrm{N}(\boldsymbol{\xi}' \mid \mathbf{0},\mathbf{I})d\boldsymbol{\xi}' \\ &- \int \mathbf{k}^{\mathrm{T}}(\boldsymbol{\xi})\mathrm{N}(\boldsymbol{\xi} \mid \mathbf{0},\mathbf{I})d\boldsymbol{\xi}(\mathbf{K} + \sigma^2\mathbf{I})^{-1} \\ &\times \int \mathbf{k}(\boldsymbol{\xi}')\mathrm{N}(\boldsymbol{\xi}' \mid \mathbf{0},\mathbf{I})d\boldsymbol{\xi}'. \quad (25)\end{aligned}$$

Using the above integration approximations we can also define a general Gaussian process transform as follows. The reason for introducing the transform is that the corresponding approximate filters and smoothers can be readily constructed in terms of the transform (cf. [17]), which we will do in the next section.

ALGORITHM III.1 *(Gaussian process transform). The Gaussian process quadrature based Gaussian approximation to the joint distribution of* $\mathbf{x}$ *and the transformed random variable* $\mathbf{y} = \mathbf{g}(\mathbf{x}) + \mathbf{q}$*, where* $\mathbf{x} \sim \mathrm{N}(\mathbf{m},\mathbf{P})$ *and* $\mathbf{q} \sim \mathrm{N}(\mathbf{0},\mathbf{Q})$*, is given by*

$$\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} \sim \mathrm{N}\left(\begin{pmatrix}\mathbf{m}\\ \boldsymbol{\mu}_{\mathrm{GP}}\end{pmatrix}, \begin{pmatrix}\mathbf{P} & \mathbf{C}_{\mathrm{GP}}\\ \mathbf{C}_{\mathrm{GP}}^{\mathrm{T}} & \mathbf{S}_{\mathrm{GP}}\end{pmatrix}\right), \quad (26)$$

where

$$\begin{aligned}\mathbf{x}_i &= \mathbf{m} + \sqrt{\mathbf{P}}\boldsymbol{\xi}_i,\\ \boldsymbol{\mu}_{\mathrm{GP}} &= \sum_{i=1}^{N} W_i\mathbf{g}(\mathbf{x}_i),\\ \mathbf{S}_{\mathrm{GP}} &= \sum_{i=1}^{N} W_i(\mathbf{g}(\mathbf{x}_i) - \boldsymbol{\mu}_{\mathrm{GP}})(\mathbf{g}(\mathbf{x}_i) - \boldsymbol{\mu}_{\mathrm{GP}})^{\mathrm{T}} + \mathbf{Q},\\ \mathbf{C}_{\mathrm{GP}} &= \sum_{i=1}^{N} W_i(\mathbf{x}_i - \mathbf{m})(\mathbf{g}(\mathbf{x}_i) - \boldsymbol{\mu}_{\mathrm{GP}})^{\mathrm{T}}. \quad (27)\end{aligned}$$

Above, $\boldsymbol{\xi}_i$ *is some fixed set of sigma/training points and the weights are given by Equation (23) with some selected covariance function* $K(\boldsymbol{\xi},\boldsymbol{\xi}')$*.*

In this article, at least in the analytical results, we usually assume that the measurements are noise-free, that is, $\sigma^2 = 0$. This enables us to obtain analytically exact relationships with the classical quadrature methods. However, when using Gaussian process quadratures as numerical integration methods, it is often beneficial to have at least a small non-zero value for σ^2 in (23). This kind of "jitter" stabilizes numerics and can even be sometimes used to compensate for inaccuracies in modeling.

EXAMPLE III.1 (GPT with squared exponential kernel). Let us now consider $\xi \in \mathbb{R}$ and select the sigma-point locations to be the ones of unscented transform (10). With the squared exponential covariance function (13) and noise-free measurements ($\sigma^2 = 0$) we then get the weights:

$$W = \begin{pmatrix} \dfrac{\mathrm{e}^{-\frac{\kappa+1}{2(\ell^2+1)}}\left(\ell\mathrm{e}^{\frac{\kappa+1}{2(\ell^2+1)}} - 2\ell\mathrm{e}^{\frac{3(\kappa+1)}{2\ell^2}} + \ell\mathrm{e}^{\frac{\kappa+1}{2(\ell^2+1)}}\mathrm{e}^{\frac{2(\kappa+1)}{\ell^2}}\right)}{\sqrt{\ell^2+1}\left(\mathrm{e}^{\frac{\kappa+1}{\ell^2}} - 1\right)^2} \\ -\dfrac{\ell\mathrm{e}^{\frac{(2\ell^2+3)(\kappa+1)}{2\ell^2(\ell^2+1)}}\left(\mathrm{e}^{\frac{\kappa+1}{2(\ell^2+1)}} - \mathrm{e}^{\frac{\kappa+1}{2\ell^2}}\right)}{\sqrt{\ell^2+1}\left(\mathrm{e}^{\frac{\kappa+1}{\ell^2}} - 1\right)^2} \\ -\dfrac{\ell\mathrm{e}^{\frac{(2\ell^2+3)(\kappa+1)}{2\ell^2(\ell^2+1)}}\left(\mathrm{e}^{\frac{\kappa+1}{2(\ell^2+1)}} - \mathrm{e}^{\frac{\kappa+1}{2\ell^2}}\right)}{\sqrt{\ell^2+1}\left(\mathrm{e}^{\frac{\kappa+1}{\ell^2}} - 1\right)^2} \end{pmatrix}. \quad (28)$$

An interesting property is that in the limit $\ell \to \infty$ we get

$$\lim_{\ell\to\infty} W = \begin{pmatrix} \dfrac{\kappa}{\kappa+1} \\ \dfrac{1}{2(\kappa+1)} \\ \dfrac{1}{2(\kappa+1)} \end{pmatrix} \quad (29)$$

which are the unscented transform weights. We return to this relationship in Section IV-D.

B. GPQs in filtering and smoothing

In this section we show how to construct filters and smoothers using the Gaussian process quadrature approximations. Because Algorithm III.1 can be seen as a sigma-point method, analogously to other sigma-point filters considered, for example, in [17], we can now formulate the following sigma-point filter for the model (3), which uses the unit sigma-points $\boldsymbol{\xi}_i$ and weights W_i defined by Algorithm III.1.

ALGORITHM III.2 *(Gaussian process quadrature filter). The filtering is started from initial mean and covariance, $\mathbf{m}_0$ and $\mathbf{P}_0$, respectively, such that $\mathbf{x}_0 \sim \mathrm{N}(\mathbf{m}_0, \mathbf{P}_0)$. Then the following prediction and update steps are applied for $k = 1,2,3,\ldots,T$.*

Prediction:

1) *Form the sigma points as follows:* $\mathcal{X}_{k-1}^{(i)} = \mathbf{m}_{k-1} + \sqrt{\mathbf{P}_{k-1}}\boldsymbol{\xi}_i$, $i = 1,\ldots,N$.
2) *Propagate the sigma points through the dynamic model:* $\hat{\mathcal{X}}_k^{(i)} = \mathbf{f}(\mathcal{X}_{k-1}^{(i)})$, $i = 1,\ldots,N$.
3) *Compute the predicted mean* $\mathbf{m}_k^-$ *and the predicted covariance* $\mathbf{P}_k^-$*:*

$$\mathbf{m}_k^- = \sum_{i=1}^{N} W_i \hat{\mathcal{X}}_k^{(i)},$$

$$\mathbf{P}_k^- = \sum_{i=1}^{N} W_i (\hat{\mathcal{X}}_k^{(i)} - \mathbf{m}_k^-)(\hat{\mathcal{X}}_k^{(i)} - \mathbf{m}_k^-)^{\mathrm{T}} + \mathbf{Q}_{k-1}.$$

Update:

1) *Form the sigma points:* $\mathcal{X}_k^{-(i)} = \mathbf{m}_k^- + \sqrt{\mathbf{P}_k^-}\boldsymbol{\xi}_i$, $i = 1,\ldots,N$.
2) *Propagate the sigma points through the measurement model:* $\hat{\mathcal{Y}}_k^{(i)} = \mathbf{h}(\mathcal{X}_k^{-(i)})$, $i = 1\ldots N$.
3) *Compute the predicted mean* $\boldsymbol{\mu}_k$*, the predicted covariance of the measurement* $\mathbf{S}_k$*, and the cross-covariance of the state and the measurement* $\mathbf{C}_k$*:*

$$\boldsymbol{\mu}_k = \sum_{i=1}^{N} W_i \hat{\mathcal{Y}}_k^{(i)},$$

$$\mathbf{S}_k = \sum_{i=1}^{N} W_i (\hat{\mathcal{Y}}_k^{(i)} - \boldsymbol{\mu}_k)(\hat{\mathcal{Y}}_k^{(i)} - \boldsymbol{\mu}_k)^{\mathrm{T}} + \mathbf{R}_k,$$

$$\mathbf{C}_k = \sum_{i=1}^{N} W_i (\mathcal{X}_k^{-(i)} - \mathbf{m}_k^-)(\hat{\mathcal{Y}}_k^{(i)} - \boldsymbol{\mu}_k)^{\mathrm{T}}.$$

4) *Compute the filter gain* $\mathbf{K}_k$ *and the filtered state mean* $\mathbf{m}_k$ *and covariance* $\mathbf{P}_k$*, conditional on the measurement* $\mathbf{y}_k$*:*

$$\mathbf{K}_k = \mathbf{C}_k \mathbf{S}_k^{-1},$$
$$\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k[\mathbf{y}_k - \boldsymbol{\mu}_k],$$
$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^{\mathrm{T}}.$$

The result of the filter is a sequence of approximations

$$p(\mathbf{x}_k \mid \mathbf{y}_1,\ldots,\mathbf{y}_k) \approx \mathrm{N}(\mathbf{x}_k \mid \mathbf{m}_k, \mathbf{P}_k), \quad k = 1,2,\ldots,T. \tag{30}$$

Further following the line of thought in [17] we can formulate a sigma-point smoother using the unit sigma-points and weights from Algorithm III.1.

ALGORITHM III.3 *(Gaussian process quadrature sigma-point RTS smoother). The smoothing recursion is started from the filtering result of the last time step $k = T$, that is, $\mathbf{m}_T^s = \mathbf{m}_T$, $\mathbf{P}_T^s = \mathbf{P}_T$ and proceeded backwards for $k = T-1, T-2, \ldots, 1$ as follows.*

1) *Form the sigma points:* $\mathcal{X}_k^{(i)} = \mathbf{m}_k + \sqrt{\mathbf{P}_k}\boldsymbol{\xi}_i$, $i = 1,\ldots,N$.
2) *Propagate the sigma points through the dynamic model:* $\hat{\mathcal{X}}_{k+1}^{(i)} = \mathbf{f}(\mathcal{X}_k^{(i)})$, $i = 1,\ldots,N$.
3) *Compute the predicted mean* $\mathbf{m}_{k+1}^-$*, the predicted covariance* $\mathbf{P}_{k+1}^-$*, and the cross-covariance* $\mathbf{D}_{k+1}$*:*

$$\mathbf{m}_{k+1}^- = \sum_{i=1}^{N} W_i \hat{\mathcal{X}}_{k+1}^{(i)},$$

$$\mathbf{P}_{k+1}^- = \sum_{i=1}^{N} W_i (\hat{\mathcal{X}}_{k+1}^{(i)} - \mathbf{m}_{k+1}^-)(\hat{\mathcal{X}}_{k+1}^{(i)} - \mathbf{m}_{k+1}^-)^{\mathrm{T}} + \mathbf{Q}_k,$$

$$\mathbf{D}_{k+1} = \sum_{i=1}^{N} W_i (\mathcal{X}_k^{(i)} - \mathbf{m}_k)(\hat{\mathcal{X}}_{k+1}^{(i)} - \mathbf{m}_{k+1}^-)^{\mathrm{T}}.$$

4) *Compute the gain* $\mathbf{G}_k$*, mean* $\mathbf{m}_k^s$ *and covariance* $\mathbf{P}_k^s$ *as follows:*

$$\mathbf{G}_k = \mathbf{D}_{k+1}[\mathbf{P}_{k+1}^-]^{-1},$$
$$\mathbf{m}_k^s = \mathbf{m}_k + \mathbf{G}_k(\mathbf{m}_{k+1}^s - \mathbf{m}_{k+1}^-),$$
$$\mathbf{P}_k^s = \mathbf{P}_k + \mathbf{G}_k(\mathbf{P}_{k+1}^s - \mathbf{P}_{k+1}^-)\mathbf{G}_k^{\mathrm{T}}.$$

The approximations produced by the smoother are

$$p(\mathbf{x}_k \mid \mathbf{y}_1,\ldots,\mathbf{y}_T) \approx \mathrm{N}(\mathbf{x}_k \mid \mathbf{m}_k^s, \mathbf{P}_k^s), \quad k = 1,2,\ldots,T. \tag{31}$$

Note that we could cope with non-additive noises in the model by using augmented forms of the above filters and smoothers as in [17]. The fixed-point and fixed-lag smoothers can also be derived analogously as was done in the same reference.

IV. SELECTION OF COVARIANCE FUNCTIONS AND SIGMA-POINT LOCATIONS

The accuracy of the Gaussian process quadrature method and hence the accuracy of the filtering and smoothing methods using it is affected by

1) the covariance function $K(\boldsymbol{\xi}, \boldsymbol{\xi}')$ used and
2) the sigma-point locations $\boldsymbol{\xi}_i$.

Once both of the above are fixed, the weights are determined by Equation (23). In this section we discuss certain useful choices of covariance functions as well as "optimal" choices of sigma-point locations for them. We also discuss the connection of the resulting methods with sigma-point methods such as unscented transforms and Gauss-Hermite quadratures.

A. Squared exponential and minimum variance point sets

In a machine learning context [7] the default choice for a covariance function of a Gaussian process is the squared exponential covariance function in Equation (13). What makes it convenient in Gaussian process quadratures is that the integral required for computing the weights in Equation (23) can be evaluated in closed form (cf. [3], [18]). It turns out that the posterior variance can be computed in closed form as well which is useful because for a given set of sigma-points we can immediately compute the expected error in the integral approximation (assuming that the integrand is indeed a GP)—this is possible because the variance does not depend on the observations at all.

One way to determine the sigma-point locations is to select them to minimize the posterior variance of the integral approximation [2], [3]. In our case this corresponds to minization of the variance in Equation (25) with respect to the points $\boldsymbol{\xi}_{1:N}$. Although the minimization is not possible in closed form, with a moderate N this optimization can be done numerically. Unfortunately, this numerical optimization problem is quite hard, because the optimum is far from being unique due to numerous symmetries appearing in the problem. Figure 1 shows examples of minimum variance point sets optimized by using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [45].

The squared exponential covariance function is not the only possible choice for a covariance function. From the machine learning context we could, for example, choose a Matérn covariance function or some of the scale-mixture-based covariance functions [7]. In that case the weight integral (23) becomes less trivial, but at least we always have a chance to precompute the weights using some (other) multivariate quadrature method. The sigma-point optimization could also be done similarly as for the squared exponential covariance function.

One potential disadvantage of using off-the-shelf covariance functions from machine learning is that they usually do not lead to filtering and smoothing methods which would give the exact result for linear state-space models. Recall that unscented Kalman filters and smoothers as well as Gauss-Hermite filters and smoothers do give the linear Kalman filter result when applied to a linear model. One way to diminish this issue is to use a covariance function which is formed as a sum of, for example, squared exponential covariance function and a suitable polynomial covariance function (see next section). Other ways include an explicit inclusion of the linear part into the regression model.

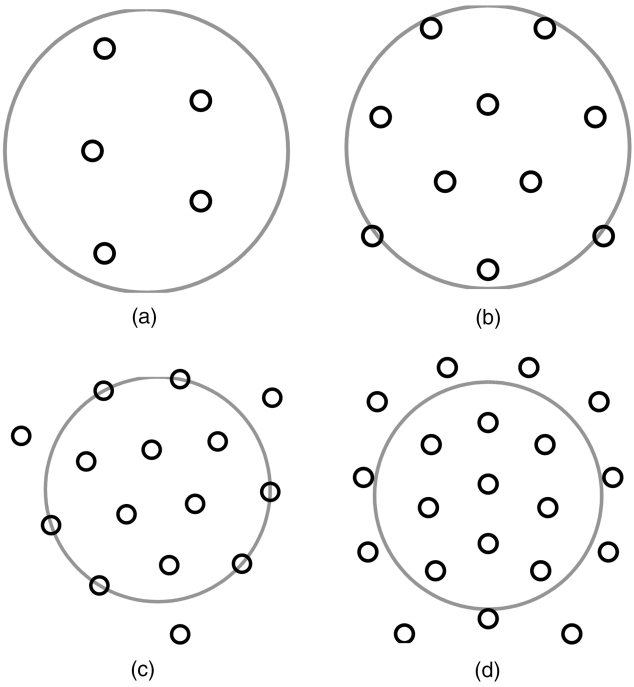


Fig. 1. Minimum variance (2d) point sets for the squared exponential covariance function. The gray circle is the unit circle depicted for visualization purposes. (a) 5 points. (b) 10 points. (c) 15 points. (d) 20 points.

B. UT and spherical cubature rules

In addition to the squared exponential covariance function, another useful class of covariance functions are polynomial covariance functions. They correspond to linear-in-parameters regression using polynomials as the regressor functions. It turns out that also for polynomial covariance functions we can compute the weights (23) in closed form. What is even more interesting is that the Gaussian process quadratures reduce to classical numerical integration methods. In this section we show that with certain selections of symmetric evaluation points we get a classical family of spherically symmetric integration methods of McNamee and Stenger [37] of which the unscented transform [8], [9] can be (retrospectively) seen as a special case [12]. More detailed information on the multivariate Hermite polynomials used below can be found in Appendix A.

THEOREM IV.1 *(UT covariance function). Assume that*

$$K(\boldsymbol{\xi},\boldsymbol{\xi}') = \sum_{q=0}^{3} \sum_{|J|=q} \sum_{p=0}^{3} \sum_{|I|=p} \frac{1}{\mathcal{I}!\mathcal{J}!} \lambda_{\mathcal{I},\mathcal{J}} H_{\mathcal{I}}(\boldsymbol{\xi}) H_{\mathcal{J}}(\boldsymbol{\xi}'), \quad (32)$$

where $\lambda_{\mathcal{I},\mathcal{J}}$s form a positive definite covariance matrix and $H_{\mathcal{I}}(\boldsymbol{\xi})$ are multivariate Hermite polynomials (see Appendix A). If we now select the evaluation points as in UT (10), then the GPQ weights W_i become the UT weights. Furthermore, the posterior variance of the integral approximation is exactly zero.

PROOF The prior $g_K \sim \mathrm{GP}(0, K(\boldsymbol{\xi}, \boldsymbol{\xi}'))$ with the above covariance is equivalent to a parametric model of the form

$$g_K(\boldsymbol{\xi}) = \sum_{p=0}^{3} \sum_{|\mathcal{I}|=p} \frac{1}{\mathcal{I}!} c_{\mathcal{I}} H_{\mathcal{I}}(\boldsymbol{\xi}), \tag{33}$$

where $c_{\mathcal{I}}$ are zero mean Gaussian random variables with the covariances $\lambda_{\mathcal{I},\mathcal{J}} = \mathrm{E}[c_{\mathcal{I}} c_{\mathcal{J}}]$. When the joint covariance matrix $\Lambda = [\lambda_{\mathcal{I},\mathcal{J}}]$ is non-singular, the posterior covariance of the integral being zero is equivalent to that the integral rule is exact for all functions of the form (33) with arbitrary coefficients. Note that we treat Λ as a covariance matrix despite formally being indexed by multi-indices. Clearly with the UT evaluations points, the UT weights are the unique ones that have this property (see, e.g., [17]) and hence the result follows.

Note that the above result also covers the cubature transform (CT), that is, the moment matching rule used in the cubature Kalman filter (CKF) and the smoother, because the transform is a special case of UT [17].

THEOREM IV.2 *(Higher order UT covariance function). Assume that*

$$K(\boldsymbol{\xi}, \boldsymbol{\xi}') = \sum_{q=0}^{P} \sum_{|J|=q} \sum_{p=0}^{P} \sum_{|I|=p} \frac{1}{\mathcal{I}!\mathcal{J}!} \lambda_{\mathcal{I},\mathcal{J}} H_{\mathcal{I}}(\boldsymbol{\xi}) H_{\mathcal{J}}(\boldsymbol{\xi}'). \tag{34}$$

If we select the evaluation points according to order $P = 5, 7, 9, \ldots$ rules in [37], we obtain the higher order integration formulas in [37], which are often referred to as the fifth order, seventh order, ninth order and higher order UTs.

PROOF The result follows analogously to the 3rd order case above.

Figure 2 shows two examples of unscented transform point sets, the 3rd and 5th order point sets (for 2 dimensions).

EXAMPLE IV.1 (Derivation of UT weights from GPQ). Let $\boldsymbol{\xi} \in \mathbb{R}^2$ and consider the GPQ with UT (10) sigma-points and the covariance function (32). With $\sigma = 0$ and $\lambda_{\mathcal{I},\mathcal{J}} = \delta_{\mathcal{I},\mathcal{J}}$ we then obtain the covariance matrix in (35).

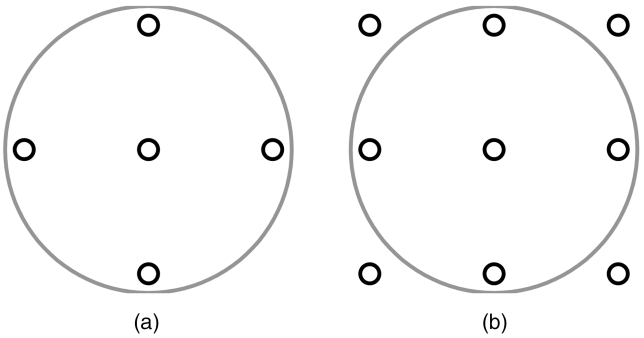


Fig. 2. Unscented transform point sets. (a) UT-3. (b) UT-5.

It also turns out that

$$\int \mathbf{k}^{\mathrm{T}}(\boldsymbol{\xi}) \mathrm{N}(\boldsymbol{\xi} \mid \mathbf{0}, \mathbf{I}) d\boldsymbol{\xi} = (1 \cdots 1) \tag{36}$$

and finally

$$W_{0:4} = \left(\frac{\kappa}{\kappa+2} \quad \frac{1}{2(\kappa+2)} \quad \frac{1}{2(\kappa+2)} \quad \frac{1}{2(\kappa+2)} \quad \frac{1}{2(\kappa+2)} \right)^{\mathrm{T}}, \tag{37}$$

which are indeed the UT weights.

C. Multivariate Gauss-Hermite point sets

The multivariate Gauss-Hermite point sets (see, e.g., [17], [21]) of order P are exact for monomials of the form $x_1^{p_1} \times \cdots \times x_n^{p_n}$, where $p_i \leq 2P-1$ for $i = 1, \ldots, n$. This implies the following covariance function class.

THEOREM IV.3 *(Gauss-Hermite covariance function). Assume that*

$$K(\boldsymbol{\xi}, \boldsymbol{\xi}') = \sum_{\max \mathcal{J} \leq 2P-1} \sum_{\max \mathcal{I} \leq 2P-1} \frac{1}{\mathcal{I}!\mathcal{J}!} \lambda_{\mathcal{I},\mathcal{J}} H_{\mathcal{I}}(\boldsymbol{\xi}) H_{\mathcal{J}}(\boldsymbol{\xi}'), \tag{38}$$

where $\lambda_{\mathcal{I},\mathcal{J}}$s form a positive definite covariance matrix and $H_{\mathcal{I}}(\boldsymbol{\xi})$ are multivariate Hermite polynomials. If we now select the evaluation points to form a cartesian product of the roots of the Hermite polynomials of order P, then the GPQ weights W_i become the multivariate Gauss-Hermite quadrature weights. The posterior variance of the integral approximation is again exactly zero.

$$\mathbf{K} = \begin{pmatrix}
\frac{3}{2} & 1-\frac{\kappa}{4} & 1-\frac{\kappa}{4} & 1-\frac{\kappa}{4} & 1-\frac{\kappa}{4} \\
1-\frac{\kappa}{4} & \frac{\kappa^3}{36}+\frac{\kappa^2}{4}+\frac{13\kappa}{6}+\frac{91}{18} & \frac{1}{2}-\frac{\kappa}{2} & -\frac{\kappa^3}{36}+\frac{\kappa^2}{4}-\frac{7\kappa}{6}-\frac{37}{18} & \frac{1}{2}-\frac{\kappa}{2} \\
1-\frac{\kappa}{4} & \frac{1}{2}-\frac{\kappa}{2} & \frac{\kappa^3}{36}+\frac{\kappa^2}{4}+\frac{13\kappa}{6}+\frac{91}{18} & \frac{1}{2}-\frac{\kappa}{2} & -\frac{\kappa^3}{36}+\frac{\kappa^2}{4}-\frac{7\kappa}{6}-\frac{37}{18} \\
1-\frac{\kappa}{4} & -\frac{\kappa^3}{36}+\frac{\kappa^2}{4}-\frac{7\kappa}{6}-\frac{37}{18} & \frac{1}{2}-\frac{\kappa}{2} & \frac{\kappa^3}{36}+\frac{\kappa^2}{4}+\frac{13\kappa}{6}+\frac{91}{18} & \frac{1}{2}-\frac{\kappa}{2} \\
1-\frac{\kappa}{4} & \frac{1}{2}-\frac{\kappa}{2} & -\frac{\kappa^3}{36}+\frac{\kappa^2}{4}-\frac{7\kappa}{6}-\frac{37}{18} & \frac{1}{2}-\frac{\kappa}{2} & \frac{\kappa^3}{36}+\frac{\kappa^2}{4}+\frac{13\kappa}{6}+\frac{91}{18}
\end{pmatrix} \tag{35}$$

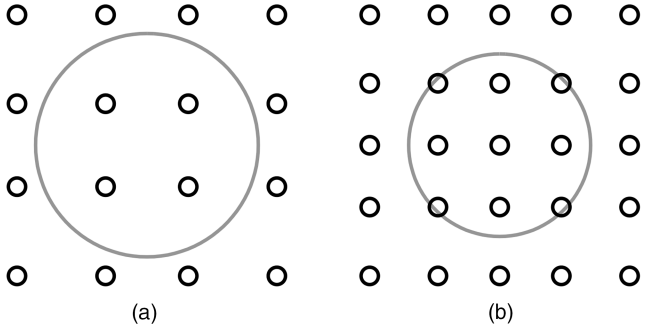


Fig. 3. Gauss-Hermite point sets. (a) GH-4. (b) GH-5.

PROOF Again the result follows from the equivalence of the polynomial approximations and polynomial covariance functions together with the uniqueness of the Gauss-Hermite rule for exact integration of this same function class.

Figure 3 shows 2d sigma-point sets formed as Cartesian products of two 4- and 5-point 1d Gauss-Hermite rules, respectively.

Even when we are using polynomial covariance functions, we are by no means restricted to using the specific points sets corresponding to the classical integration rules. However, given the order of the polynomial kernel and number of sigma-points they are also minimum variance points sets and hence good choices also in average—provided that the integrand is indeed a polynomial. In any case, for an arbitrary set of sigma-points we can use Equation (23) to give the corresponding minimum variance weights.

D. Connection between squared exponential and polynomial Gaussian process quadratures

As discussed in [3], the Gaussian process quadrature with squared exponential covariance function also has a strong connection with classical quadrature methods. This is because we can consider a set of damped polynomial basis functions of the form $\phi_i(\xi) = x^i \exp(-x^2/(2\ell^2))$, which at least informally speaking can be seen to converge to a polynomial basis when $\ell \to \infty$. We can now construct a family of random functions (Gaussian processes) of the form

$$g_\ell(x) = \sum_j c_j \phi_k(x) = \sum_j c_j x^j \exp\left(-\frac{x^2}{2\ell^2}\right), \qquad (39)$$

where $c_j \sim \mathrm{N}(0, (j! l^{2j})^{-1})$. The covariance function of this class is

$$\begin{aligned} K(x,y) &= \sum_i \frac{1}{i!\ell^{2i}} x^i \exp\left(-\frac{x^2}{2\ell^2}\right) y^i \exp\left(-\frac{y^2}{2\ell^2}\right) \\ &= \exp\left(\frac{xy}{\ell^2}\right) \exp\left(-\frac{x^2}{2\ell^2}\right) \exp\left(-\frac{y^2}{2\ell^2}\right) \\ &= \exp\left(-\frac{(x-y)^2}{2\ell^2}\right), \end{aligned} \qquad (40)$$

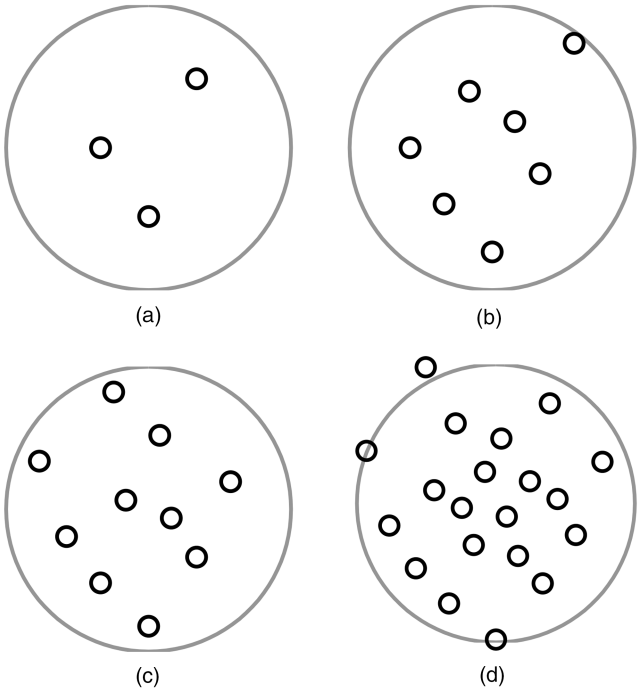


Fig. 4. Hammersley point sets. (a) 3 points. (b) 7 points. (c) 10 points. (d) 20 points.

which is the squared exponential covariance function.

Based on the above, Minka [3] argued (although did not formally prove) that GPQs with squared exponential covariance functions should converge to the classical quadratures. This argument is indeed backed up by our analytical example in Example III.1 where this covergence indeed happens.

E. Random and quasi-random point sets

Recall that one way to approximate the expectation of $\mathbf{g}(\boldsymbol{\xi})$ over a Gaussian distribution $\mathrm{N}(\mathbf{0}, \mathbf{I})$ is to use Monte Carlo integration. In that method we simply draw N samples from the Gaussian distribution $\xi_i \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$ and use them as sigma-points. The classical Monte Carlo approximation to the integral would now correspond to setting $W_i = 1/N$. Alternatively, we could use these random points as sigma-points and evaluate their weights by Equation (23). This leads to an approximation, which is sometimes called the Bayesian Monte Carlo approximation [46], [47].

Instead of sampling from the normal distribution, we can also use quasi-random points sets such as the Hammersley point sets [48], [49]. These are points sets which are designed to give a smaller error in average than random points. The classical method would correspond to setting all weights to $W_i = 1/N$, but again, we can also use Equation (23) to evaluate the weights for the GP quadrature. This corresponds to a "Bayesian quasi Monte Carlo" approximation to the integral. Some examples of Hammersley point sets are shown in Figure 4.

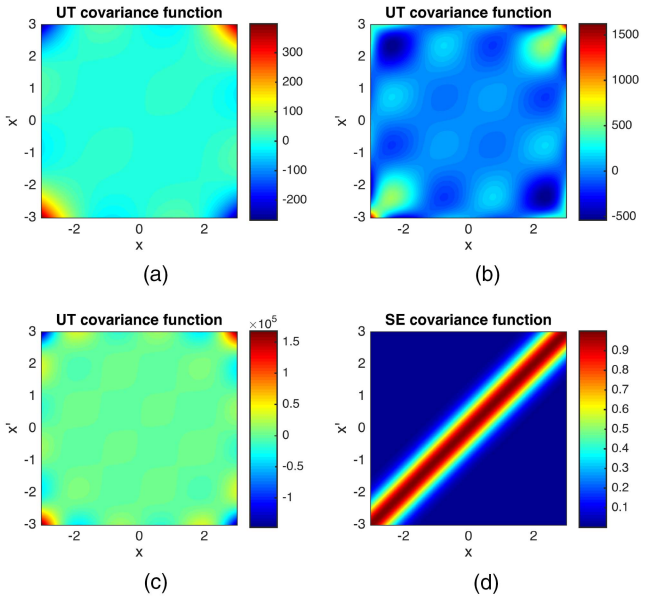


Fig. 5. Covariance functions corresponding to different orders of unscented transforms (UT) and the squared exponential (SE) covariance function ($s = 1$, $\ell = 1/2$) for a single-input scalar-valued Gaussian process. (a) UT-3. (b) UT-5. (c) UT-7. (d) SE.

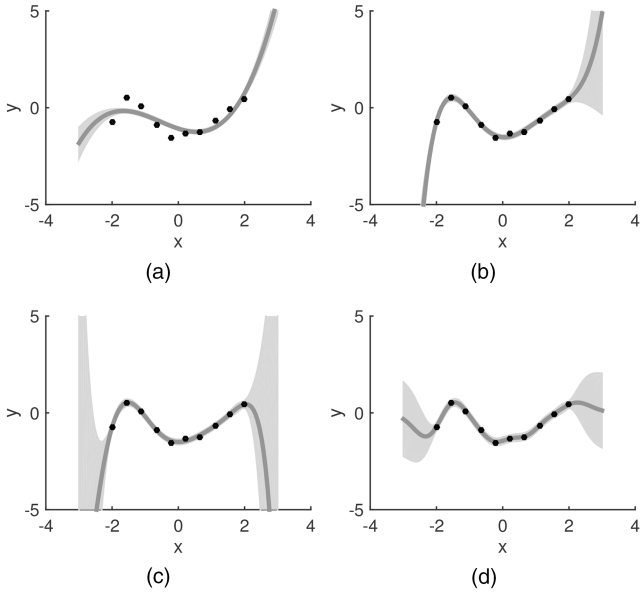


Fig. 6. Regression with covariance functions for UT and SE. (a) UT-3. (b) UT-5. (c) UT-7. (d) SE.

V. NUMERICAL RESULTS

A. Covariance functions and regression implied by unscented transform

The unscented transform covariance functions of orders 3–7 (see Theorems IV.1 and IV.2) and the exponentiated quadratic (i.e., the squared exponential, SE) covariance function (Eq. (13)) are illustrated in Fig. 5. The polynomial nature of the unscented transform (UT) covariance function can be clearly seen in the figures—the UT covariance function as such does not have such a simple local-correlation-interpretation as the SE covariance function has, because the UT covariance functions simply blow up polynomially when moving away from the diagonal.

The corresponding Gaussian process regression results on random data are illustrated in Fig. 6. The polynomial nature of the unscented transform can be clearly seen in the figures. The Gaussian process prediction with the unscented transform covariance function has a clear polynomial shape as expected. Clearly the polynomial fit has less flexibility to explain the data than the exponentiated quadratic fit although the flexibility certainly grows with the polynomial (and thus UT) order.

B. Illustrative high-dimensional example

We use the same test case as in Section VIII.A. of [24], that is, the computation of the first two moments of the function $y(\mathbf{x}) = (\sqrt{1 + \mathbf{x}^{\mathrm{T}}\mathbf{x}})^p$ for $p = 1, -2, -3, -5$. We thus aim to approximate the following integrals:

$$\mathrm{E}[y(\mathbf{x})] = \int \left(\sqrt{1 + \mathbf{x}^{\mathrm{T}}\mathbf{x}}\right)^p \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P}) d\mathbf{x}, \quad (41)$$

$$\mathrm{E}[y^2(\mathbf{x})] = \int (1 + \mathbf{x}^{\mathrm{T}}\mathbf{x})^p \mathrm{N}(\mathbf{x} \mid \mathbf{m}, \mathbf{P}) d\mathbf{x}. \quad (42)$$

Figure 7 shows the result of using the following methods as function of the state-dimensionality:

- *Cubature:* The 3rd order spherical cubature sigma-points ($2n$ points) with the standard integration weights.
- *GPQ-Cubature:* The Gaussian process quadrature with SE covariance function and the 3rd order spherical cubature sigma-points above.
- *GPQ-Hammersley:* The Gaussian process quadrature with SE covariance and $2n$ Hammersley points.

The 3rd spherical cubature points refer to the integration rule proposed in [37], which was also used in the cubature Kalman filter (CKF) in [24]. In the rule, the sigma-points are placed to the intersections of coordinate axes with the origin-centered n-dimensional hypersphere of radius $\sqrt{n}$. Following [24] we measured the accuracy of the methods by computing accurate mean $\boldsymbol{\mu}_0$ and covariance $\boldsymbol{\Sigma}_0$ via Monte Carlo sampling and by comparing it to the approximate means $\mathbf{m}_1$ and covariances $\boldsymbol{\Sigma}_1$ using the following KL-divergence for two Gaussian distributions:

$$\mathrm{KL}[\mathrm{N}_0 \| \mathrm{N}_1] = \frac{1}{2}\left\{ \mathrm{tr}(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0) + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - n + \log\left(\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|}\right)\right\}. \quad (43)$$

The results in Figure 7 show that the GPQ quite consistently gives a bit lower KL-divergence and hence better result than the plain cubature when the cubature points are used. When Hammersley point sets are used, the results vary a bit more: with small state dimensions the results are slightly worse than with the cubature points. When $p \neq 1$, the Hammersley results are much better

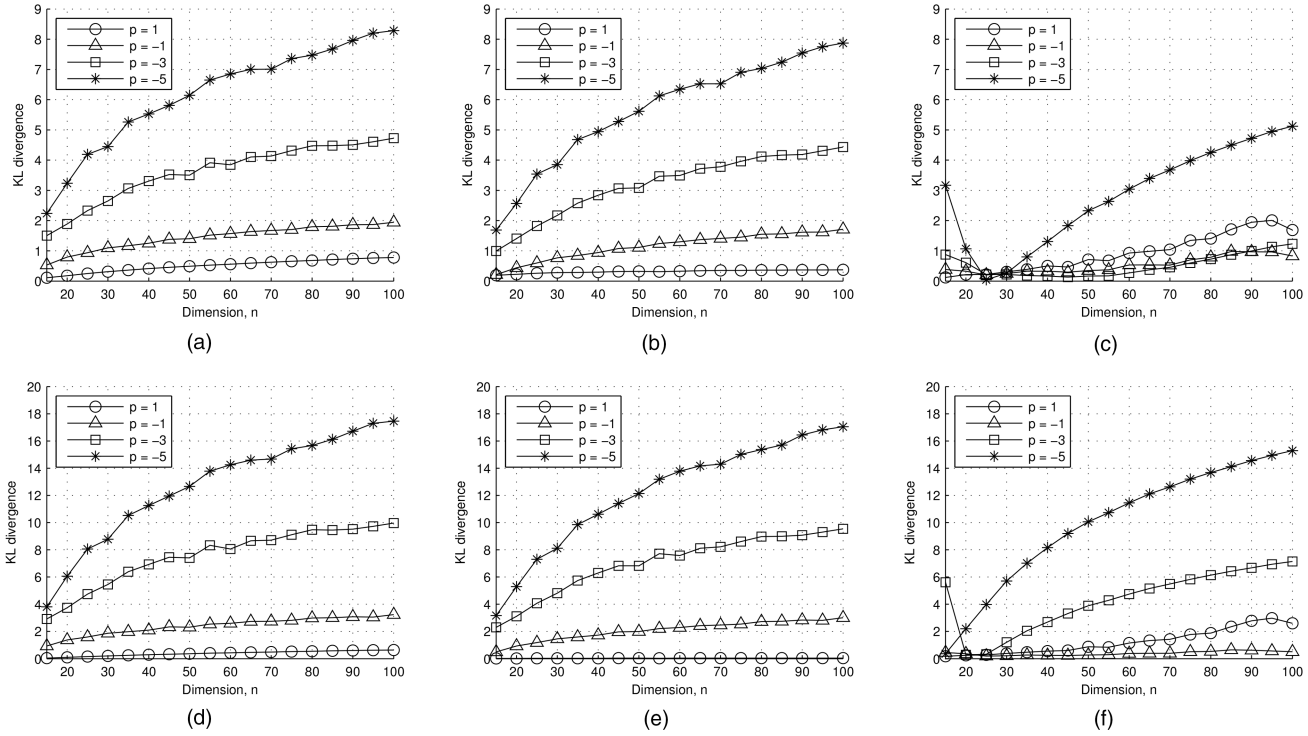


Fig. 7. Comparison of different methods in computing the moment integrals used in [24] for illustrating the performance of the cubature rule. It can be seen that the GPQ methods outperform the cubature rule in most of the cases. (a) Cubature for (41). (b) GPQ-Cubature for (41). (c) GPQ-Hammersley for (41). (d) Cubature for (42). (e) GPQ-Cubature for (42). (f) GPQ-Hammersley for (42).

in high dimensions whereas with $p = 1$ the results are worse than with the cubature point sets.

C. Univariate non-linear growth model

In this section we compare the performance of the different methods in the following univariate non-linear growth model (UNGM) which is often used for benchmarking non-linear estimation methods:

$$x_k = \frac{1}{2}x_{k-1} + 25\frac{x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2k) + q_{k-1},$$
$$y_k = \frac{1}{20}x_k^2 + r_k, \tag{44}$$

where $x_0 \sim \mathrm{N}(0,5)$, $q_{k-1} \sim \mathrm{N}(0,10)$, and $r_k \sim \mathrm{N}(0,1)$.

We generated 100 independent datesets with 500 time steps each and applied the following methods to it: extended, unscented ($\kappa = 2$), and cubature filters and smoothers (EKF/UKF/CKF/ERTS/URTS/CRTS); Gauss-Hermite filters and smoothers with 3, 7, and 10 points (GHKF/GHRTS); Gaussian process quadrature filter and smoother with unscented transform points (GPKFU/GPRTSU) and cubature points (GPKFC/GPRTSC); with Hammersley point sets of sizes 3, 7, and 10 (GPKFH/GPRTSH); and with minimum variance points sets of sizes 3, 7, and 10 (GPKFO/GPRTSO). The covariance function was the exponentiated quadratic with $s = 1$ and $\ell = 3$, and the noise variance was set to 10^{-8}. The RMSE results together with single standard derivation bars are shown in Figures 8 and 9. As can be seen in the figures, with 7 and 10 points the

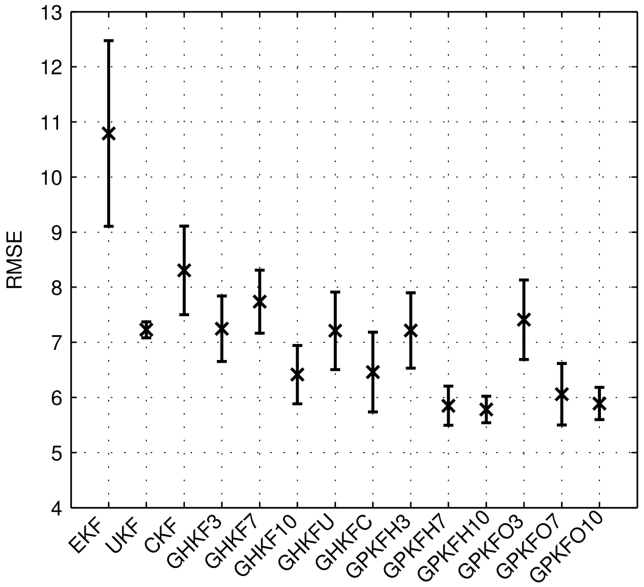


Fig. 8. RMSE results of filters in the UNGM experiment.

Gaussian process quadrature based filters and smoothers have significantly lower errors than almost all the other methods—only Gauss-Hermite with 10 points and the unscented RTS smoother come close.

D. Bearings only target tracking

In this section we evaluate the methods in the bearings only target tracking problem with a coordinated-turn dynamic model, which was also considered in Sec-

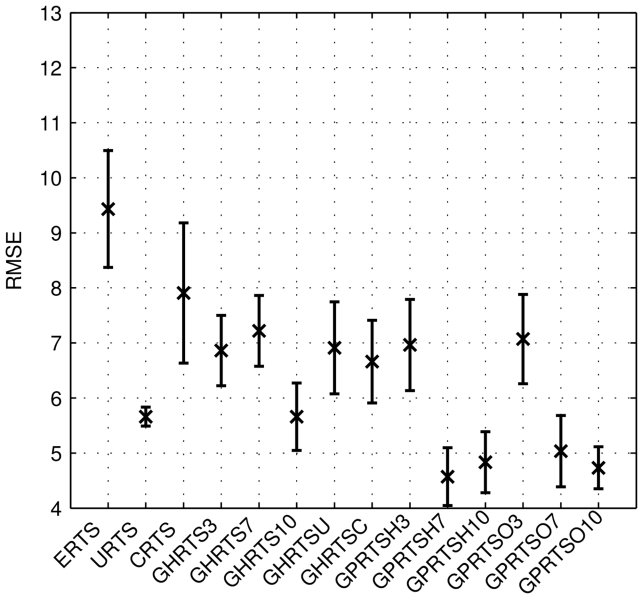


Fig. 9. RMSE results of smoothers in the UNGM experiment.

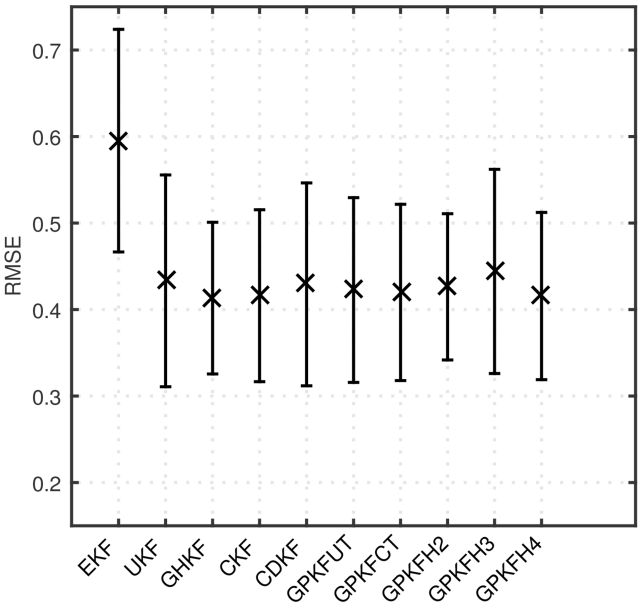


Fig. 10. Position RMSE results of filters in the bearings only experiment.

tion III.A of the article [22]. The non-linear dynamic model is

$$\mathbf{x}_k = \begin{pmatrix} 1 & \frac{\sin(\omega_k \Delta t)}{\omega} & 0 & -\left(\frac{1-\cos(\omega_k \Delta t)}{\omega}\right) & 0 \\ 0 & \cos(\omega_k \Delta t) & 0 & -\sin(\omega_k \Delta t) & 0 \\ 0 & \frac{1-\cos(\omega_k \Delta t)}{\omega_k} & 1 & \frac{\sin(\omega \Delta t)}{\omega} & 0 \\ 0 & \sin(\omega_k \Delta t) & 0 & \cos(\omega_k \Delta t) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \times \mathbf{x}_{k-1} + \mathbf{q}_{k-1}, \tag{45}$$

where the state of the target is $\mathbf{x} = (x_1, \dot{x}_1, x_2, \dot{x}_2, \omega)$, and x_1, x_2 are the coordinates and $\dot{x}_1, \dot{x}_2$ are the velocities in two dimensional space. The time step size is set to $\Delta t = 1$ s and the covariance of the process noise $q_k \sim N(0, Q)$ is

$$Q = \begin{pmatrix} q_1 \frac{\Delta t3}{3} & q_1 \frac{\Delta t2}{2} & 0 & 0 & 0 \\ q_1 \frac{\Delta t2}{2} & q_1 \Delta t & 0 & 0 & 0 \\ 0 & 0 & q_1 \frac{\Delta t3}{3} & q_1 \frac{\Delta t2}{2} & 0 \\ 0 & 0 & q_1 \frac{\Delta t2}{2} & q_1 \Delta t & 0 \\ 0 & 0 & 0 & 0 & q_2 \Delta t \end{pmatrix}, \tag{46}$$

where we used $q_1 = 0.1$ m^{2}s^{-3} and $q_2 = 1.75 \times 10^{-4}$ s^{-3}.

In the simulation setup we have four sensors measuring the angles θ between the target and the sensors. The non-linear measurement model for sensor i can be written as

$$\theta^i = \arctan\left(\frac{x_2 - s_2^i}{x_1 - s_1^i}\right) + r^i, \tag{47}$$

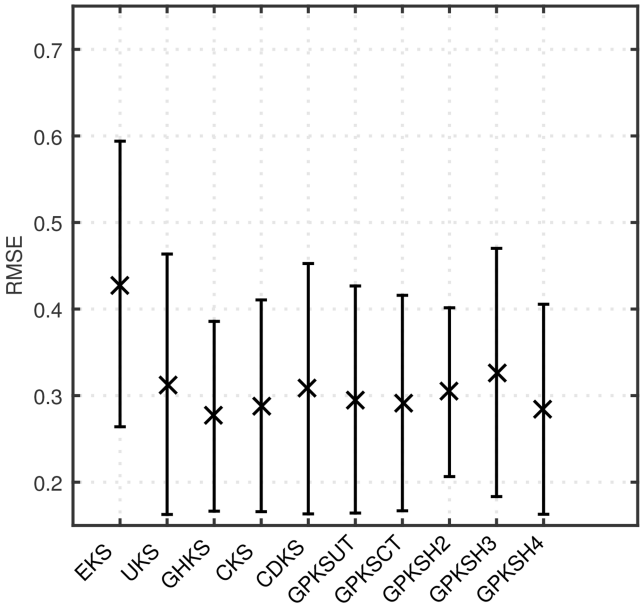


Fig. 11. Position RMSE results of smoothers in the bearings only experiment.

where (s_1^i, s_2^i) is the position of the sensor i in two dimensions, and $r^i \sim N(0, \sigma_\theta^2)$ is the measurement noise. The used parameters were the same as in the article [22].

The RMSE results for the position errors are shown in Figures 10 and 11. Clearly all of the sigma-point methods outperform the Taylor series based methods (EKF/EKS). However, the performances of all the sigma-point methods are very similar: also the Gaussian process quadrature methods give very similar results to the other sigma-point methods. There is a small dip in the errors at the Gauss-Hermite based methods as well as in the highest order Hammersley GPQ method, but

practically the performance of all the sigma-point methods is the same.

VI. CONCLUSION

In this article we have provided a Gaussian process quadrature viewpoint to sigma-point methods and multivariate numerical integration methods for non-linear filtering and smoothing. Using this viewpoint, we have also developed new Gaussian process quadrature based non-linear Kalman filtering and smoothing methods and analyzed their relationship with other sigma-point filters and smoothers. We have also discussed the selection of the evaluation points for the quadratures with respect to different criteria: exactness for multivariate polynomials up to a given order, minimum average error, and quasi-random point sets. We have shown that with suitable selections of (polynomial) covariance functions for the Gaussian processes the filters and smoothers reduce to unscented Kalman filters of different orders, as well as to Gauss-Hermite Kalman filters and smoothers. By numerical experiments we have also shown that the Gaussian process quadrature rules as well as the corresponding filters and smoothers often outperform previously proposed (polynomial) integration rules and sigma-point filters and smoothers.

At this point it is useful to reiterate where the relation of Gaussian process quadratures (GPQs) and sigma-point methods actually originates from. First of all, sigma-point filtering and smoothing methods can be seen as multivariate (classical) quadrature approximations to formal Gaussian (Kalman) filtering and smoothing equations. We also know that classical quadratures can be seen as methods that integrate a polynomial approximant of the function instead of the function itself. From probabilistic numerics we know that Gaussian process (i.e., Bayes-Hermite) quadrature corresponds to integrating a Gaussian process approximant of the function. Now the "kernel trick" tells us that polynomial interpolants can be converted into Gaussian process regressors by using a suitable polynomial covariance function. This implies that using a suitable polynomial covariance function in GPQ approximation of Gaussian (Kalman) filtering and smoothing equations will give us back the conventional sigma-point methods. The known limit results of GPQs converging to classical quadratures also directly translate to convergence of the GPQ based filters and smoothers to the conventional sigma-point methods.

APPENDIX A FOURIER-HERMITE SERIES

Fourier-Hermite series (see, e.g., [50]) are orthogonal polynomial series in a Hilbert space, where the inner product is defined via an expectation of the product over a Gaussian distributions. These series are also inherently related to non-linear Gaussian filtering as they can be seen as generalizations of statistical linearization and they also have a deep connection with unscented transforms, Gaussian quadrature integration, and Gaussian process regression [17], [29], [30].

We define the inner product of the multivariate scalar functions $f(\mathbf{x})$ and $g(\mathbf{x})$ as follows:

$$\langle f, g \rangle = \int f(\mathbf{x}) g(\mathbf{x}) \mathrm{N}(\mathbf{x} \mid \mathbf{0}, \mathbf{I}) d\mathbf{x}. \tag{48}$$

If we now define a norm via $\|f\|_{\mathcal{H}}^2 = \langle f, f \rangle$, and the corresponding distance function $d(f, g) = \|f - g\|_{\mathcal{H}}$, then the functions $\|f\|_{\mathcal{H}} < \infty$ form a Hilbert space $\mathcal{H}$. It now turns out that the multivariate Hermite polynomials form a complete orthogonal basis of the resulting Hilbert space [50].

A multivariate Hermite polynomial with multi-index $\mathcal{I} = \{i_1, \ldots, i_n\}$ can be defined as

$$H_{\mathcal{I}}(\mathbf{x}) = H_{i_1}(x_1) \times \cdots \times H_{i_n}(x_n), \tag{49}$$

which is a product of univariate Hermite polynomials

$$H_p(x) = (-1)^p \exp(x^2/2) \frac{d^p}{dx^p} \exp(-x^2/2). \tag{50}$$

The orthogonality property can now be expressed as

$$\langle H_{\mathcal{I}}, H_{\mathcal{J}} \rangle = \begin{cases} \mathcal{I}!, & \text{if } \mathcal{I} = \mathcal{J} \\ 0, & \text{otherwise,} \end{cases} \tag{51}$$

where we have denoted $\mathcal{I}! = i_1! \cdots i_n!$ and $\mathcal{I} = \mathcal{J}$ means that each of the elements in the multi-indices $\mathcal{I} = \{i_1, \ldots, i_n\}$ and $\mathcal{J} = \{j_1, \ldots, j_n\}$ are equal. We will also denote the sum of indices as $|\mathcal{I}| = i_1 + \cdots + i_n$.

A function $g(\mathbf{x})$ with $\langle g, g \rangle < \infty$ can be expanded into Fourier-Hermite series [50]

$$g(\mathbf{x}) = \sum_{p=0}^{\infty} \sum_{|\mathcal{I}|=p} \frac{1}{\mathcal{I}!} c_{\mathcal{I}} H_{\mathcal{I}}(\mathbf{x}), \tag{52}$$

where $H_{\mathcal{I}}(\mathbf{x})$ are multivariate Hermite polynomials and the series coefficients are given by the inner products $c_{\mathcal{I}} = \langle H_{\mathcal{I}}, g \rangle$.

Consider a Gaussian process $g_G(\mathbf{x})$ that has zero mean and a covariance function $K(\mathbf{x}, \mathbf{x}')$. In the same way as deterministic functions, Gaussian processes can also be expanded into Fourier-Hermite series:

$$g_G(\mathbf{x}) = \sum_{p=0}^{\infty} \sum_{|\mathcal{I}|=p} \frac{1}{\mathcal{I}!} \tilde{c}_{\mathcal{I}} H_{\mathcal{I}}(\mathbf{x}), \tag{53}$$

where the coefficients are given as $\tilde{c}_{\mathcal{I}} = \langle H_{\mathcal{I}}, g_G \rangle$. The coefficients $\tilde{c}_{\mathcal{I}}$ are zero mean Gaussian random variables and their covariance is given as

$$\begin{aligned} \mathrm{E}[\tilde{c}_{\mathcal{I}} \tilde{c}_{\mathcal{J}}] &= \mathrm{E}[\langle H_{\mathcal{I}}, g_G \rangle \langle H_{\mathcal{J}}, g_G \rangle] \\ &= \iint H_{\mathcal{I}}(\mathbf{x}) K(\mathbf{x}, \mathbf{x}') H_{\mathcal{J}}(\mathbf{x}') \\ &\quad \times \mathrm{N}(\mathbf{x} \mid \mathbf{0}, \mathbf{I}) \mathrm{N}(\mathbf{x}' \mid \mathbf{0}, \mathbf{I}) d\mathbf{x} d\mathbf{x}'. \end{aligned} \tag{54}$$

If we define constants $\lambda_{\mathcal{I},\mathcal{J}} = \mathrm{E}[\tilde{c}_{\mathcal{I}}\tilde{c}_{\mathcal{J}}]$ then the covariance function $K(\mathbf{x},\mathbf{x}')$ can be further written as series

$$K(\mathbf{x},\mathbf{x}') = \sum_{q=0}^{\infty} \sum_{|\mathcal{J}|=q} \sum_{p=0}^{\infty} \sum_{|\mathcal{I}|=p} \frac{1}{\mathcal{I}!\mathcal{J}!} \lambda_{\mathcal{I},\mathcal{J}} H_{\mathcal{I}}(\mathbf{x}) H_{\mathcal{J}}(\mathbf{x}'). \tag{55}$$

REFERENCES

[1] A. O'Hagan
Curve fitting and optimal design for prediction (with discussion),
Journal of the Royal Statistical Society. Series B (Methodological), vol. 40(1), pp. 1–42, 1978.

[2] ———
Bayes-Hermite quadrature,
Journal of Statistical Planning and Inference, vol. 29, pp. 245–260, 1991.

[3] T. P. Minka
"Deriving quadrature rules from Gaussian processes,"
Statistics Department, Carnegie Mellon University, Tech. Rep., 2000.

[4] M. A. Osborne, R. Garnett, S. J. Roberts, C. Hart, S. Aigrain, N. P. Gibson, and S. Aigrain
Bayesian quadrature for ratios: Now with even more Bayesian quadrature,
in *International Conference on Artificial Intelligence and Statistics (AISTATS 2012)*, 2012.

[5] M. Osborne, D. Duvenaud, R. Garnett, C. Rasmussen, S. Roberts, and Z. Ghahramani
Active learning of model evidence using Bayesian quadrature,
in *Advances in Neural Information Processing Systems 25*, 2012, pp. 46–54.

[6] S. Särkkä, J. Hartikainen, L. Svensson, and F. Sandblom
Gaussian process quadratures in nonlinear sigma-point filtering and smoothing,
in *Proceedings of FUSION 2014*, 2014.

[7] C. E. Rasmussen and C. K. I. Williams
Gaussian Processes for Machine Learning.
MIT Press, 2006.

[8] S. J. Julier and J. K. Uhlmann
"A general method of approximating nonlinear transformations of probability distributions," Robotics Research Group,
Department of Engineering Science, University of Oxford, Tech. Rep., 1995.

[9] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte
A new method for the nonlinear transformation of means and covariances in filters and estimators,
IEEE Transactions on Automatic Control, vol. 45(3), pp. 477–482, 2000.

[10] E. A. Wan and R. Van der Merwe
The unscented Kalman filter,
in *Kalman Filtering and Neural Networks*, S. Haykin, Ed. Wiley, 2001, ch. 7.

[11] R. 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, OR, USA, 2004.

[12] S. J. Julier and J. K. Uhlmann
Unscented filtering and nonlinear estimation,
Proceedings of the IEEE, vol. 92(3), pp. 401–422, 2004.

[13] S. Särkkä
"Recursive Bayesian inference on stochastic differential equations,"
Doctoral dissertation, Helsinki University of Technology, 2006.

[14] M. Šimandl and J. Duník
Design of derivative-free smoothers and predictors,
in *Preprints of the 14th IFAC Symposium on System Identification*, 2006, pp. 991–996.

[15] S. Särkkä
Unscented Rauch-Tung-Striebel smoother,
IEEE Transactions on Automatic Control, vol. 53(3), pp. 845–849, 2008.

[16] M. Šimandl and J. Duník
Derivative-free estimation methods: New results and performance analysis,
Automatica, vol. 45, no. 7, pp. 1749–1757, 2009.

[17] S. Särkkä
Bayesian filtering and smoothing.
Cambridge University Press, 2013.

[18] M. Deisenroth, R. Turner, M. Huber, U. Hanebeck, and C. Rasmussen
Robust filtering and smoothing with Gaussian processes,
IEEE Transactions on Automatic Control, vol. 57, no. 7, pp. 1865–1871, 2012.

[19] H. J. Kushner
Approximations to optimal nonlinear filters,
IEEE Transactions on Automatic Control, vol. 12, no. 5, pp. 546–556, 1967.

[20] ———
Numerical approximations to optimal nonlinear filters,
in *The Oxford Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovskiĭ, Eds. Oxford, 2011, ch. 28.

[21] K. Ito and K. Xiong
Gaussian filters for nonlinear filtering problems,
IEEE Transactions on Automatic Control, vol. 45(5), pp. 910–927, 2000.

[22] S. Särkkä and J. Hartikainen
On Gaussian optimal smoothing of non-linear state space models,
IEEE Transactions on Automatic Control, vol. 55, no. 8, pp. 1938–1941, 2010.

[23] Y. Wu, D. Hu, M. Wu, and X. Hu
A numerical-integration perspective on Gaussian filters,
IEEE Transactions on Signal Processing, vol. 54(8), pp. 2910–2921, 2006.

[24] I. Arasaratnam and S. Haykin
Cubature Kalman filters,
IEEE Transactions on Automatic Control, vol. 54(6), pp. 1254–1269, 2009.

[25] ———
Cubature Kalman smoothers,
Automatica, vol. 47, no. 10, pp. 2245–2250, 2011.

[26] A. H. Jazwinski
Stochastic Processes and Filtering Theory.
Academic Press, New York, 1970.

[27] M. Nørgaard, N. K. Poulsen, and O. Ravn
New developments in state estimation for nonlinear systems,
Automatica, vol. 36(11), pp. 1627–1638, 2000.

[28] F. Gustafsson and G. Hendeby
Some relations between extended and unscented Kalman filters,
IEEE Transactions on Signal Processing, vol. 60, no. 2, pp. 545–555, 2012.

[29] J. Sarmavuori and S. Särkkä
Fourier-Hermite Kalman filter,
IEEE Transactions on Automatic Control, vol. 57, pp. 1511–1515, 2012.

[30] F. Sandblom and L. Svensson
Moment estimation using a marginalized transform,
IEEE Transactions on Signal Processing, vol. 60, pp. 6138–6150, 2012.

[31] J. Duník, O. Straka, and M. Šimandl
Stochastic integration filter,
IEEE Transactions on Automatic Control, vol. 58, no. 6, pp. 1561–1566, 2013.

[32] O. Straka, J. Duník, and M. Šimandl
Unscented Kalman filter with advanced adaptation of scaling parameter,
Automatica, vol. 50, no. 10, pp. 2657–2664, 2014.

[33] J. Kokkala, A. Solin, and S. Särkkä
Expectation maximization based parameter estimation by sigma-point and particle smoothing,
in *Proceedings of FUSION 2014*, 2014.

[34] J. Duník, O. Straka, M. Šimandl, and E. Blasch
Random-point-based filters: analysis and comparison in target tracking,
IEEE Transactions on Aerospace and Electronic Systems, vol. 51, no. 2, pp. 1403–1421, 2015.

[35] A. Garcia-Fernandez, L. Svensson, M. Morelande, and S. Särkkä
Posterior linearization filter: Principles and implementation using sigma points,
IEEE Transactions on Signal Processing, vol. 63, no. 20, pp. 5561–5573, 2015.

[36] P. Davis and P. Rabinowitz
Methods of Numerical Integration,
2nd ed., ser. Computer science and applied mathematics. Academic Press, 1984.

[37] J. McNamee and F. Stenger
Construction of fully symmetric numerical integration formulas,
Numerische Mathematik, vol. 10, pp. 327–344, 1967.

[38] J. Ko and D. Fox
GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models,
Autonomous Robots, vol. 27, no. 1, pp. 75–90, 2009.

[39] P. Diaconis
Bayesian numerical analysis,
Statistical decision theory and related topics IV, vol. 1, pp. 163–175, 1988.

[40] A. O'Hagan
Some Bayesian numerical analysis,
Bayesian Statistics, vol. 4, pp. 345–363, 1992.

[41] P. Hennig, M. A. Osborne, and M. Girolami
Probabilistic numerics and uncertainty in computations,
Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 471, no. 2179, 2015.

[42] S. Särkkä
Linear operators and stochastic partial differential equations in Gaussian process regression,
in *Artificial Neural Networks and Machine Learning—ICANN 2011*. Springer, 2011, pp. 151–158.

[43] M. Alvarez and N. D. Lawrence
Sparse convolved Gaussian processes for multi-output regression,
in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. The MIT Press, 2009, pp. 57–64.

[44] M. Alvarez, D. Luengo, M. K. Titsias, and N. D. Lawrence
Efficient multioutput Gaussian processes through variational inducing kernels,
in *Proceedings of the 13th International Workshop on Artificial Intelligence and Statistics*, Y. W. Teh and M. Titterington, Eds., 2010, pp. 25–32.

[45] R. Fletcher
Practical methods of optimization,
2nd ed. John Wiley & Sons, 1987.

[46] A. O'Hagan
Monte Carlo is fundamentally unsound,
Journal of the Royal Statistical Society. Series D (The Statistician), vol. 36, pp. 247–249, 1987.

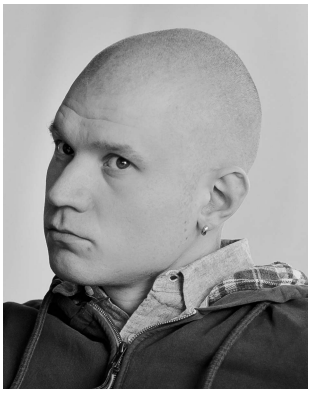
[47] Z. Ghahramani and C. E. Rasmussen
Bayesian Monte Carlo,
in *Advances in neural information processing systems*, 2002, pp. 489–496.

[48] J. M. Hammersley
Monte Carlo methods for solving multivariable problems,
Annals of the New York Academy of Sciences, vol. 86, no. 3, pp. 844–874, 1960.

[49] J. M. Hammersley and D. C. Handscomb
Monte Carlo Methods.
Fletcher & Son, 1964.

[50] P. Malliavin
Stochastic Analysis,
ser. Grundlehren der mathematischen Wissenschaften 313. Springer, 1997.

Simo Särkkä received his M.Sc. degree in engineering physics and mathematics, and D.Sc. degree in electrical and communications engineering from Helsinki University of Technology, Espoo, Finland, in 2000 and 2006, respectively. Currently he is an Associate Professor and Academy Research Fellow with Aalto University, and Technical Advisor and Director of IndoorAtlas Ltd. His research interests are in multi-sensor data processing systems with applications in location sensing, health technology, machine learning, inverse problems, and brain imaging.



Jouni Hartikainen received his M.Sc. degree in computer science and engineering, and D.Sc. (Ph.D.) degree in computational engineering from Aalto University, Espoo, Finland, in 2008 and 2013, respectively. Currently, he is a Senior Technology Manager with Rocsole Ltd. and his research interests are in Bayesian methods for dynamic inverse problems in industrial processes.

Lennart Svensson was born in Älvängen, Sweden in 1976. He received the M.S. degree in electrical engineering in 1999 and the Ph.D. degree in 2004, both from Chalmers University of Technology, Gothenburg, Sweden. He is currently Associate Professor at the Signal Processing group, again at Chalmers University of Technology. His main research interests include machine learning and Bayesian inference in general, and nonlinear filtering and tracking in particular.

Fredrik Sandblom was born in Mölndal, Sweden in 1979. He received the M.Sc. and Ph.D. degrees from Chalmers University of Technology in Gothenburg, Sweden, in 2004 and 2011 respectively. Since 2005 he has been with the Volvo group, working with research and development of active safety systems. His research interests concern system design, object tracking and sensor data fusion; particularly methods for estimating statistical moments and their application to recursive filtering.

Nonlinear Kalman Filters Explained: A Tutorial on Moment Computations and Sigma Point Methods

MICHAEL ROTH
GUSTAF HENDEBY
FREDRIK GUSTAFSSON

Nonlinear Kalman filters are algorithms that approximately solve the Bayesian filtering problem by employing the measurement update of the linear Kalman filter (KF). Numerous variants have been developed over the past decades, perhaps most importantly the popular sampling based sigma point Kalman filters.

In order to make the vast literature accessible, we present nonlinear KF variants in a common framework that highlights the computation of mean values and covariance matrices as the main challenge. The way in which these moment integrals are approximated distinguishes, for example, the unscented KF from the divided difference KF.

With the KF framework in mind, a moment computation problem is defined and analyzed. It is shown how structural properties can be exploited to simplify its solution. Established moment computation methods, and their basics and extensions, are discussed in an extensive survey. The focus is on the sampling based rules that are used in sigma point KF. More specifically, we present three categories of methods that use sigma-points 1) to represent a distribution (as in the UKF); 2) for numerical integration (as in Gauss-Hermite quadrature); 3) to approximate nonlinear functions (as in interpolation). Prospective benefits and downsides are listed for each of the categories and methods, including accuracy statements. Furthermore, the related KF publications are listed.

The theoretical discussion is complemented with a comparative simulation study on instructive examples.

Manuscript received March 31, 2015; revised June 18, 2015; released for publication January 22, 2016.

Refereeing of this contribution was handled by Jindrich Dunik.

Authors' address: Dept. Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden. (E-mail: {roth,hendeby,fredrik}@isy.liu.se).

1557-6418/16/$17.00

I. INTRODUCTION

Many real world problems have a common underlying structure in which the task is to estimate a latent dynamic state from measurements that are taken at specific time instances. Examples include air traffic control [1], where the state comprises the position and velocities of aircraft that are observed by an airport radar; navigation [2], where the state includes the user position and its derivatives, and the measurements might come from a GPS or an inertial measurement unit; or speech processing [3], where the states might be formant frequencies and the measurements are extracted from an estimate of the spectrum. Stochastic state-space models can be used to describe all of these examples in a realistic and mathematically appealing way. The resulting estimation task appears in the form of a Bayesian state estimation problem [4].

The Bayesian filtering problem that is addressed in this tutorial has an elegant mathematical solution, the Bayesian filtering equations [4]. Unfortunately, the mathematical elegance cannot be translated to algorithms in all but a few special cases. In contrast, the Kalman filter [5, 6] *is* a recursive algorithm that has been developed for optimal filtering in linear models, and has been used in countless applications since the 1960s. Interestingly, the first problem where the KF made an impact was in fact a *nonlinear* navigation problem in the Apollo mission [7]. This immediate development of the extended Kalman filter (EKF) has been followed by many nonlinear KF adaptations [6, 8–11].

The interest in the Bayesian filtering problem has significantly increased in the last two decades, mainly due to advances in computing. In the early 1990s, the particle filter [12, 13] was developed to approximately implement the Bayesian filtering equations using sequential Monte Carlo sampling. At the same time, a sampling based extension of the Kalman filter was suggested [14] to improve upon the EKF, eliminate the need for Jacobian matrices, while still retaining the KF measurement update and its computational complexity. Several related algorithms followed: the unscented Kalman filter (UKF) and the underlying unscented transformation were further refined in [15, 16]; numerical integration filters [17, 18] were developed from an alternative starting point to arrive at a very similar recursion; and interpolation variants [17, 19, 20] introduced sampling for function approximation in the spirit of the EKF. In this tutorial, we address these aforementioned sampling based sigma point Kalman filters [21].

Although the literature on nonlinear Kalman filters appears very diverse, it turns out that all variants can be expressed in an intuitive common framework (Gaussian filters [22], Riccati-free Kalman filters [2, 23]). The computation of mean values and covariance matrices of nonlinearly transformed (Gaussian) random variables

appears as a common challenge. The employed methods to approximate these statistical moments distinguishes, for example, the UKF from the divided difference filters of [20]. By emphasizing this common structure, we hope to simplify the reader's access to the extensive literature.

The occurring moment computation problem has also been explored beyond the filtering context [24, 25]. We provide a detailed analysis that reveals useful structural properties, some of which are exploited in the existing methods. Deeper understanding of these properties can give rise to new methods and, in special cases, even reduce n-fold moment integrals to one-dimensional problems.

The moment computation methods that are used in sigma point Kalman filters avoid analytical approximations in favor of sampling techniques. We thoroughly investigate the most common approaches and group them into three categories in which the sigma points carry different roles. More specifically, we present sigma points that approximate distributions as in the UT; sigma points that are used in numerical integration; and sigma points that are used to interpolate nonlinear functions. Interestingly, many equivalences between methods can be shown. Accuracy statements are given where available and also the related KF publications are listed, including recent developments. With this critical survey we hope to clarify the basis of most nonlinear Kalman filters and provide realistic expectations of what a KF can achieve.

Existing overview papers on nonlinear Kalman filters include [26], which significantly extends the numerical integration perspective on the UKF in [17]; [27], in which the UKF and interpolation filters are analyzed and compared as local derivative-free estimators; [23], which establishes relations between sigma point filters and analytical EKF variants; and the recent survey [28], which assumes the linear regression Kalman filter perspective of [29] and discusses algorithms in similar categories as we do in Sec. IV, albeit with less material on the numerical integration and function approximation perspectives. The present tutorial is based on the thesis [30] and complements the existing literature by highlighting the moment computation problem that is central to all nonlinear Kalman filters. Especially the extensive treatment of the analytical moment expressions and the resulting insights for sigma point methods distinguish our paper from previous work.

The structure of the paper is as follows. The introduction is followed by an overview of Bayesian filtering and the unifying framework for nonlinear Kalman filters in Sec. II. The moment computation problem is defined in Sec. III and analyzed in greater detail. Methods to approximately solve the moment integrals are surveyed in Sec. IV. A simulation study with several experiments is presented in Sec. V and followed by Sec. VI with concluding remarks.

II. BAYESIAN STATE ESTIMATION AND NONLINEAR KALMAN FILTERS

This section provides the required background on state-space models and filtering theory. Most importantly, nonlinear Kalman filters are presented in a unified framework that highlights moment computations as central step in each of the algorithms. This motivates for the detailed treatment of moments in Sec. III and IV.

The discussion is extensive. Readers with a background in Bayesian filtering can advance to the nonlinear KF part in Sec. II-D. Alternative references include [4] as a classic on Bayesian state estimation, and more recent treatments in [2] or [22]. A standard reference on the Kalman filter and linear estimation is [6].

A. Stochastic State-Space Models

Stochastic state-space models permit an intuitive and powerful mathematical framework for describing real world processes that are subject to uncertainty, for example the position and velocities of aircraft that are observed by an airport radar.

We consider the general discrete-time model

$$x_k = f(x_{k-1}, v_{k-1}), \tag{1a}$$

$$y_k = h(x_k, e_k), \tag{1b}$$

with the state $x \in \mathcal{X}$, the measurement $y \in \mathcal{Y}$, and the process and measurement noise $v \in \mathcal{V}$ and $e \in \mathcal{E}$, respectively. The function $f : \mathcal{X} \times \mathcal{V} \to \mathcal{X}$ determines the evolution of x in the state difference equation (1a). Similarly, the function $h : \mathcal{X} \times \mathcal{E} \to \mathcal{Y}$ determines the measurement in the measurement equation (1b). Uncertainties in the state evolution and measurements are modeled by asserting that the initial state x_0 and the noise v_{k-1} and e_k are random variables with known distribution for all $k > 0$. Common assumptions are that v_{k-1} and e_k are white and uncorrelated to x_0 and each other; or that the noise sequences[1] and the initial state are independent and admit the joint probability density function

$$p(x_0, v_{0:k-1}, e_{1:k}) = p(x_0) \prod_{l=1}^{k} p(v_{l-1}) p(e_l). \tag{1c}$$

The functions f and h and all densities in (1c) are assumed to be known, but allowed to be time-varying. That allows for including deterministic input signals in (1) through either the function f or the process noise density $p(v_{k-1})$. An extra time index on the functions and densities is omitted for brevity. Following physical reasoning, it is assumed that all measurements are affected by noise and that $\dim(\mathcal{E}) = \dim(\mathcal{Y})$. In fact, additive measurement noise is most common in (1b). For some cases, e.g., when the system (1) is operating in a feedback control loop, the joint density $p(x_0, v_{0:k-1}, e_{1:k})$ cannot be factored as in (1c) because e_k affects v_k.

[1] We use the short hand notation $v_{0:k-1}$ to denote the sequence $\{v_0, v_1, \ldots, v_{k-1}\}$.

An alternative to specifying (1) is to describe the state and measurement processes in terms of the conditional densities[2]

$$x_k \sim p(x_k \mid x_{k-1}), \tag{2a}$$

$$y_k \sim p(y_k \mid x_k). \tag{2b}$$

The symbol $\sim$ in (2a) denotes that, given x_{k-1}, the state x_k is drawn from a conditional distribution that admits the transition density $p(x_k \mid x_{k-1})$. The density $p(y_k \mid x_k)$ is termed likelihood. For the case of additive independent noise in (1), the transition density and likelihood can be specified in terms of the process and measurement noise densities $p(v_{k-1})$ and $p(e_k)$, respectively. The formulation (2) highlights some conditional independence properties in the independent noise case that are important in the derivation of the Bayesian filtering equations [4]:

$$p(x_k \mid x_{0:k-1}, y_{1:k-1}) = p(x_k \mid x_{k-1}), \tag{3a}$$

$$p(y_k \mid x_{0:k}, y_{1:k-1}) = p(y_k \mid x_k). \tag{3b}$$

B. Bayesian State Estimation

The models (1) and (2) fully characterize the state and measurement sequences in a probabilistic manner. This interpretation of the state as a random variable is the cornerstone of Bayesian state estimation, and facilitates the elegant mathematical treatment in terms of probability density functions. In this paper, we shift our attention to estimation problems in which we try to recover the marginal[3] density $p(x_k \mid y_{1:l})$ of a single state x_k given the model and a sequence of measurements $y_{1:l}$.

The previous section has shown how $x_{0:k}$ and $y_{1:k}$ are generated from x_0, $v_{0:k-1}$, and $e_{1:k}$. As a consequence, the joint density $p(x_{0:k}, y_{1:k})$ is a nonlinear transformation of the density $p(x_0, v_{0:k-1}, e_{1:k})$. Slightly more general, the density $p(x_{0:k}, y_{1:l})$ can be obtained in the same way. The manipulations of probability density functions

$$p(x_k, y_{1:l}) = \int p(x_{0:k}, y_{1:l}) dx_{0:k-1}, \tag{4a}$$

$$p(x_k \mid y_{1:l}) = \frac{p(x_k, y_{1:l})}{p(y_{1:l})}, \tag{4b}$$

$$p(y_{1:l}) = \int p(x_k, y_{1:l}) dx_k, \tag{4c}$$

show how the Bayesian state estimation problem of finding (4b) can, in principle, be reduced to the basic operations of conditioning and marginalization. This holds for any l and k. In fact, the order of marginalization and conditioning can be interchanged. Depending on the relation between k and l, three different Bayesian state estimation problems can be distinguished. The case $k > l$ constitutes a prediction problem, the case $k < l$ gives a smoothing or retrodiction problem, and for $k = l$ a filtering problem is obtained. The focus of this article is on filtering and the inherent one-step-ahead prediction, but similar challenges appear in the smoothing problem [22].

The above view of Bayesian state estimation in terms of transformation, conditioning, and marginalization of probability density functions is intuitive. It includes filtering, prediction, and smoothing, and it covers the case of arbitrary correlation in the noise. Unfortunately, it faces some issues. First, finding $p(x_{0:k}, y_{1:k})$ from $p(x_0, v_{0:k-1}, e_{1:k})$ is a challenge beyond hope for all but the simplest models (1). Second, the use of state and measurement sequences does not immediately yield a recursive filtering algorithm.

The Bayesian filtering equations, in contrast, do yield a recursion for the filtering density $p(x_k \mid y_{1:k})$:

$$p(x_k \mid y_{1:k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid y_{1:k-1}) dx_{k-1}, \tag{5a}$$

$$p(x_k \mid y_{1:k}) = \frac{p(x_k, y_k \mid y_{1:k-1})}{p(y_k \mid y_{1:k-1})} = \frac{p(y_k \mid x_k) p(x_k \mid y_{1:k-1})}{p(y_k \mid y_{1:k-1})}, \tag{5b}$$

$$p(y_k \mid y_{1:k-1}) = \int p(y_k \mid x_k) p(x_k \mid y_{1:k-1}) dx_k. \tag{5c}$$

The derivation of (5) relies on the structural properties (3) of the model (1) and can be found in [4]. Similarities between (5) and (4) are apparent, and so it comes as no surprise that some of the challenges of (4) are inherited. First, the marginalization integrals might not be tractable. Second, the filtering density $p(x_k \mid y_{1:k})$ cannot in general be described by a finite number of parameters. For example, in a linear system that is driven by two-component Gaussian mixture noise v_k, the number of parameters to describe $p(x_k \mid y_{1:k})$ grows exponentially in k [2].

C. Linear Gaussian Models and the Kalman Filter

One of the few exceptions for which the Bayesian filtering equations are tractable is the linear Gaussian case. Here, the state transition and measurement equations of (1) can be written as

$$x_k = F x_{k-1} + G v_{k-1}, \tag{6a}$$

$$y_k = H x_k + e_k. \tag{6b}$$

Moreover, the initial state and process and measurement noise are mutually independent Gaussian with

$$p(x_0, v_{0:k-1}, e_{1:k}) = \mathcal{N}(x_0; \hat{x}_0, P_0) \prod_{l=1}^{k} \mathcal{N}(v_{l-1}; 0, Q) \mathcal{N}(e_l; 0, R). \tag{6c}$$

Again, we allow the model to vary with time, e.g., F or Q, but omit the time indices.

[2] For simplicity, we assume the densities to exist. A more general treatment is possible by working on transition kernels [31] instead.

[3] as opposed to the joint density $p(x_{1:k} \mid y_{1:l})$

By induction, it can be easily shown that the filtering density $p(x_k \mid y_{1:k})$ remains Gaussian for all $k > 0$. As a consequence, the Bayesian filtering equations (5) appear as the update equations for mean values and covariance matrices that are known as the Kalman filter (KF) of [5].

From the filtering density $p(x_{k-1} \mid y_{1:k-1}) = \mathcal{N}(x_{k-1}; \hat{x}_{k-1|k-1}, P_{k-1|k-1})$, the one-step-ahead prediction density $p(x_k \mid y_{1:k-1}) = \mathcal{N}(x_k; \hat{x}_{k|k-1}, P_{k|k-1})$ can be obtained. The updated mean value and covariance matrix

$$\hat{x}_{k|k-1} = F\hat{x}_{k-1|k-1}, \tag{7a}$$

$$P_{k|k-1} = FP_{k-1|k-1}F^{\mathrm{T}} + GQG^{\mathrm{T}} \tag{7b}$$

constitute the KF time update or prediction step.

Slightly more general, the joint prediction density of the state and output is given by

$$p(x_k, y_k \mid y_{1:k-1}) = \mathcal{N}\left(\begin{bmatrix} x_k \\ y_k \end{bmatrix}; \begin{bmatrix} \hat{x}_{k|k-1} \\ \hat{y}_{k|k-1} \end{bmatrix}, \begin{bmatrix} P_{k|k-1} & M_k \\ M_k^{\mathrm{T}} & S_k \end{bmatrix}\right), \tag{8}$$

with the mean and covariance of the predicted output

$$\hat{y}_{k|k-1} = H\hat{x}_{k|k-1}, \tag{9a}$$

$$S_k = HP_{k|k-1}H^{\mathrm{T}} + R, \tag{9b}$$

and the cross-covariance

$$M_k = P_{k|k-1}H^{\mathrm{T}}. \tag{9c}$$

The conditioning step in the Bayesian filtering equations simplifies considerably, because rules for the Gaussian distribution can be applied. The KF measurement update yields $p(x_k \mid y_{1:k}) = \mathcal{N}(x_k; \hat{x}_{k|k}, P_{k|k})$ with the filtering mean and covariance

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + M_k S_k^{-1}(y_k - \hat{y}_{k|k-1}) \tag{10a},$$

$$P_{k|k} = P_{k|k-1} - M_k S_k^{-1} M_k^{\mathrm{T}}. \tag{10b}$$

It is common to introduce the Kalman gain $K_k = M_k S_k^{-1}$ and express the measurement update as

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1}) \tag{11a}$$

$$P_{k|k} = (I - K_k H)P_{k|k-1}(I - K_k H)^{\mathrm{T}} + K_k R K_k^{\mathrm{T}} \tag{11b}$$

$$= P_{k|k-1} - K_k S_k K_k^{\mathrm{T}} \tag{11c}$$

$$= (I - K_k H)P_{k|k-1}, \tag{11d}$$

where we listed several alternative expressions for the filtering covariance [6].

The above presentation of the Kalman filter assumes a linear Gaussian model (6), and in fact implements the Bayesian filtering equations. The derivation differs from Kalman's perspective [5] and does not establish certain optimality properties that also hold for non-Gaussian linear systems with known noise statistics [6]. In that case, the KF is still the best linear unbiased estimator (BLUE) but the propagated mean values and covariance matrices are no longer those of Gaussian random variables.

D. Kalman Filters for Nonlinear Models

The preceding sections have shown two extremes. The Bayesian filtering equations (5) are a conceptual solution to a general filtering problem that typically cannot be implemented. In contrast, the Kalman filter solves a specific problem but gives a simple recursive algorithm for updating mean values and covariance matrices. The following section shows how these complementing realities are combined in nonlinear Kalman filters.

Most real world processes exhibit nonlinearities. Furthermore, the process and measurement noise are not necessarily Gaussian. If the nonlinearities are mild and if the noise is not too far from Gaussian, however, a filter that shares a measurement update of the form (10) is a promising candidate. These filters are henceforth termed nonlinear Kalman filters.

The measurement update in the linear KF (10) can be derived from the Gaussian density in (8). In analogy, all nonlinear KF variants employ a Gaussian density

$$\hat{p}(x_k, y_k \mid y_{1:k-1}) \approx \mathcal{N}\left(\begin{bmatrix} x_k \\ y_k \end{bmatrix}; \begin{bmatrix} \hat{x}_{k|k-1} \\ \hat{y}_{k|k-1} \end{bmatrix}, \begin{bmatrix} P_{k|k-1} & M_k \\ M_k^{\mathrm{T}} & S_k \end{bmatrix}\right), \tag{12}$$

to approximate $p(x_k, y_k \mid y_{1:k-1})$, which can be easily shown to be non-Gaussian in the nonlinear case [22]. The measurement update (10) of the linear KF follows and is thus part of any nonlinear KF. This unifying framework is known under the names assumed density filtering [32] or Gaussian filtering [22]. Another interpretation is that the resulting filters locally approximate the Bayesian filtering equations [10, 27]. The differences between specific KF algorithms lie in the way in which the mean values $\hat{x}_{k|k-1}$ and $\hat{y}_{k|k-1}$, and the covariance matrices $P_{k|k-1}$, M_k, and S_k are computed.

Apart from the algorithmic convenience of the KF measurement update, the approximation in (12) is reasonable because the Gaussian is the maximum entropy distribution for a given mean and covariance [33]. That is, if we use the exact mean and covariance of $p(x_k, y_k \mid y_{1:k-1})$ then the Gaussian density in (12) is the least restrictive choice in some sense. Moreover, the Kullback-Leibler divergence

$$\mathrm{KL}(p\|\hat{p}) = -\int p(x_k, y_k \mid y_{1:k-1}) \times \ln\left(\frac{\hat{p}(x_k, y_k \mid y_{1:k-1})}{p(x_k, y_k \mid y_{1:k-1})}\right) dx_k dy_k \tag{13}$$

is minimized by matching the moments [34] of the exact density and its Gaussian approximation, which gives further motivation for using the exact mean values and covariance matrices in (12).

Next, we present the moment integrals to compute the parameters of (12). First, the one-step-ahead prediction of the state is discussed. The model (1a) shows that x_k depends nonlinearly on x_{k-1} and v_{k-1}. In the Kalman filter context with independent noise we assume that the joint density

$$p(x_{k-1}, v_{k-1} \mid y_{1:k-1}) \approx \mathcal{N}(x_{k-1}; \hat{x}_{k-1|k-1}, P_{k-1|k-1}) \mathcal{N}(v_{k-1}; 0, Q) \quad (14)$$

is Gaussian. The mean value and covariance matrix of the predicted state are then given by the moment integrals

$$\begin{aligned} \hat{x}_{k|k-1} &\approx \mathrm{E}\{x_k \mid y_{1:k-1}\} \\ &= \iint f(x_{k-1}, v_{k-1}) \\ &\quad \times p(x_{k-1}, v_{k-1} \mid y_{1:k-1}) dx_{k-1} dv_{k-1}, \end{aligned} \quad (15a)$$

and

$$\begin{aligned} P_{k|k-1} &\approx \mathrm{cov}\{x_k \mid y_{1:k-1}\} \\ &= \iint (f(x_{k-1}, v_{k-1}) - \mathrm{E}\{x_k \mid y_{1:k-1}\}) \\ &\quad \times (f(x_{k-1}, v_{k-1}) - \mathrm{E}\{x_k \mid y_{1:k-1}\})^{\mathrm{T}} \\ &\quad \times p(x_{k-1}, v_{k-1} \mid y_{1:k-1}) dx_{k-1} dv_{k-1}. \end{aligned} \quad (15b)$$

For linear models, (15) simplifies to the KF time update (7). If the noise v_{k-1} enters additively, it does not influence (15a) and appears in (15b) similar to the Q-dependent term in (7b).

The computation of the remaining parameters $\hat{y}_{k|k-1}$, S_k, and M_k can be approached in two different ways. The first option is to accept that

$$y_k = h(f(x_{k-1}, v_{k-1}), e_k) \quad (16)$$

is a function of x_{k-1} and both the process and the measurement noise. Consequently, the expected values must be carried out with respect to a joint density $p(x_{k-1}, v_{k-1}, e_k \mid y_{1:k-1})$.

Alternatively, and common in Kalman filtering algorithms, an intermediate approximation

$$p(x_k \mid y_{1:k-1}) \approx \mathcal{N}(x_k; \hat{x}_{k|k-1}, P_{k|k-1}) \quad (17)$$

is used in the remaining computations. We present the second option below but note that both approaches have their pros and cons. The choice should depend on the accuracy of utilized moment computation methods and the nonlinearities. The intermediate Gaussian density (17) might not reflect the true prediction density well. On the other hand, the composition of the measurement and state transition functions in (16) might be too complicated to be treated directly. For the linear case, there is no difference between the two.

Combining (17) with the measurement noise density yields

$$p(x_k, e_k \mid y_{1:k-1}) \approx \mathcal{N}(x_k; \hat{x}_{k|k-1}, P_{k|k-1}) \mathcal{N}(e_k; 0, R) \quad (18)$$

in the case of independent Gaussian noise. The remaining moment integrals follow as

$$\begin{aligned} \hat{y}_{k|k-1} &\approx \mathrm{E}\{y_k \mid y_{1:k-1}\} \\ &= \iint h(x_k, e_k) p(x_k, e_k \mid y_{1:k-1}) dx_k de_k, \end{aligned} \quad (19a)$$

$$\begin{aligned} S_k &\approx \mathrm{cov}\{y_k \mid y_{1:k-1}\} \\ &= \iint (h(x_k, e_k) - \mathrm{E}\{y_k \mid y_{1:k-1}\}) \\ &\quad \times (h(x_k, e_k) - \mathrm{E}\{y_k \mid y_{1:k-1}\})^{\mathrm{T}} \\ &\quad \times p(x_k, e_k \mid y_{1:k-1}) dx_k de_k, \end{aligned} \quad (19b)$$

$$\begin{aligned} M_k &\approx \mathrm{cov}\{x_k, y_k \mid y_{1:k-1}\} \\ &= \iint (x_k - \mathrm{E}\{x_k \mid y_{1:k-1}\}) \\ &\quad \times (h(x_k, e_k) - \mathrm{E}\{y_k \mid y_{1:k-1}\})^{\mathrm{T}} \\ &\quad \times p(x_k, e_k \mid y_{1:k-1}) dx_k de_k. \end{aligned} \quad (19c)$$

For the linear case, the above expressions simplify to (9).

The computation of (15) and (19) is the crucial challenge in any nonlinear KF. Due to the involved nonlinearities in the model (1) the integrals are typically intractable. Therefore, approximations are utilized. The list of employed concepts includes linearization in the EKF [6]; truncated Taylor polynomials beyond linearization [8, 9, 35]; statistical linearization and alternative function approximations [11, 36–39]; interpolation approaches in the divided difference filters [17, 19, 20]; Monte Carlo [40] and deterministic numerical integration [17, 18, 41, 42] or a combination [43]; and variants of the unscented transformation [15, 16, 44] in the UKF. The majority of the above algorithms are sampling based and can be categorized as sigma point Kalman filters [21].

With the nonlinear KF application in mind we define and analyze the moment computation problems in Sec. III. The methods to solve it, with reference to the related KF variants, are discussed in Sec. IV.

Due to the approximations, many results of the linear KF are no longer valid. For example, there is no nonlinear equivalent to the stationary KF [6] in which $P_{k|k}$ converges to a constant. Furthermore, the order in which measurements are processed in the case of a (block) diagonal measurement noise covariance R matters, as opposed to the linear case [6]. A recommendation from [45] is to process the most accurate measurement first, and then refine the utilized approximation by, for example, re-linearization in an EKF.

The above discussion used Gaussian noise in (14) and (18) as this results in desirable structural properties of the moment integrals (15) and (19). In the non-Gaussian case the relevant densities can be replaced.

We conclude this section with a warning. The Gaussian distribution has limitations because it cannot model phenomena such as heavy tails, skewness, bounded support, or multimodality. An example is shown in Fig. 1, where a Gamma density is illustrated along a Gaussian with the same mean and covariance. The Gamma density is non-zero only for $x > 0$, which is clearly not the case for the Gaussian. Similarly, the Gaussian is symmetric about its mean value. Accordingly, it is a realistic view that Kalman filters, with their close relation to the Gaussian distribution, cannot solve any arbitrary filtering problem. However, more advanced algorithms such as the interacting multiple model (IMM) filter [1] or marginalized particle filters [46] can solve more complicated cases, and employ nonlinear Kalman filters as building blocks.

III. A DETAILED TREATMENT OF THE MOMENT COMPUTATION PROBLEM

The following section introduces and discusses the moment computation problem that is the central challenge in the nonlinear Kalman filters of Sec. II-D, but also relevant beyond the filtering context [24]. We cover structural properties of the moment integrals and ways to exploit them in greater mathematical detail. Furthermore, a compact solution in terms of the Taylor series is presented.

A. The Moment Computation Problem

Sec. II-D highlighted that the challenges in nonlinear Kalman filtering lie in the computation of mean values and covariance matrices of nonlinearly transformed Gaussian random variables. We here generalize this problem to arbitrary functions of Gaussian random variables and functions, that are not necessarily to be seen in a filtering context. Accordingly, the notation is adjusted: the symbol x now represents a generic random variable[4] rather than a state; $f(x)$ represents a generic function rather than the state transition of (1a).

A problem that is encountered in many applications can be formulated as follows. Let the n-dimensional random variable x have a Gaussian distribution $\mathcal{N}(\hat{x}, P)$ with the known mean vector and the covariance matrix

$$\mathrm{E}\{x\} = \hat{x}, \quad \mathrm{cov}\{x\} = P, \tag{20a}$$

and the probability density function

$$p(x) = \mathcal{N}(x; \hat{x}, P). \tag{20b}$$

Furthermore, let $f : \mathbb{R}^n \to \mathbb{R}^m$ be a known function. How can the random variable

$$z = f(x) \tag{21}$$

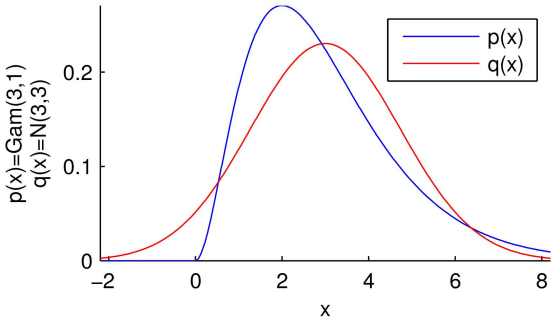


Fig. 1. A Gamma density and a Gaussian with the same mean value and variance.

be characterized in an informative but compact manner?

For special cases, the distribution of z can be obtained via a transformation theorem [47]. In general, however, this is not possible or even desirable. An alternative is to characterize z by its moments. In the Kalman filtering context, the first[5] raw moment and second central moment of z are utilized. Of course, these are better known as the mean value and the covariance matrix and given by the integrals

$$\mathrm{E}\{z\} = \int f(x)p(x)dx, \tag{22a}$$

$$\mathrm{cov}\{z\} = \int (f(x) - \mathrm{E}\{f(x)\}) \times (f(x) - \mathrm{E}\{f(x)\})^{\mathrm{T}} p(x)dx. \tag{22b}$$

Furthermore, cross-covariance matrices of the form

$$\mathrm{cov}\{x,z\} = \int (x - \hat{x})(f(x) - \mathrm{E}\{f(x)\})^{\mathrm{T}} p(x)dx \tag{22c}$$

are required in nonlinear Kalman filters.

An example of a relevant moment computation problem is the conversion between coordinate frames. For example, a Kalman filter might estimate a position in polar coordinates. However, feedback to the user is much more intuitive in Cartesian coordinates, which requires conversion of the polar mean value and its covariance by computing (22a) and (22b). A similar problem is to initialize a Kalman filter with Cartesian state based on the first radar measurements.

The restriction to first and second moments of z is a convenient one, as these are intuitively stored in a vector and a matrix. In principle also higher moments can be considered, although their number grows exponentially with z having m first moments, in the order of m^2 second moments, and so on. Both covariance matrices (22b) and (22c) can be reduced to solving nested problems of the form (22a). Therefore, we concentrate on (22a) in some of the following discussions.

The Gaussian restriction (20b) is chosen because of the Kalman filtering context and because of the structural properties that follow. These are the basis of

[4]The same symbols are used for random variables and their realizations. What is meant should be clear from context.

[5]We avoid the term "order" in connection to moments and reserve it to describe the order of a Taylor series or partial derivatives.

the methods in Sec. IV-C. An extension to arbitrary densities $p(x)$ is immediate.

The moment computation problem (22) requires integration of nonlinear functions over the entire $\mathbb{R}^n$, which is a global problem. However, the integrands are weighted by the probability density function $p(x)$. Hence, only regions in which x has some probability mass contribute to the moment integrals. This is an important aspect to consider when trying to solve (22) by locally approximating f as in Sec. IV-D.

B. Exploiting Structure in the Moment Integrals

The following section shows how the integrals in (22) can be manipulated in order to simplify them and cast them into the structure that some methods of Sec. IV require.

1) Stochastic decoupling:

The following technique is called stochastic decoupling because it can be used to turn expected values with respect to $\mathcal{N}(\hat{x},P)$ into expected values with respect to the standard Gaussian distribution $\mathcal{N}(0,I)$. It employs an $n \times n$ matrix square root $P^{1/2}$ of the covariance P, as presented in (77) of App. A.

Using the rules for substitution in multivariate integrals and $x = P^{1/2}s + \hat{x}$, it is easy to show that

$$\begin{aligned} \mathrm{E}\{z\} &= \int f(x)\mathcal{N}(x;\hat{x},P)dx \\ &= \int f(P^{1/2}s + \hat{x})\mathcal{N}(s;0,I)ds. \end{aligned} \tag{23}$$

The Jacobian determinant of the substitution rule is $\det(P^{1/2}) = \det(P)^{1/2}$ and cancels with the normalization constant of $\mathcal{N}(x;\hat{x},P)$. The quadratic form in the Gaussian density collapsed to $s^{\mathrm{T}}s$.

The reformulation (23) is utilized in several methods throughout Sec. IV. An early reference of its use in filtering is [19].

A related variant is obtained from the substitution rule $x = \sqrt{2}P^{1/2}t + \hat{x}$ and yields

$$\mathrm{E}\{z\} = \int \frac{f(P^{1/2}t + \hat{x})}{\pi^{n/2}} \exp(-t^{\mathrm{T}}t)dt, \tag{24}$$

an integral expression with respect to an exponential weight $\exp(-t^{\mathrm{T}}t)$. The problem of computing (24) is the starting point for Gauss-Hermite quadrature [48] and the numerical integration methods of Sec. IV-C.

2) From Cartesian to spherical coordinates:

The following technique can help to simplify moments of functions that depend on the range only. Furthermore, the separation of moment integrals into spherical and range components has been used to derive integration rules [18, 42, 49].

The change of variables below is a straightforward extension of the Cartesian to polar coordinate change that is taught in undergraduate calculus courses. Consider the relation $s = \Phi(r,\theta_1,\ldots,\theta_{n-1})$ with

$$\begin{aligned} s_1 &= r\sin(\theta_1), \\ s_2 &= r\cos(\theta_1)\sin(\theta_2), \\ s_3 &= r\cos(\theta_1)\cos(\theta_2)\sin(\theta_3), \\ &\vdots \\ s_{n-1} &= r\cos(\theta_1)\cdots\cos(\theta_{n-2})\sin(\theta_{n-1}), \\ s_n &= r\cos(\theta_1)\cdots\cos(\theta_{n-2})\cos(\theta_{n-1}), \end{aligned} \tag{25}$$

where $r \in [0,\infty)$, $\theta_{n-1} \in [-\pi,\pi)$, and $\theta_i \in [-\pi/2,\pi/2)$ for $i = 1,\ldots,n-2$. From (25) follows that $s^{\mathrm{T}}s = r^2$ and thus

$$\mathcal{N}(s;0,I) = \frac{1}{(2\pi)^{n/2}} \exp\left(-\frac{r^2}{2}\right). \tag{26}$$

Using the substitution rule (25) in the moment integral (22a) yields an integral in terms of the range r and the $n-1$ angles θ

$$\begin{aligned} \mathrm{E}\{z\} = \iint f(P^{1/2}\Phi(r,\theta) + \hat{x}) \frac{1}{(2\pi)^{n/2}} \exp\left(-\frac{r^2}{2}\right) \\ \times r^{n-1}\cos^{n-2}(\theta_1)\cos^{n-3}(\theta_2)\cdots\cos(\theta_{n-2})dr\,d\theta \end{aligned} \tag{27}$$

over the above mentioned values, e.g., all $r \geq 0$.

A closely related result can be obtained from the fact that any Gaussian variable with the spherically symmetric distribution $\mathcal{N}(0,I)$ can be generated from a random variable ξ that is uniform on the unit sphere [50]. The substitution rule $x = P^{1/2}r\xi + \hat{x}$ yields

$$\mathrm{E}\{z\} = \iint f(P^{1/2}r\xi + \hat{x}) \frac{1}{(2\pi)^{n/2}} \exp\left(-\frac{r^2}{2}\right) r^{n-1}dr\,d\xi, \tag{28}$$

where the integration is over all $r \geq 0$ and $\xi^{\mathrm{T}}\xi = 1$.

An interesting case occurs when the function depends on the range only, that is, $f(P^{1/2}\Phi(r,\theta) + \hat{x}) = f(P^{1/2}r\xi + \hat{x}) = \varphi(r)$. Then the spherical integration can be performed analytically with the result

$$\mathrm{E}\{z\} = \frac{2\pi^{n/2}}{\Gamma\left(\frac{n}{2}\right)} \int \varphi(r) \frac{1}{(2\pi)^{n/2}} \exp\left(-\frac{r^2}{2}\right) r^{n-1}dr. \tag{29}$$

Alternatively, if $\varphi(r) = \psi(r^2)$ can be written as a function of the squared range, we can arrive at a familiar expression. The substitution rule $r = \sqrt{\rho}$ is a one-to-one relation for $r \geq 0$. The Jacobian determinant is $dr/d\rho = \frac{1}{2}\rho^{-1/2}$, and the integral can be written as

$$\mathrm{E}\{z\} = \int \psi(\rho) \frac{2^{-n/2}}{\Gamma\left(\frac{n}{2}\right)} \rho^{n/2-1} \exp\left(-\frac{\rho}{2}\right) d\rho. \tag{30}$$

The integrand can be recognized as $\psi(\rho)$ times a chi-squared density with n degrees of freedom [49]. In [30], the above technique is used to evaluate the Kullback-Leibler divergence between Gaussian and Student's t

densities, a problem which relates to the filter development of [51].

3) Extension beyond the Gaussian:

The Gaussian distribution is one instance of the wider class of elliptically contoured distributions [49, 50], in which the probability density function $p(x)$ is constant for elliptical regions

$$(x-\hat{x})^{\mathrm{T}}P^{-1}(x-\hat{x}) = c, \tag{31}$$

for $c > 0$. These densities can be written as

$$p(x) = \frac{1}{\sqrt{\det(P)}}h((x-\hat{x})^{\mathrm{T}}P^{-1}(x-\hat{x})), \tag{32}$$

where the function $h : \mathbb{R} \to \mathbb{R}$ is known as density generator [50] and satisfies

$$\int \cdots \int h(s_1^2 + \cdots + s_n^2)ds_1 \cdots ds_n = 1 \tag{33}$$

to ensure that (32) is a valid probability density function. In the Gaussian case, the density generator is

$$h(s^{\mathrm{T}}s) = (2\pi)^{-n/2}\exp\left(-\frac{s^{\mathrm{T}}s}{2}\right) = \mathcal{N}(s;0,I). \tag{34}$$

Both stochastic decoupling and the change to spherical coordinates can be applied to problems in which $p(x)$ is elliptically contoured. A discussion of this is given in [30].

4) Exploiting structure in the function:

In contrast to the preceding paragraphs, the following discussion evolves around structure in $f(x)$ rather than $p(x)$. In fact, the techniques can be applied to arbitrary $p(x)$ and show how the moment computation problem (22) can be broken down systematically.

We consider the case in which x is composed of two random vectors x_1 and x_2, and

$$z = f(x_1, x_2). \tag{35}$$

Following the rules for joint probability density functions, any density $p(x)$ can be factored into

$$p(x) = p(x_1, x_2) = p(x_1 \mid x_2)p(x_2). \tag{36}$$

In the Gaussian case the expression for the marginal density $p(x_2)$ and the conditional density $p(x_1 \mid x_2)$ are well known [49].

Next, we recall to the reader's attention the expressions for the conditional mean

$$\mathrm{E}\{z \mid x_2\} = \int f(x_1, x_2)p(x_1 \mid x_2)dx_1 \tag{37a}$$

and the conditional covariance

$$\begin{aligned}\mathrm{cov}\{z \mid x_2\} = \int (f(x_1, x_2) - \mathrm{E}\{f(x_1, x_2) \mid x_2\})(f(x_1, x_2) \\ - \mathrm{E}\{f(x_1, x_2) \mid x_2\})^{\mathrm{T}}p(x_1 \mid x_2)dx_1.\end{aligned} \tag{37b}$$

These resemble (22a) and (22b) except for the extra conditioning on x_2. A useful consequence of the above is that the moment computation problem (22) can be formulated in terms of the nested expressions

$$\begin{aligned}\mathrm{E}\{z\} &= \mathrm{E}\{\mathrm{E}\{f(x_1, x_2) \mid x_2\}\} \\ &= \int \left(\int f(x_1, x_2)p(x_1 \mid x_2)dx_1 \right) p(x_2)dx_2\end{aligned} \tag{38a}$$

and

$$\begin{aligned}\mathrm{cov}\{z\} = \mathrm{E}\{\mathrm{cov}\{f(x_1, x_2) \mid x_2\}\} \\ + \mathrm{cov}\{\mathrm{E}\{f(x_1, x_2) \mid x_2\}\},\end{aligned} \tag{38b}$$

where the outer expectations average over x_2. The nested covariance expression (38b) is less known than its counterpart (38a) and therefore derived in App. B.

The above expressions are especially useful if $f(x_1, x_2)$ is an affine (or linear) function for a given x_2, i.e.,

$$f(x_1, x_2) = F(x_2)x_1 + g(x_2). \tag{39}$$

Then, the conditional expectations appear as the familiar expressions

$$\mathrm{E}\{z \mid x_2\} = F(x_2)\mathrm{E}\{x_1 \mid x_2\} + g(x_2), \tag{40a}$$

$$\mathrm{cov}\{z \mid x_2\} = F(x_2)\mathrm{cov}\{x_1 \mid x_2\}F(x_2)^{\mathrm{T}}. \tag{40b}$$

The moment computation problem (22) appears as

$$\mathrm{E}\{z\} = \mathrm{E}\{F(x_2)\mathrm{E}\{x_1 \mid x_2\} + g(x_2)\}, \tag{41a}$$

$$\begin{aligned}\mathrm{cov}\{z\} = \mathrm{E}\{F(x_2)\mathrm{cov}\{x_1 \mid x_2\}F(x_2)^{\mathrm{T}}\} \\ + \mathrm{cov}\{F(x_2)\mathrm{E}\{x_1 \mid x_2\} + g(x_2)\}.\end{aligned} \tag{41b}$$

C. The Exact Solution for Differentiable Functions

The section is concluded by a discussion about the exact solution to the moment computation problem (22) for functions that have a convergent Taylor series.

1) Taylor series expansion:

Given that f is differentiable in a neighborhood of $\hat{x}$, it can be expanded as an infinite Taylor series

$$\begin{aligned}f(x) = f(\hat{x}) + \sum_{i=1}^{n} \frac{\partial}{\partial x_i} f(\hat{x})\tilde{x}_i \\ + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2}{\partial x_i \partial x_j} f(\hat{x})\tilde{x}_i\tilde{x}_j \\ + \frac{1}{3!}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} \frac{\partial^3}{\partial x_i \partial x_j \partial x_k} f(\hat{x})\tilde{x}_i\tilde{x}_j\tilde{x}_k + \ldots.\end{aligned} \tag{42}$$

Here, $\tilde{x} = x - \hat{x}$ denotes the deviation from $\hat{x}$, and $(\partial/\partial x_i)f(\hat{x})$ denotes the partial derivative of f with respect to x_i that is evaluated at $\hat{x}$. For polynomial f of degree d, the dth order Taylor series is exact everywhere. From (20) follows that $\tilde{x} \sim \mathcal{N}(0, P)$.

The infinite series (42) is a polynomial in the components of $\tilde{x}$ with the partial derivatives as coefficients.

Consequently, its expected value can be computed from the moments

$$\mathrm{E}\{\tilde{x}_i\},\mathrm{E}\{\tilde{x}_i\tilde{x}_j\},\mathrm{E}\{\tilde{x}_i\tilde{x}_j\tilde{x}_k\},\ldots \tag{43}$$

for all combinations of $i = 1,\ldots,n$; $j = 1,\ldots,n$; $k = 1,\ldots,n$; and so on. The existence of all moments is guaranteed in the Gaussian case. Moreover, all odd moments are zero due to symmetry, for example the first and third item in (43). The even moments can be computed from the characteristic function [49]. For example, the second and fourth moments,

$$\mathrm{E}\{\tilde{x}_i\tilde{x}_j\} = P_{ij}, \tag{44}$$

$$\mathrm{E}\{\tilde{x}_i\tilde{x}_j\tilde{x}_k\tilde{x}_l\} = P_{ij}P_{kl} + P_{ik}P_{jl} + P_{il}P_{jk}, \tag{45}$$

can be expressed in terms of the scalar entries of the covariance matrix P.

We have established that (22a) can, in principle, be computed if the required partial derivatives and moments are available. For polynomial functions, we can compute the exact mean. For arbitrary differentiable functions, we can compute the moments of their Taylor polynomial of degree d. A compact formula that includes all the moments of a scalar x is given in [52].

The next step towards computing (22b) involves an outer product of the infinite series (42). The monomials of a specific degree are no longer grouped as in (42), which renders the approach impractical. A reformulation of the Taylor series can improve upon this situation as we show below.

2) A compact Taylor series notation:

The moment computation problem (22) is easily solved for affine functions $Ax + b$:

$$\mathrm{E}\{Ax + b\} = A\mathrm{E}\{x\} + b, \tag{46a}$$

$$\mathrm{cov}\{Ax + b\} = A\mathrm{cov}\{x\}A^{\mathrm{T}}, \tag{46b}$$

$$\mathrm{cov}\{x, Ax + b\} = \mathrm{cov}\{x\}A^{\mathrm{T}}. \tag{46c}$$

Motivated by the above simplicity, we now turn the nonlinear problem (22) into an infinite dimensional linear problem.

A related technique that uses the Kronecker product to reshape (42) is described in [53]. Higher moments of the Gaussian distribution and the Kronecker product are explored in [54]. Derivative operators in conjunction with the Kronecker product, as used below, are also treated in [55]. The formulas below have been developed in [30], where a more detailed account can be found. A related discussion with a KF background is given in [56].

The tools that facilitate a compact expression for (42) are the d-fold Kronecker product [53]

$$a^{\otimes d} = a \otimes \cdots \otimes a, \tag{47a}$$

the derivative operator

$$D = \left[\frac{\partial}{\partial x_1},\ldots,\frac{\partial}{\partial x_n}\right] \tag{47b}$$

that acts on each row of a function, and a combination of the two

$$D^{\otimes d} = D \otimes \cdots \otimes D \tag{47c}$$

that gives all partial derivatives of order d when applied to a function f.

Using the above, the infinite series in (42) can be written as the linear relation

$$f(x) = f(\hat{x}) + \mathbf{F}(\hat{x})\tilde{\mathbf{x}}(\tilde{x}) \tag{48a}$$

in the infinite dimensional coefficient matrix and monomial vector

$$\mathbf{F}(\hat{x}) = [Df(\hat{x}), \quad (D^{\otimes 2})f(\hat{x}), \quad (D^{\otimes 3})f(\hat{x}), \quad \ldots] \tag{48b}$$

$$\tilde{\mathbf{x}}(\tilde{x}) = [\tilde{x}^{\mathrm{T}}, \quad \tfrac{1}{2}(\tilde{x}^{\otimes 2})^{\mathrm{T}}, \quad \tfrac{1}{3!}(\tilde{x}^{\otimes 3})^{\mathrm{T}}, \quad \ldots]^{\mathrm{T}} \tag{48c}$$

The expression (48a) mimics $Ax + b$. The randomness has been concentrated in $\tilde{\mathbf{x}}(\tilde{x})$, while $\mathbf{F}(\hat{x})$ is fully deterministic. Because of the structure in f, $\mathbf{F}(\hat{x})$ is typically sparse because not all rows of f depend on all entries in x. The $1/d!$ factors of the degree d terms can be either assigned to (48b) or (48c). The latter option turns out to be more convenient.

Now, the moment computation problem can be formulated as

$$\mathrm{E}\{z\} = f(\hat{x}) + \mathbf{F}(\hat{x})\mathrm{E}\{\tilde{\mathbf{x}}(\tilde{x})\}, \tag{49a}$$

$$\mathrm{cov}\{z\} = \mathbf{F}(\hat{x})\mathrm{cov}\{\tilde{\mathbf{x}}(\tilde{x})\}\mathbf{F}(\hat{x})^{\mathrm{T}}, \tag{49b}$$

$$\mathrm{cov}\{x,z\} = \mathrm{cov}\{x,\tilde{\mathbf{x}}(\tilde{x})\}\mathbf{F}(\hat{x})^{\mathrm{T}}. \tag{49c}$$

Apparently, we have replaced (22) by the simpler problem of finding the moments of $\tilde{\mathbf{x}}(\tilde{x})$. From the symmetry in the distribution of $\tilde{x}$ follows a "checkerboard pattern" in

$$\mathrm{E}\{\tilde{\mathbf{x}}(\tilde{x})\} = \\ [0^{\mathrm{T}}, \quad \tfrac{1}{2}\mathrm{E}\{\tilde{x}^{\otimes 2}\}^{\mathrm{T}}, \quad 0^{\mathrm{T}}, \quad \tfrac{1}{4!}\mathrm{E}\{\tilde{x}^{\otimes 4}\}^{\mathrm{T}}, \quad \ldots]^{\mathrm{T}} \tag{50a}$$

and

$$\mathrm{cov}\{\tilde{\mathbf{x}}(\tilde{x})\} = \\ \begin{bmatrix} P & 0 & \frac{1}{3!}\mathrm{cov}\{\tilde{x},\tilde{x}^{\otimes 3}\} & \\ 0 & \frac{1}{2^2}\mathrm{cov}\{\tilde{x}^{\otimes 2}\} & 0 & \cdots \\ \frac{1}{3!}\mathrm{cov}\{\tilde{x}^{\otimes 3},\tilde{x}\} & 0 & \frac{1}{(3!)^2}\mathrm{cov}\{\tilde{x}^{\otimes 3}\} & \\ & \vdots & & \ddots \end{bmatrix}. \tag{50b}$$

The cross-covariance $\mathrm{cov}\{x,\tilde{\mathbf{x}}(\tilde{x})\}$ is the first row of the matrix in (50b). A remaining challenge is the sheer number of terms in the expressions. However, each entry can be obtained from the moments of the Gaussian distribution [49]. These moment expressions simplify significantly if the stochastic decoupling technique is

applied. Then, however, the inherent sparsity that might be present in $\mathbf{F}(\hat{x})$ typically disappears.

Beyond moment computation, the above results are useful for assessing the nonlinearity of f in the neighborhood of an expansion point $\hat{x}$. If the error between the degree d Taylor approximation of f and f is small for carefully chosen x, then f is well represented by a polynomial. In Sec. V-B, Taylor polynomials up to degree 5 are used to assess a range rate measurement in terms of its nonlinearity.

3) Quadratic functions as special case:

We here investigate the special case of a specific $z = f(x)$ with m rows that are given by

$$z_l = a_l + b_l^{\mathrm{T}} x + x^{\mathrm{T}} C_l x. \tag{51}$$

The a_l are scalars, the b_l are n-vectors, and the C_l are $n \times n$-matrices. With the Jacobian and Hessian matrices

$$J_l(x) = b_l^{\mathrm{T}} + x^{\mathrm{T}}(C_l + C_l^{\mathrm{T}}), \quad H_l = (C_l + C_l^{\mathrm{T}}) \tag{52}$$

that contain the first and second order partial derivatives, respectively, (51) can be written as

$$z_l = f_l(\hat{x}) + J_l(\hat{x})(x - \hat{x}) + \tfrac{1}{2}(x - \hat{x})^{\mathrm{T}} H_l (x - \hat{x}). \tag{53}$$

Because f is a polynomial of degree 2, (53) is exact everywhere.

The moments of z can be derived using the tools from the previous subsection together with (45). The mean and covariance are determined by [1]

$$\mathrm{E}\{z_l\} = f_l(\hat{x}) + \tfrac{1}{2}\mathrm{tr}(H_l P), \tag{54a}$$

$$\mathrm{cov}\{z_l, z_k\} = J_l(\hat{x}) P J_k(\hat{x})^{\mathrm{T}} + \tfrac{1}{2}\mathrm{tr}(H_l P H_k P), \tag{54b}$$

$$\mathrm{cov}\{x, z\} = P J(\hat{x})^{\mathrm{T}}, \tag{54c}$$

with $k = 1,\ldots,m$, and $l = 1,\ldots,m$. These expressions can be implemented entirely sampling based as shown in Sec. IV-D.

IV. SIGMA POINT METHODS FOR SOLVING THE MOMENT COMPUTATION PROBLEM

In this section we discuss methods for solving the moment computation problem (22) of Sec. III. The discussion is focused on approximate techniques that are all based on sampling. Apart from a brief coverage of Monte Carlo integration, deterministic methods are presented. The interpretation of the sigma point differs considerably in the unscented transformation, numerical integration, and interpolation approaches.

For convenience, we introduce the abbreviations

$$\hat{z} \approx \mathrm{E}\{z\}, \quad S \approx \mathrm{cov}\{z\}, \quad M \approx \mathrm{cov}\{x, z\} \tag{55}$$

for the moment approximations of the transformed random variable $z = f(x)$.

A. Monte Carlo Integration

This section describes Monte Carlo integration in its most basic form. It is the only *stochastic* moment computation method that we discuss in detail. For a more thorough treatment of Monte Carlo methods in general the reader is referred to [57] and the dedicated chapters in [34, 58].

The basis of Monte Carlo integration is sampling. Hence, it is applicable to the moment computation problem (22) with arbitrary distributions of the random variable x, as long as they allow sampling. Specifically, a large number N of realizations $\{x^{(i)}\}_{i=1}^{N}$ is generated, passed through the function f to yield $\{z^{(i)} = f(x^{(i)})\}_{i=1}^{N}$, and then used to approximate $\mathrm{E}\{z\}$ in (22a) by the sample average

$$\hat{z} = \frac{1}{N}\sum_{i=1}^{N} z^{(i)}. \tag{56a}$$

In a similar manner, the covariance matrices $\mathrm{cov}\{z\}$ and $\mathrm{cov}\{x, z\}$ of (22b) and (22c) are approximated by

$$S = \frac{1}{N-1}\sum_{i=1}^{N} (z^{(i)} - \hat{z})(z^{(i)} - \hat{z})^{\mathrm{T}}, \tag{56b}$$

$$M = \frac{1}{N-1}\sum_{i=1}^{N} (x^{(i)} - \hat{x})(z^{(i)} - \hat{z})^{\mathrm{T}}. \tag{56c}$$

Justification for the above expressions is given by the law of large numbers which states that the averages in (56) indeed converge to the true moments as N tends to infinity. Another important point is that the accuracy of (56a), expressed in terms of its covariance

$$\mathrm{cov}\{\hat{z}\} = \frac{1}{N}\mathrm{cov}\{z\}, \tag{57}$$

depends solely on the true covariance, not the dimension n of x [58]. As a result some problems can be solved with very few samples N, although n is big, whereas others cannot be solved by Monte Carlo integration at all. The standard deviation of a scalar $\hat{z}$ decays as $1/\sqrt{N}$ which is rather slow.

Monte Carlo methods impose no restriction on the functional form of f. If the true moments exist, then (56) will work for sufficiently large N. Due to the random sampling, there is always some variation in the results of (56). These variations decrease with increased sample size N, and can be reduced further by employing variance reduction techniques [57]. If it is difficult to generate samples of x directly, the concept of importance sampling can be used, which actually builds the basis for the particle filter [12, 13]. Even more advanced sampling schemes are Markov chain Monte Carlo methods [57].

The idea to use Monte Carlo integration in a nonlinear Kalman filter is mentioned in [23, 40]. It could be argued that the required computational complexity to run a Monte Carlo Kalman filter contradicts the idea of maintaining only mean values and covariance matrices. Interestingly, a related KF variant beyond the scope of this paper, the ensemble Kalman filter [59], uses Monte Carlo sampling to mimic the KF algorithm in

high-dimensional state spaces that prohibit the storage of $n \times n$ covariance matrices.

B. The Sigma Point Approximation of a Distribution and the Unscented Transformation

The ideas of the following paragraphs were introduced in the filtering context in [14], together with the term sigma points. Later, the somewhat mysterious name *unscented* transformation (UT) was given to the moment computation method, and the resulting filter became known as the unscented Kalman filter (UKF) [16].

As starting point, we can assume the moment computation problem (22) for a discrete random variable x that takes values from the set $\{x^{(i)}\}_{i=1}^{N}$ with the probabilities $\Pr\{x^{(i)}\} = w^{(i)}$. The integrals in (22) then appear as finite sums. The resulting simplicity therefore justifies the idea to approximate the continuous density $p(x)$ of (22) with a discrete point mass function or Dirac mixture density [60].

The UT [15] implements the above idea by systematically selecting N sigma points and weights $\{x^{(i)}, w^{(i)}\}_{i=1}^{N}$. As in MC integration, the function f is evaluated at every sigma point to yield $\{z^{(i)}\}_{i=1}^{N}$. The moment integrals (22) are approximated by

$$\hat{z} = \sum_{i=1}^{N} w^{(i)} z^{(i)}, \tag{58a}$$

$$S = \sum_{i=1}^{N} w^{(i)} (z^{(i)} - \hat{z})(z^{(i)} - \hat{z})^{\mathrm{T}}, \tag{58b}$$

$$M = \sum_{i=1}^{N} w^{(i)} (x^{(i)} - \hat{x})(z^{(i)} - \hat{z})^{\mathrm{T}}. \tag{58c}$$

The expressions resemble a weighted version of (56). However, the number of sigma points is typically much smaller than in Monte Carlo sampling with $N = 2n + 1$ in the most common variant [14].

1) Sigma point and weight selection:

The problem of approximating a probability density function by a number of representative samples goes beyond the moment computation context. For example, the quantization problem [33, 61] in information theory is closely related. For computational reasons, the number of points should be small. In order to approximate the continuous density well, however, more points would be valuable.

The UT sigma points are chosen deterministically such that the mean and covariance of x are preserved in the weighted sample mean and covariance

$$\sum_{i=1}^{N} w^{(i)} x^{(i)} = \hat{x}, \tag{59a}$$

$$\sum_{i=1}^{N} w^{(i)} (x^{(i)} - \hat{x})(x^{(i)} - \hat{x})^{\mathrm{T}} = P. \tag{59b}$$

The condition (59) guarantees that (58) is correct for affine functions. Also constraints on higher moments can, in principle, be included [16]. For any fixed number of points $N > n$, infinitely many combinations of weights and samples satisfy (59).

The original sampling scheme [15] utilizes the columns $\{u_i\}_{i=1}^{n}$ (78) of the $n \times n$ matrix square root $P^{1/2}$ (77) (App. A) to generate $N = 2n + 1$ sigma points and weights

$$x^{(0)} = \hat{x}, \tag{60a}$$

$$x^{(\pm i)} = \hat{x} \pm \sqrt{(n + \kappa)} u_i, \tag{60b}$$

$$w^{(0)} = \frac{\kappa}{n + \kappa}, \tag{60c}$$

$$w^{(\pm i)} = \frac{1}{2(n + \kappa)}, \tag{60d}$$

where $i = 1, \ldots, n$. The signed superscripts $(\pm i)$ underline the symmetry in the sigma points. Accordingly, the summation in (58) is replaced by a sum from $-n$ to n. The alternative expression for (58a)

$$\hat{z} = w^{(0)} z^{(0)} + \sum_{i=1}^{n} w^{(\pm i)} (z^{(+i)} + z^{(-i)}) \tag{61}$$

establishes a relation to the divided difference methods in Sec. IV-D.

The authors of [15] suggest that κ can be chosen as any number such that $\kappa \neq -n$ and recommend $\kappa = 3 - n$. The square root in (60b), however, suggests that $\kappa > -n$. Furthermore, the obtained negative $w^{(i)}$ for $0 > \kappa > -n$ compromise the interpretation as point mass function and can lead to indefinite S in (58b). For $\kappa = 0$, the UT is identical to a cubature rule (Sec. IV-C) that is used in [18].

For large n, the sigma points in (60) are located far from the mean $\hat{x}$, which might yield degraded results for functions f that change severely over the x-space. The scaled UT [62] addresses this by moving the samples towards $\hat{x}$ without violating (59). The procedure has been condensed into a selection scheme similar to (60) by [21].

In the selection scheme, the parameter κ is replaced by $\lambda = \alpha^2(n + \kappa) - n$ with $0 < \alpha \leq 1$. The samples and weights for (58a) or (61) are

$$x^{(0)} = \hat{x}, \tag{62a}$$

$$\begin{aligned} x^{(\pm i)} &= \hat{x} \pm \sqrt{n + \lambda} u_i \\ &= \hat{x} \pm \alpha \sqrt{n + \kappa} u_i, \end{aligned} \tag{62b}$$

$$w^{(0)} = \frac{\lambda}{n + \lambda}, \tag{62c}$$

$$w^{(\pm i)} = \frac{1}{2(n + \lambda)}, \tag{62d}$$

with $i = 1, \ldots, n$. The covariance (58b) is computed with the weight $w_c^{(0)} = (\lambda/(n + \lambda)) + 1 - \alpha^2 + \beta$ for the $z^{(0)}$ term instead. Here, the scalar parameter β is a correction

term [16] which is recommended to be 2 for Gaussian x. With some effort, the motivation for the above choice can be extracted from [16] or [21]. The latter recommends that the choice of α should depend on f as well as $\mathrm{cov}\{x\} = P$, whereas the choice of κ is secondary.

Alternative sigma point generation schemes beyond moment matching were investigated in [60]. Instead of a simple generation rule, the points are chosen to minimize a cost function. The procedure is more costly but allows the user to choose an arbitrary number of sigma points N. In the spirit of the stochastic decoupling technique (Sec. III-B) the authors of [39] employ the method of [60] to generate a sigma point approximation to $\mathcal{N}(0, I)$ offline. These carefully selected samples are then transformed to represent $x \sim \mathcal{N}(\hat{x}, P)$.

2) Accuracy considerations:

The UT is built upon the idea that a good approximation of the distribution of x should also give a good approximation of the distribution of z. Its implementation is simple and requires hardly more than evaluating f for all sigma points. Albeit the intuition, accuracy statements are difficult to make. After all, how the characteristics of the weighted sigma points in x carry over to z is highly dependent on the function f.

In an attempt to show the UT accuracy, the authors of [15] expand the covariance (22b) as infinite Taylor series and compare the terms of low degree to the solutions provided by linearization and UT. It is argued that the former gives only the first term in the infinite series, whereas the UT does not truncate. So, more terms of low degree in the infinite Taylor expansion of (22b) are matched by UT. This observation has lead to the easily misinterpreted statement that the UT is "correct up to the second order" [15, App. II], but it is not clear whether this "order" refers to degree of the Taylor polynomial of (22b) or to the moments of the transformed samples. Certainly, the statement has led to misunderstandings.

To be clear, the UT does not give the correct first and second[6] moments (22a) and (22b) for arbitrary nonlinearities. Furthermore, the UT does not give the moments of a degree two Taylor approximation of f (Sec. III-C). This is discussed and demonstrated for a simple example in [23], and further clarified in the simulation study of polynomial function in Sec. V-A. Another extensive discussion of the statements in [15] is given in [21]. Although the UT often works well, concise accuracy statements are difficult to make. The numerical integration perspective in Sec. IV-C alleviates this to some extent.

3) Unscented and related Kalman filters:

The first UKF was suggested in [14] but the ideas took several years before they appeared in journal format [15]. A complete account of the developers is given in [16]. An interpretation of the UKF as performing statistical linearization is discussed in [63]. The UKF as one member of the sigma point KF class is discussed in [21], including its use for parameter estimation and an account of the accuracy statements in [15]. The relations between the UKF and divided difference filters [17, 20] are investigated in [27], and the authors of [23] discuss asymptotic relations to analytical EKF. An extensive account of the numerical integration perspective of UKF is given in [26]. The authors of [44] investigate the scaling parameter in UKF variants with the sample set (60) and devise an adaptive scaling method for κ in an online filter. The recent smart sampling Kalman filter [39] makes use of the alternative sampling method [60] and allows the user to select an arbitrary number of sigma points N. References to comparative simulation studies are listed in Sec. V.

C. Sigma Points and Numerical Integration

The discussion below results in methods that appear very similar to the UT and, in fact, share the formulas (58).

As starting point, we consider the problem of computing the integral of an exponentially weighted function $g(t)$

$$\int \cdots \int g(t) \exp(-t^{\mathrm{T}} t) dt_1 \ldots dt_n. \tag{63}$$

The relation to the moment computation problem (22) with Gaussian x is established using the stochastic decoupling technique (Sec. III-B). Specifically, the conversion of (22a) to (63) is given in (24).

The problem of computing (63) has been addressed by mathematicians for centuries. The suggested solutions, for example Gauss-Hermite quadrature rules [48], use formulas that are similar to (58a):

$$\int g(t) \exp(-t^{\mathrm{T}} t) dt \approx \sum_{i=1}^{N} w^{(i)} g(t^{(i)}). \tag{64}$$

The classical numerical integration literature [64, 65] provides rules to select the N weights $w^{(i)}$ and sigma points $t^{(i)}$ such that (64) is exact for polynomial functions of degree[7] d. The number of required points N increases as d increases. The built-in accuracy statement is an advantage over the UT. However, by showing the equivalence between an integration rule and a UT variant, the accuracy statements can be transferred to the latter.

For the moment computation problem (22) a rule of degree d gives the correct mean value (22a) for polynomial f of degree d. The covariance (22b), however, requires integration of outer products of f. Therefore, an integration rule of degree d computes (22b) correctly only if f has at most degree $d/2$. Furthermore, the cross-covariance (22c) is computed correctly if f has at most

[6] As stated earlier, we reserve "order" to describe the order of a Taylor series of partial derivative to avoid confusion.

[7] For a vector or matrix function g, the degree is the highest appearing monomial degree. For example, $g(t) = [t_1^3 t_2^2 - 3, t_2]^{\mathrm{T}}$ has degree 5.

degree $d-1$. The above considerations are confirmed by the simulation results in Sec. V-A.

Integration rules can be grouped into product and nonproduct rules [64] that are further explained below.

1) Product rules:

Product rules compute the n-fold integral (63) by applying integration rules to one dimension at a time. For example, the integration with respect to t_1 can be approximated by an M_1-point rule. The result is a sum of M_1 integrals with respect to the remaining components $t_{2:n}$

$$\sum_{i_1=1}^{M_1} w_1^{(i_1)} \int g(t_1^{(i_1)}, t_{2:n}) \exp\left(-\sum_{j=2}^{n} t_j^2\right) dt_{2:n}. \tag{65}$$

Continuing in this way requires $N = \prod_{i=1}^{n} M_i$ points and weights in total, which renders the method unfeasible for all but small n. The overall degree of such an n-dimensional product rule is determined by the smallest M_i. The Kronecker product can be used to compute the overall weights and samples [66].

Gauss-Hermite quadrature rules [48] can be used to generate the M_i weights and points for the one-dimensional integrals from the Hermite polynomials. An M-point Gauss-Hermite rule integrates polynomials of degree $d = 2M-1$ exactly.

In a Kalman filter context, product rules were investigated first in [17] and later in [26, 41]. As pointed out, the required number of points is problematic for larger n. A technique to divide the filtering problem into smaller chunks that can be approached by product rules is given in [67]. The simulations in Sec. V include 3-point and 5-point (per dimension) quadrature rules.

2) Nonproduct rules:

In contrast to product rules, nonproduct rules are tailored formulas to integrate monomials[8] of at most degree d exactly. The number of required point scales much better with n.

An example from the filtering literature is the rule of degree 3 that is used in the cubature Kalman filter (CKF) of [18]. It requires $N = 2n$ weights and sigma points that are given by

$$t^{(\pm i)} = \pm\sqrt{\frac{n}{2}} e_i, \tag{66a}$$

$$w^{(\pm i)} = \frac{\pi^{n/2}}{2n}, \tag{66b}$$

with $i = 1,\ldots,n$ and the ith column e_i of the identity matrix. The selection scheme (66) can be found in as early references as [64, 65]. Using the variable substitution (24) of Sec. III-B the $t^{(i)}$ can be mapped to

$$x^{(\pm i)} = \hat{x} \pm \sqrt{2} P^{1/2} t^{(\pm i)} = \hat{x} \pm \sqrt{n} u_i. \tag{67a}$$

Furthermore, the weighted transformed samples can be written as

$$w^{(\pm i)} g(t^{(\pm i)}) = \frac{\pi^{n/2}}{2n} \frac{1}{\pi^{n/2}} f(x^{(\pm i)}) = \frac{1}{2n} z^{(\pm i)}. \tag{67b}$$

The above (67) discloses that the degree 3 rule corresponds to the UT (60) with $\kappa = 0$. Therefore, $\kappa = 0$ can be regarded as well founded parameter choice in the UT.

The authors of [18] show how to use a change to spherical coordinates (Sec. III-B) and subsequent separation of the integral (63) into a range and a spherical component to arrive at the above expressions. The authors of [26] assume a numerical integration perspective on the UT to arrive at a similar expression.

An extensive numerical integration source is [64], where the above degree-3 rule with $2n$ points, a degree-5 rule with $2n^2+1$ points, and a degree-7 rule with $(4n^3+8n+3)/3$ points are provided. Related Kalman filtering advances are described in [26], including an assessment of the stability factor that relates to numerical accuracy [65]. The recent high-degree cubature Kalman filters of [42] provide arbitrary degree rules developed from the integral representation (28). The number of required points grows polynomially with n, which yields an improvement over the exponentially growing number of points in Gauss-Hermite quadrature. Also, [42] highlights the degree-5 rule of [64] in their discussion and KF simulations.

3) Randomized integration rules:

We conclude this section with a numerical integration approach that combines Monte Carlo integration with deterministic integration rules.

In (28) of Sec. III-B it is shown how the integral (63) can be separated into a range and a spherical part. This variable change is exploited in [68] in a Monte Carlo scheme, and implemented within the Kalman filtering context by [69] as randomized UKF variant and [43] with the numerical integration perspective. The suggested stochastic integration filter [43] uses randomized rules for both the radial and spherical integration, and refines the results in an iterative scheme. Thereby, less sigma points are required compared to a direct Monte Carlo approach. The required number of points N depends on the user parameters, which are set to determine the degree of the radial and spherical integration rules. The approach is further discussed and successfully tested in [70].

An immediate variation of the above is to use a deterministic integration rule for the spherical and Monte Carlo integration for the range component. Such derived rules can be viewed as randomized deterministic integration rules.

D. Sigma Points for Function Approximation

This section summarizes how sigma points can be used to approximate the function f in the moment computation problem (22). Such local approximations

[8] and hence polynomials.

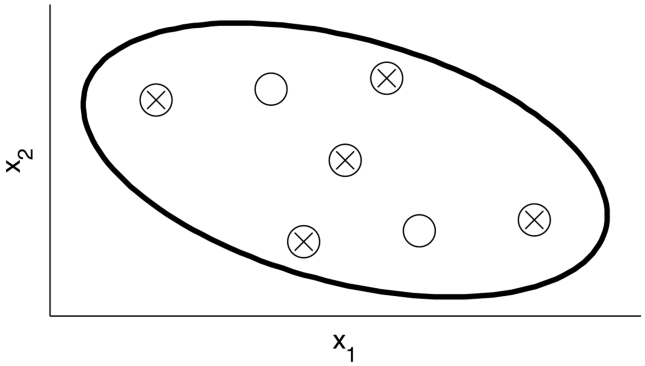


Fig. 2. The extended sigma points of (70) (circles) and the original samples of (68) (crosses) for $n = 2$. The ellipse indicates a contour for which $\mathcal{N}(x;\hat{x},P)$ is constant. The required matrix square root is computed using the singular value decomposition of P.

are in the spirit of linearization or Taylor approximation of f, but with divided differences rather than analytical derivatives.

1) Divided difference methods:

The solutions of the moment computation problem (22) are given by (46) in the case of linear f, and by (54) for quadratic f (51). If we consider a nonlinear f, we might want to mimic these solutions using a local Taylor expansion of f. The required partial derivatives are then evaluated at the expansion point only, and do not take into account the uncertainty in x. If, instead, the local approximation is carried out by an interpolation approach, the interpolation points can be chosen to also reflect the uncertainty in x.

A linearization based on the above idea was first suggested in [19] in the Kalman filtering context, and extended independently to include quadratic approximations by [20] and [17]. Both motivate their development from an interpolation point of view that replaces analytical derivatives with divided differences. The method generates sigma points similar to the UT

$$x^{(0)} = \hat{x}, \tag{68a}$$

$$x^{(\pm i)} = \hat{x} \pm \sqrt{\gamma} u_i, \tag{68b}$$

where $i = 1,\ldots,n$. The u_i are the columns (78) of the square root $P^{1/2}$ (77) (see App. A) and act as perturbation points in the divided differences. The parameter γ determines the length of the interpolation interval.

The sigma points are transformed to yield $z^{(0)}$ and all $z^{(\pm i)}$ and processed as

$$\hat{z} = \frac{\gamma - n}{\gamma} z^{(0)} + \frac{1}{2\gamma} \sum_{i=1}^{n} (z^{(i)} + z^{(-i)}), \tag{69a}$$

$$S = \frac{1}{4\gamma} \sum_{i=1}^{n} (z^{(i)} - z^{(-i)})(z^{(i)} - z^{(-i)})^{\mathrm{T}} + \frac{\gamma - 1}{4\gamma^2} \sum_{i=1}^{n} (z^{(i)} - 2z^{(0)} + z^{(-i)})(z^{(i)} - 2z^{(0)} + z^{(-i)})^{\mathrm{T}}, \tag{69b}$$

$$M = \frac{1}{2\sqrt{\gamma}} \sum_{i=1}^{n} u_i (z^{(i)} - z^{(-i)})^{\mathrm{T}}. \tag{69c}$$

The expressions sacrifice some inconvenient terms for the sake of a faster algorithm. Further details on the omitted terms are provided in [71]. The second order divided difference method from [20] is illustrated in (69). A first order divided difference method [19] is obtained by retaining only the first sum in (69b), and setting $\hat{z} = z^{(0)}$ similar to the Taylor based linearization.

The expression (69a) has the same functional form as the UT mean in (61). A detailed discussion on the relations between divided difference methods and the UT is given in [27].

2) Sigma Point Implementation for the Moments of a Quadratic Function:

The exact solution of the moment computation problem for quadratic f is given by (54). The divided difference solution (69) is an approximation to this in general. If f is indeed quadratic then (69a) and (69c) are exact, but (69b) is not. We here show how an exact S can be computed entirely from sigma points. The material is based on [23] and the extensions in [30, 35].

The sigma points (68) are not sufficient to make S exact. Therefore, the set is extended to include n^2 points

$$x^{(0)} = \hat{x}, \tag{70a}$$

$$x^{(\pm ij)} = \hat{x} \pm \frac{\sqrt{\gamma}}{2}(u_i + u_j), \tag{70b}$$

with $i = 1,\ldots,n$ and $j = 1,\ldots,n$. By construction, the points of (68) are contained in (70): $x^{(\pm ii)} = x^{(\pm i)}$. In addition, $n(n-1)$ unique points are generated. The increased number of points N corresponds to that of a degree-5 integration rule [42, 64]. Fig. 2 illustrates the extended as well as the points of (68) for $n = 2$.

The exact S for quadratic f is given by

$$S = \frac{1}{4\gamma} \sum_{i=1}^{n} (z^{(i)} - z^{(-i)})(z^{(i)} - z^{(-i)})^{\mathrm{T}} + \frac{1}{8\gamma^2} \sum_{i=1}^{n} \sum_{j=1}^{n} (4z^{(ij)} + 4z^{(-ij)} - z^{(i)} - z^{(-i)} - z^{(j)} - z^{(-j)} - 4z^{(0)}) (4z^{(ij)} + 4z^{(-ij)} - z^{(i)} - z^{(-i)} - z^{(j)} - z^{(-j)} - 4z^{(0)})^{\mathrm{T}}. \tag{71}$$

The derivation of the above expression is provided in App. C.

In the Kalman filter context the above leads to a sample implementation of the second order EKF [35].

V. SIMULATION EXPERIMENTS

In this section we provide simulations to illustrate the performance of the moment computation methods of Sec. IV. Insights about differences and advantages are highlighted. Furthermore, it is shown that the methods do not always work as out-of-the-box solutions to any moment computation problem (22).

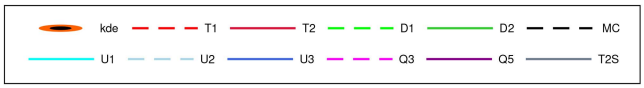


Fig. 3. The legend for all subsequent figures.

TABLE I
List of evaluated methods, their abbreviation, and a reference motivating the selected parameters.

Abbrev.	Method	Parameters
MC	Monte Carlo	$N = 50000$
T1	1st order Taylor [23]	—
T2	2nd order Taylor [23]	—
D1	1st order div. diff. [20]	$\gamma = n$
D2	2nd order div. diff. [20]	$\gamma = n$
U1	Unscented [14]	$\alpha = 1$; $\beta = 0$; $\kappa = 3 - n$
U2	Unscented [21]	$\alpha = 10^{-3}$; $\beta = 2$; $\kappa = 0$
U3	Cubature [18]	$\alpha = 1$; $\beta = 0$; $\kappa = 0$
Q3	3-point Gauss-Hermite [17, 48]	—
Q5	5-point Gauss-Hermite [17, 48]	—
T2S	2nd order Taylor, sigma points [35]	$\gamma = 0.1$

The examples include polynomial functions as well as a radar target tracking problem that occurs in air traffic control. The displayed methods are chosen in favor of a compact description that aims at both new and experienced researchers in the field. Hence, simple rules are favored over the more advanced methods that are mentioned in Sec. IV. The reader is referred to [23, 28, 30, 42] for other comparative simulation studies of moment computation methods in the spirit of our discussion below, and to [21, 26, 27, 29, 70] for comparative studies of the related filtering algorithms.

The investigated moment computation methods are first and second order Taylor approximation using analytical derivatives (T1 and T2, Sec. III-C); first and second order divided difference approximations (D1 and D2, Sec. IV-D.1) and the related second order Taylor approximation using sigma points (T2S, Sec. IV-D.2); Monte Carlo integration (MC, Sec. IV-A); three variations of the unscented transformation with different parameter settings including a degree-3 cubature rule (U1–U3, Sec. IV-B); and 3-point and 5-point (per dimension) Gauss-Hermite quadrature rules (Q3 and Q5, Sec. IV-C) which are more accurate than UT at the expense of more sigma points (3^n and 5^n). Table I summarizes the abbreviations used and the parameters according to Sec. IV. The plot legend for all subsequent illustrations is given in Fig. 3.

A. Polynomial functions

First, two quadratic functions are used to study the moment computation problem (22) with Gaussian input. In this case Sec. III-C describes the exact analytical solution T2 that can be compared to the approximate methods. Second, the composition of the two functions is studied, both in a two stage and single stage approach.

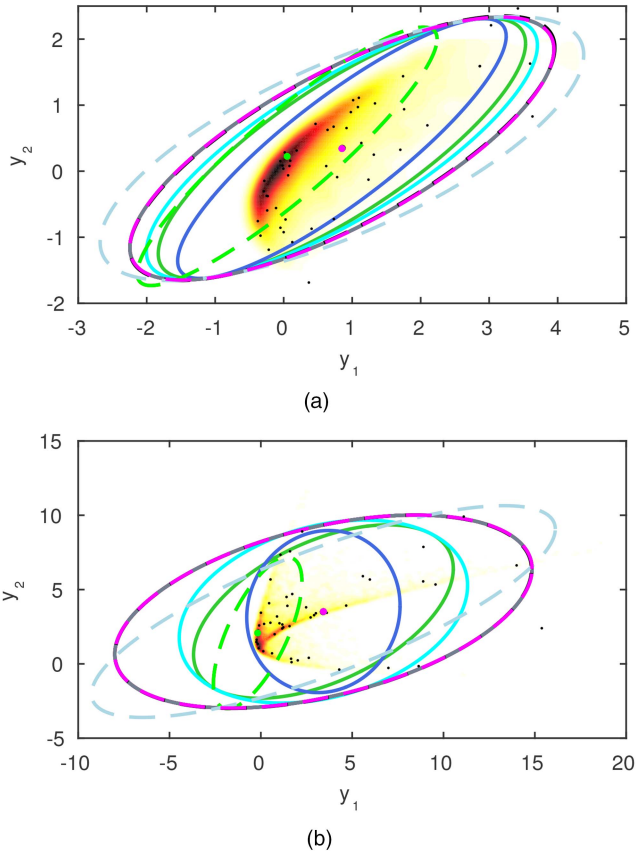


Fig. 4. The result obtained when using different methods to transforming x. The legend for this and subsequent figures is given in Figure 3. (a) $y = f^1(x)$. (b) $y = f^2(x)$.

The two two-dimensional quadratic functions f^1 and f^2 are specified as in (51) with the parameters

$$\begin{bmatrix} a_1^1 \\ a_2^1 \end{bmatrix} = \begin{bmatrix} -0.2 \\ 0 \end{bmatrix}, \quad [b_1^1 \quad b_2^1] = \begin{bmatrix} 0.1 & 0.2 \\ 0.2 & 0.3 \end{bmatrix},$$
$$C_1^1 = \begin{bmatrix} 0.15 & 0 \\ 0 & 0.05 \end{bmatrix}, \quad C_2^1 = \begin{bmatrix} 0.025 & 0.009 \\ 0.003 & 0.005 \end{bmatrix}, \tag{72a}$$

$$\begin{bmatrix} a_1^2 \\ a_2^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1.2 \end{bmatrix}, \quad [b_1^2 \quad b_2^2] = \begin{bmatrix} -0.8 & 0.4 \\ 0.8 & 0.4 \end{bmatrix},$$
$$C_1^2 = \begin{bmatrix} 0.625 & -0.303 \\ -0.303 & 0.275 \end{bmatrix}, \quad C_2^2 = \begin{bmatrix} 0.08 & 0 \\ 0 & 0.28 \end{bmatrix}, \tag{72b}$$

where the superscript is used to distinguish between the two functions. The input is assumed Gaussian $x \sim \mathcal{N}([0,1]^{\mathrm{T}}, 4I)$.

1) Quadratic Transformation:

The results from applying the different methods in Table I to compute the mean and covariance of $f^1(x)$ and $f^2(x)$ are presented in Fig. 4. Illustrated are the ellipses that contain 95 percent of the probability mass of a Gaussian with the computed mean and covariance. We notice that f^1 is less nonlinear than f^2, which

can be observed from the visualization of the correct target distribution which has been obtained as a kernel density estimate from MC samples. Both T2 and T2S by construction provide the exact mean and covariance. With f^1 and f^2 being polynomials of degree 2, the correct mean is recovered by all integration rules of degree 2 and above. This includes the UT variants U1, U2, U3 which can be seen as integration rules of degree 3 [26], but also the algebraically equivalent D2. The 3-point and 5-point (per dimension) quadrature rules with degrees $2 \cdot 3 - 1 = 5$ and $2 \cdot 5 - 1 = 9$, respectively, also manage to capture the correct covariance which requires an integration rule of degree 4 for f^1 and f^2. In contrast to the UT variants ($N = 2n = 4$ or $N = 2n + 1 = 5$), Q3 and Q5 use more sigma points ($N = 3^n = 9$ and $N = 5^n = 25$). Still, with $n = 2$ the computational demands remain low. The MC method with 50 000 samples is very close to the correct solution. Only T1 and D1 with their underlying linearity assumption fail to capture the correct mean and severely underestimate the covariance.

The methods perform mostly well for the quadratic functions, especially for the mildly nonlinear f^1 the approximations are similar, including T1 and D1. For f^2 the differences are more pronounced and the degree 3 rules U1, U3, and D2 do not reflect the true uncertainty ellipse well.

In summary, T2, T2S, Q3, and Q5 are exact for quadratic functions. The UT variants fail to capture the correct covariance with U2 slightly better for the chosen function f^2. D1 and T1 can perform very poorly, even though the nonlinearity is mild in the sense that it is only "one degree above linear." Other experiments with randomly selected quadratic functions support the above findings [30].

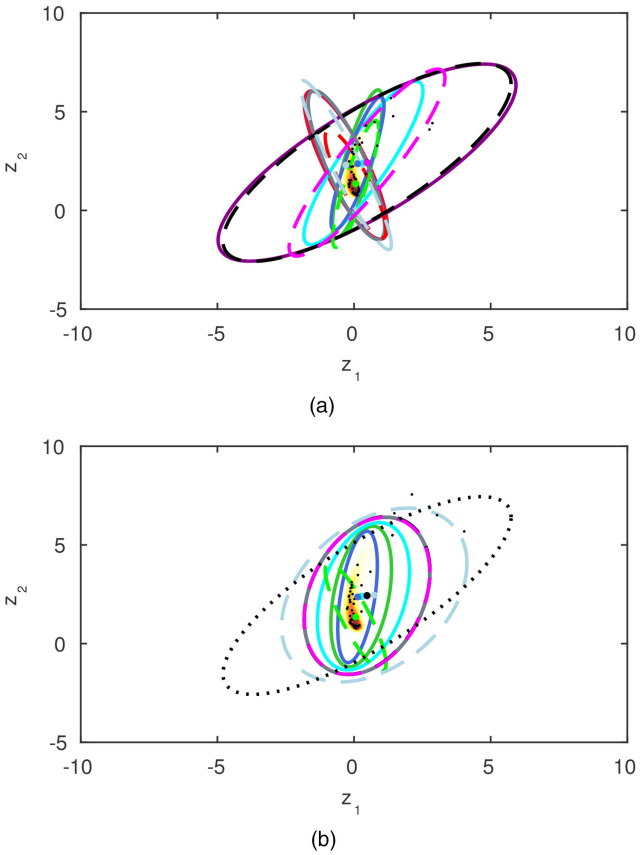


Fig. 5. The result obtained when applying the different approximations to a quartic function. Note that the MC solution is only correct in (a) as an intermediate approximation is performed in (b). Therefore the composed MC estimate has been added as dotted black line in the latter case. (a) Direct function evaluation, $z = f^3(x)$, corresponding to keeping the same sigma points. (b) Sequential function evaluation, $y = f^1(x)$ and $z = f^2(y)$, with intermediate sigma point regeneration.

2) Quartic Transformation:

In a second simulation a quartic function f^3 is constructed by composing f^2 and f^1, $f^3(x) = f^2(f^1(x))$. This allows us to study the selected methods on a higher degree polynomial than covered by most methods in Table I, as well as touching on the interesting questions: *"When considering a series of transformations, is it preferable to keep the same sigma points throughout the approximation, or should new sigma points be selected in each step?"* This has implications to the filtering problem where the time propagation step and measurement prediction are performed in series (Sec. II-D) and for cases where a function can be decomposed into simpler ones.

Fig. 5 shows the result obtained when applying the methods in Table I to the quartic function f^3. The difference between the result from applying the methods directly to the quartic function (Fig. 5(a)) and from performing the approximation in two steps (Fig. 5(b)) is striking. In the former case, only the Q5 rule with its degree 9 is accurate enough to capture the true mean and covariance. The MC method also recovers the true mean and covariance, albeit with some small sampling errors. With degree 5, the rule Q3 still provides the correct mean whereas the remaining methods fail to do so. In the later case, none of the methods (including the MC method) can be expected to recover the true mean and covariance, as a result of the intermediate step where the non-Gaussian distribution of $f^1(x)$ is approximated as a Gaussian, which results in information loss.

The two step computations result in comparable approximations in the sense that all methods underestimate the true covariance (as provided by MC and Q5 in Fig. 5(a) and as dotted line in Fig. 5(b)) and rotate its principal axes. T1 and D1 assert the smallest covariance and put the mean close to the most likely value of the resulting distribution. The similarity among the remaining approximations can be explained with the comparable Gaussian approximations of $f^1(x)$ that all methods provide (Fig. 4(a)). The lack of performance can be explained by the fact that all methods ignore the skewness in the distribution of $f^1(x)$. Note that MC, T2, T2S, Q3, and Q5 are all identical, as they all perfectly capture the two quadratic transformations, but that none of them are correct due to the intermediate approximation.

The result obtained from applying the approximations directly to f^3 yields less comparable results. D1, U1, and D2 provide roughly the same estimates, slightly worse than the sequential results. U2, T2, and T2S seriously underestimates the uncertainty and provide results far from the correct Q5 estimate. Q3 obtains the correct mean but underestimates the covariance. T1 and D1 no longer agree due to the differences between the numerical and analytical derivatives for the degree-4 polynomial.

Overall, we conclude that of the studied methods only Q5 and MC are suitable for the quartic function. Interestingly, there seems to be a gain from performing a two stage procedure where new sigma points are selected when using methods of insufficient accuracy, albeit the intermediate Gaussian approximation. Still, directly computing the moments of $f^3(x)$ with a rule of sufficient degree such as Q5 should be preferred if computationally feasible.

B. Tracking Application

In this second set of examples we consider moment computations problems that occur in target tracking, say, for air traffic control. In our set-up the tracked aircraft are described with a simple state-space model with the state vector

$$x = [\mathsf{x}, \quad \mathsf{y}, \quad \mathsf{s}, \quad \mathsf{h}]^{\mathrm{T}}, \tag{73}$$

where x and y make up the position in Cartesian coordinates [m], and s and h are the speed [m/s] and heading [rad], respectively.

A radar provides measurements $y = h(x) + e$ of the range, r, the bearing, ϕ, and the range rate, $\dot{\mathsf{r}}$, with

$$h(x) = \begin{bmatrix} \mathsf{r} \\ \phi \\ \dot{\mathsf{r}} \end{bmatrix} = \begin{bmatrix} \sqrt{\mathsf{x}^2 + \mathsf{y}^2} \\ \mathrm{atan2}(\mathsf{y}, \mathsf{x}) \\ \frac{\mathsf{s}}{\mathsf{r}}(\mathsf{x}\cos(\mathsf{h}) + \mathsf{y}\sin(\mathsf{h})) \end{bmatrix}, \tag{74}$$

where atan2 is the quadrant compensated arctangent function.

The dynamic model for how the state evolves in time is given by $x_k = f(x_{k-1}, v_{k-1})$, where

$$\begin{bmatrix} \mathsf{x}_k \\ \mathsf{y}_k \\ \mathsf{s}_k \\ \mathsf{h}_k \end{bmatrix} = \begin{bmatrix} \mathsf{x}_{k-1} + \cos(\mathsf{h}_{k-1})(\mathsf{s}_{k-1} + \Delta\mathsf{s}_{k-1}) \\ \mathsf{y}_{k-1} + \sin(\mathsf{h}_{k-1})(\mathsf{s}_{k-1} + \Delta\mathsf{s}_{k-1}) \\ \mathsf{s}_{k-1} + \Delta\mathsf{s}_{k-1} \\ \mathsf{h}_{k-1} + \Delta\mathsf{h}_{k-1} \end{bmatrix} \tag{75}$$

and $v = [\Delta\mathsf{s}, \Delta\mathsf{h}]^{\mathrm{T}}$ is process noise. Such models can be used to describe highly maneuvering targets, for example a flying robot, but also slowly maneuvering aircraft.

One time step and measurement will be studied, with

$$\begin{bmatrix} \mathsf{x}_{k-1} \\ \mathsf{y}_{k-1} \\ \mathsf{s}_{k-1} + \Delta\mathsf{s}_{k-1} \\ \mathsf{h}_{k-1} + \Delta\mathsf{h}_{k-1} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 2500 \\ 4330 \\ 250 \\ -\frac{2\pi}{3} \end{bmatrix}, \begin{bmatrix} 2500 & 0 & 0 & 0 \\ 0 & 2500 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & \frac{900\pi^2}{180^2} \end{bmatrix} \right) \tag{76}$$

in the simulations below. This represents one step in a filter solution to track the target, but retains the focus on moment computation problems. For simplicity, the speed and heading uncertainty for the current time step is assumed to be included in the state already. This has no impact on the end result as they always appear together in (75). The position and speed accuracy has been exaggerated and the heading error is large to make the example more illustrative. Hence, the observed effects can be expected to be less pronounced in practice.

1) Range and Bearing Measurements:

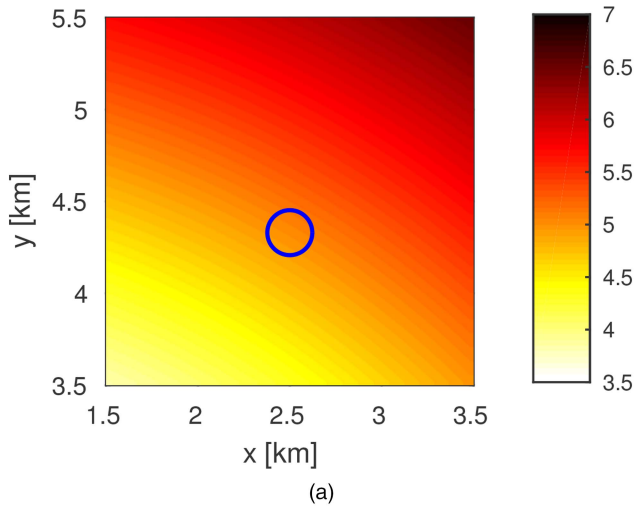
The range and bearing measurements obtained by a radar are mildly nonlinear functions of the position, as shown in the heatmaps of Fig. 6. The state estimate is illustrated with an uncertainty ellipse that contains 95 percent of the probability mass. For the relevant positions, the heatmap appears similar to a linear function. The approximate linearity is confirmed by computing the moments of Taylor polynomials of different degree, as suggested in Sec. III-C. There is hardly any change for degrees above one.

Following the above considerations, it is not surprising to see that all the methods perform equally well. The result is illustrated in Fig. 7, where it is virtually impossible to tell the different methods apart.

2) Range Rate Measurement:

Adding range rate to the measurement vector provides important additional information about the target motion, but does at the same time add a more complex and nonlinear function that depends on all the state components. At this distance, the range rate is almost constant with respect to changes in the position, whereas Fig. 8 illustrates that the range rate is nonlinear (approximately sinusoidal) in the heading angle.

Again, the methods in Table I are used to obtain the mean and variance of the range rate. The result is given in Fig. 9. The methods T2, T2S, D2, U1, U2, U3, Q3, and Q5 are almost perfect matches with the MC method. T1 and D1 provide virtually identical results considerably underestimating the variance, and with a significant bias coinciding with the most likely values of the distribution. Thus, a reasonable choice of method would be U3 which uses the least sigma points.



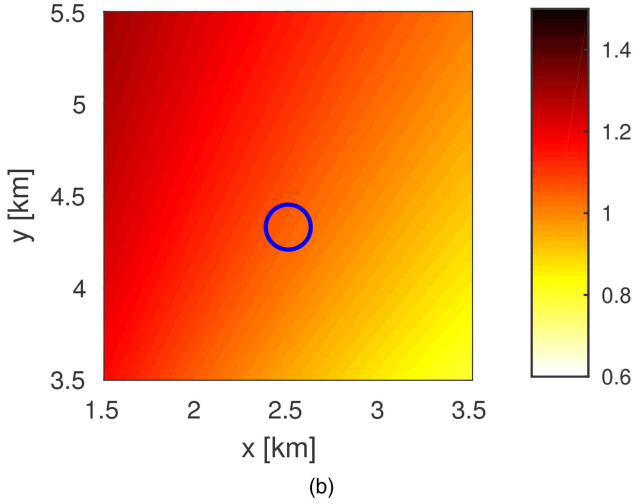


Fig. 6. Illustration of the severity of the radar measurement nonlinearity. Note, the range and bearing ((a) and (b)) are independent from the speed and heading. The blue ellipse represents the state uncertainty. (a) Range as function of target position. (b) Bearing as function of target position.

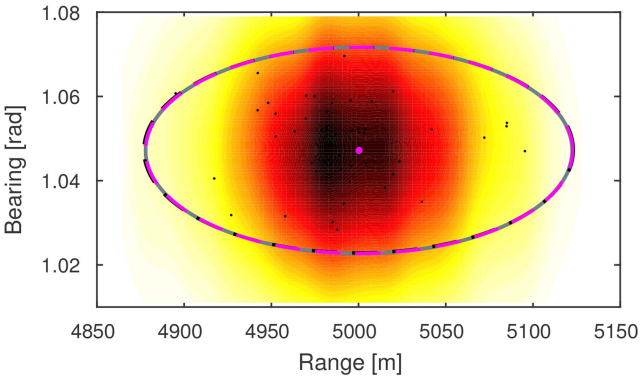


Fig. 7. Result obtained when applying the measurement function to x_{k-1} and visualizing the range and bearing components.

Table II shows the moments of different Taylor polynomials of the range rate that have been computed using the compact representation of Sec. III-C. It can be seen that the mean value changes for every two polynomial degrees added to the approximation. One

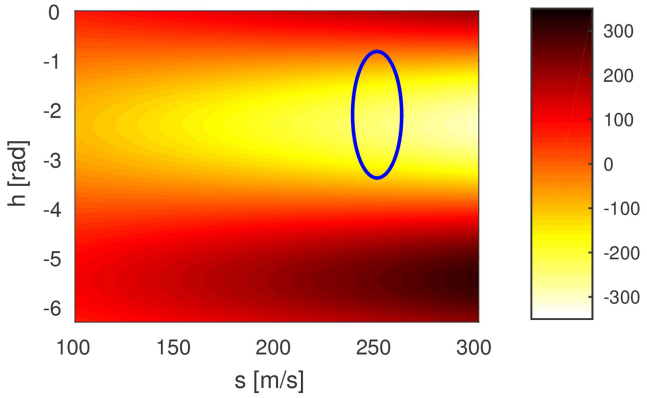


Fig. 8. Range rate as a function of speed and heading of the target when x and y have their mean values. The blue ellipse illustrates the state uncertainty.

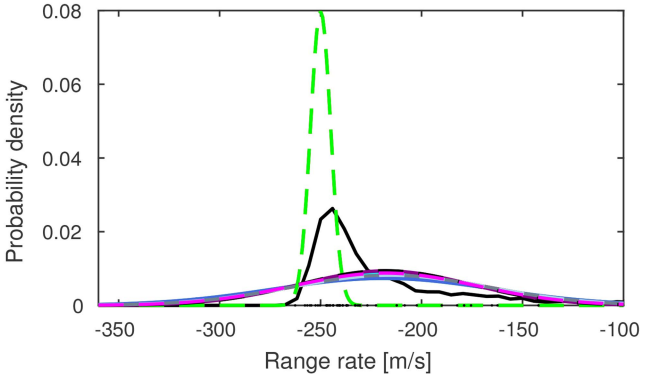


Fig. 9. Illustration of transformed range rate using the different approximations. The black solid line here visualize the true underlying function obtained with MC simulations.

TABLE II
Mean and variance matrix of the range rate based on Taylor expansions of different degrees.

Degree	Mean	Covariance
1	−250	25
2	−215.7	2375.5
3	−215.7	2370.1
4	−218	1784.4
5	−218	1784.5

interpretation of the result is that the range rate is difficult to approximate by a polynomial of low degree for the given input uncertainty.

3) Dynamic Model and Measurement:

In the linear KF it is common to alternate between time updates and measurement updates. As discussed in Sec. II-D, in nonlinear KF the situation is not as clear because the user can either combine the state transition and measurement functions to jointly compute the moments of the predicted state and output; or introduce an intermediate Gaussian approximation for the predicted state and then compute the moments of the predicted output. Also, see the related example in Sec. V-A.2.

Here, the two approaches are compared. The result is presented in Fig. 10. First, notice that the state pre-

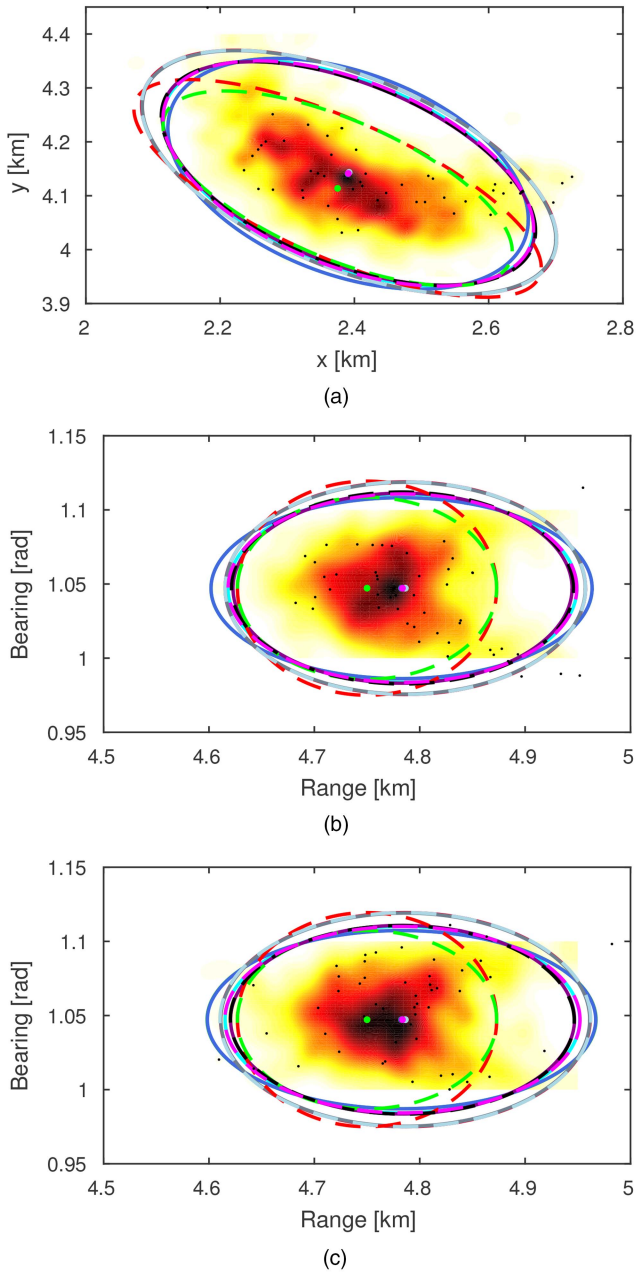


Fig. 10. Illustration of separated and composed transformation from x_{k-1} to y_k. Note that the MC solution is only correct in (c) as an intermediate approximation is performed in (b). Therefore the composed MC estimate has been added as dotted black line in the latter case. (a) Time propagation, $x_k = f(x_{k-1})$. (b) Sequential time propagation and measurement prediction, $y_k = h(x_k)$. (c) Combined time propagation and measurement prediction, $y_k = h(f(x_{k-1}))$.

diction results in a slightly banana shaped distribution (Fig. 10(a)), as a result of the uncertain heading but certain speed. Hence, the state transition is only mildly nonlinear. All methods, except for T1 and D1, produce mean and covariance estimates close to the true values. Again, T1 and D1 stand out by underestimating the covariance.

Comparing the result from performing the transformation in two steps (Fig. 10(b)) and in a single step (Fig. 10(c)) it is hard to tell the two approaches apart. In both cases the true distribution is well approximated by Q3, Q5, T2, T2S, D2, U1, U2, and U3, whereas T1 and D1 provide worse results with bias and too small covariances.

A conclusion from this experiment, supported by the experiences in Sec. V-A.2, is that it is not obvious whether to keep the sigma points between the two steps or to generate new ones. With the almost linear range and bearing measurements it made no difference. Furthermore, our results confirm the widespread view that T1 and D1 tend to introduce biases and to underestimate the covariance due to their underlying linearity assumptions.

VI. CONCLUDING REMARKS

In this tutorial we have discussed the nonlinear Kalman filter solution to the Bayesian filtering problem, with a focus on sigma point filters. It was shown that the central challenge in all Kalman filter algorithms is the computation of the mean values and the covariance matrices of nonlinearly transformed random variables. These moments are then used in the measurement update. This unifying view facilitates an easier access to the filtering literature for researchers who are new to the field, and provides additional (perhaps subtle) insights for the more experienced audience.

The underlying moment computation problem was discussed in greater detail including its structural properties, common reformulations, and even a solution in terms of the Taylor series. It is hoped that the presentation will facilitate a greater understanding of existing methods as well as catalyze new methods with a solid theoretical foundation.

The presentation of sigma point based moment computation methods which are motivated from the different concepts of density approximation (UT), numerical integration, and interpolation has shown significant similarities between the approaches. Noticing the interrelations is beneficial, for example in the case of the degree of integration rules and the accuracy statements that follow for the UT.

The simulations have shown that for mildly nonlinear functions all approaches perform well. It was seen that methods that are based on analytical Taylor expansion work well if the function is well approximated by polynomials of the assumed degree. However, if the degree of the approximation is too low, then the performance can deteriorate. The investigated UT variants (including the cubature setup) showed solid performance throughout the examples, but not without failure on some. Similar statements can be made for the divided difference approaches. Both failed for one of the investigated quadratic functions. Two more accurate Gauss-Hermite integration rules were shown to perform well, but at the expense of more sigma points which prohibits their use for high-dimensional problems.

We conclude with the insight that for moment computation problems, it is important to understand the addressed nonlinearities, to understand the type of problems that can be accurately handled, and to understand the limitations of sigma point methods. This is especially crucial in nonlinear Kalman filters which are based on the chosen moment computation methods.

APPENDIX

A. About Matrix Square Roots

Any positive definite $n \times n$ matrix, in particular the covariance P of $x \sim \mathcal{N}(\hat{x}, P)$, can be factored as

$$P^{1/2} P^{\mathrm{T}/2} = P \tag{77}$$

with an $n \times n$ matrix square root $P^{1/2}$.

The square root is not uniquely defined because any $P^{1/2}U$ with orthogonal U is also a matrix square root for P.

Matrix square roots can be computed using different algorithms [72]. The Cholesky factorization is a fast method for computing a triangular $P^{1/2}$. The singular value decomposition enjoys desirable numerical properties but is computationally more demanding.

The ith column of $P^{1/2}$ is given by

$$u_i = P^{1/2} e_i, \tag{78}$$

where e_i is the ith basis vector in the standard basis of $\mathbb{R}^n$.

From the columns of $P^{1/2}$

$$P = \sum_{i=1}^{n} u_i u_i^{\mathrm{T}} \tag{79}$$

can be re-constructed.

B. Nested Covariance Computation for Partitioned x

Expression (38b) of Sec. III-B can be derived as follows:

$$\begin{aligned}
\operatorname{cov}\{z\} &= \mathrm{E}\{zz^{\mathrm{T}}\} - \mathrm{E}\{z\}\mathrm{E}\{z\}^{\mathrm{T}} \\
&= \mathrm{E}\{\mathrm{E}\{zz^{\mathrm{T}} \mid x_2\}\} \\
&\quad - \mathrm{E}\{\mathrm{E}\{z \mid x_2\}\}\mathrm{E}\{\mathrm{E}\{z \mid x_2\}\}^{\mathrm{T}} \\
&= \mathrm{E}\{\operatorname{cov}\{z \mid x_2\}\} \\
&\quad + \mathrm{E}\{\mathrm{E}\{z \mid x_2\}\mathrm{E}\{z \mid x_2\}^{\mathrm{T}}\} \\
&\quad - \mathrm{E}\{\mathrm{E}\{z \mid x_2\}\}\mathrm{E}\{\mathrm{E}\{z \mid x_2\}\}^{\mathrm{T}} \\
&= \mathrm{E}\{\operatorname{cov}\{z \mid x_2\}\} + \operatorname{cov}\{\mathrm{E}\{z \mid x_2\}\}.
\end{aligned}$$

C. Sigma Point Implementation of the Exact Covariance for Quadratic f

The function f is quadratic. Each row f_k can be expressed by its Taylor expansion (53) with the Jacobian $J_k(\hat{x})$ and the Hessian matrix H_k. All J_k compose the Jacobian matrix J. The dependence on $\hat{x}$ is omitted for convenience. For the following discussion we discuss the first two rows f_1 and f_2.

First, we use the sigma points in (68) to derive the Jacobian term of (54b). Using (53), the transformed points can be written as

$$z_1^{(\pm i)} = f_1(\hat{x}) \pm \sqrt{\gamma} J_1 u_i + \frac{\gamma}{2} u_i^{\mathrm{T}} H_1 u_i. \tag{80}$$

For quadratic f this is not an approximation but exact for any γ. After pairwise processing and rearranging, we can isolate the Jacobian

$$z^{(i)} - z^{(-i)} = 2\sqrt{\gamma} J u_i, \tag{81a}$$

$$J u_i = \frac{1}{2\sqrt{\gamma}} (z^{(i)} - z^{(-i)}). \tag{81b}$$

A summation over all Ju_i together with the result (79) gives

$$\begin{aligned}
JPJ^{\mathrm{T}} &= \sum_{i=1}^{n} J u_i u_i^{\mathrm{T}} J^{\mathrm{T}} \\
&= \frac{1}{4\gamma} \sum_{i=1}^{n} (z^{(i)} - z^{(-i)})(z^{(i)} - z^{(-i)})^{\mathrm{T}},
\end{aligned} \tag{82}$$

the desired first term of (54b).

Next, we address the trace term of (54b) which for f_1 and f_2 can be written as

$$\begin{aligned}
\operatorname{tr}(H_1 P H_2 P) &= \sum_{i=1}^{n} \sum_{j=1}^{n} \operatorname{tr}(H_1 u_i u_i^{\mathrm{T}} H_2 u_j u_j^{\mathrm{T}}) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{n} u_j^{\mathrm{T}} H_1 u_i u_i^{\mathrm{T}} H_2 u_j,
\end{aligned} \tag{83}$$

a sum of n^2 terms. One term in the sum is given by

$$u_j^{\mathrm{T}} H_1 u_i u_i^{\mathrm{T}} H_2 u_j. \tag{84}$$

For $i = j$, we can extract (84) from (80)

$$z_1^{(i)} + z_1^{(-i)} = 2 f_1(\hat{x}) + \gamma u_i^{\mathrm{T}} H_1 u_i, \tag{85a}$$

$$u_i^{\mathrm{T}} H_1 u_i = \frac{1}{\gamma} (z_1^{(i)} - 2 z_1^{(0)} + z_1^{(-i)}) \tag{85b}$$

and

$$\begin{aligned}
&u_i^{\mathrm{T}} H_1 u_i u_i^{\mathrm{T}} H_2 u_i \\
&\quad = \frac{1}{\gamma^2} (z_1^{(i)} - 2 z_1^{(0)} + z_1^{(-i)})(z_2^{(i)} - 2 z_2^{(0)} + z_2^{(-i)}).
\end{aligned} \tag{85c}$$

That is as far as we can get with the original sigma points (68). From here we start working with the extended sigma points (70) which are passed through f to yield

$$\begin{aligned}
z_1^{(\pm ij)} &= f_1(\hat{x}) \pm \frac{\sqrt{\gamma}}{2} J_1 (u_i + u_j) \\
&\quad + \frac{\gamma}{8} (u_i + u_j)^{\mathrm{T}} H_1 (u_i + u_j).
\end{aligned} \tag{86}$$

Again, we can isolate the Hessian term by a combination of transformed samples similar to (85b). The resulting expression

$$\begin{aligned} z_1^{(ij)} - 2z_1^{(0)} + z_1^{(-ij)} \\ &= \frac{\gamma}{4}(u_i + u_j)^{\mathrm{T}} H_1 (u_i + u_j) \\ &= \frac{\gamma}{4}(u_i^{\mathrm{T}} H_1 u_i + u_j^{\mathrm{T}} H_1 u_j + 2u_j^{\mathrm{T}} H_1 u_i) \end{aligned} \tag{87}$$

involves a mixed j,i-term that we require for computing (84) and also homogeneous terms (85c). We can now isolate the mixed term $2u_j^{\mathrm{T}} H_1 u_i$ of (87) by subtracting expressions of the form (85b):

$$\begin{aligned} 2u_j^{\mathrm{T}} H_1 u_i = \frac{4}{\gamma}(z_1^{(ij)} - 2z_1^{(0)} + z_1^{(-ij)}) \\ - \frac{1}{\gamma}(z_1^{(i)} - 2z_1^{(0)} + z_1^{(-i)}) \\ - \frac{1}{\gamma}(z_1^{(j)} - 2z_1^{(0)} + z_1^{(-j)}). \end{aligned} \tag{88}$$

Simplification yields

$$\begin{aligned} u_j^{\mathrm{T}} H_1 u_i = \frac{1}{2\gamma}(4z_1^{(ij)} + 4z_1^{(-ij)} \\ - z_1^{(i)} - z_1^{(-i)} - z_1^{(j)} - z_1^{(-j)} - 4z_1^{(0)}). \end{aligned} \tag{89}$$

Next, (84) can be written as product

$$\begin{aligned} u_j^{\mathrm{T}} H_1 u_i u_i^{\mathrm{T}} H_2 u_j = \frac{1}{4\gamma^2} \\ \times (4z_1^{(ij)} + 4z_1^{(-ij)} - z_1^{(i)} - z_1^{(-i)} - z_1^{(j)} - z_1^{(-j)} - 4z_1^{(0)}) \\ \times (4z_2^{(ij)} + 4z_2^{(-ij)} - z_2^{(i)} - z_2^{(-i)} - z_2^{(j)} - z_2^{(-j)} - 4z_2^{(0)}). \end{aligned} \tag{90}$$

Summation over i and j from 1 to n yields the trace term of (83) and hence the second term of (54b).

REFERENCES

[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan
Estimation with Applications to Tracking and Navigation.
Wiley-Interscience, Jun. 2001.

[2] F. Gustafsson
Statistical Sensor Fusion.
Studentlitteratur AB, Mar. 2010.

[3] Y. Zheng and M. Hasegawa-Johnson
Particle filtering approach to Bayesian formant tracking,
in *IEEE Workshop on Statistical Signal Processing*, Oct. 2003, pp. 601–604.

[4] A. H. Jazwinski
Stochastic Processes and Filtering Theory.
Academic Press, Mar. 1970.

[5] R. E. Kalman
A new approach to linear filtering and prediction problems,
Journal of basic Engineering, vol. 82, no. 1, pp. 35–45, Mar. 1960.

[6] B. D. O. Anderson and J. B. Moore
Optimal Filtering.
Prentice Hall, Jun. 1979.

[7] M. Grewal and A. Andrews
Applications of Kalman filtering in aerospace 1960 to the present,
IEEE Control Systems Magazine, vol. 30, no. 3, pp. 69–78, 2010.

[8] M. Athans, R. Wishner, and A. Bertolini
Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements,
IEEE Transactions on Automatic Control, vol. 13, no. 5, pp. 504–514, 1968.

[9] H. W. Sorenson and A. R. Stubberud
Recursive filtering for systems with small but non-negligible non-linearities,
International Journal of Control, vol. 7, no. 3, pp. 271–280, 1968.

[10] H. W. Sorenson
On the development of practical nonlinear filters,
Information Sciences, vol. 7, pp. 253–270, 1974.

[11] A. Gelb
Applied Optimal Estimation.
MIT Press, May 1974.

[12] N. J. Gordon, D. J. Salmond, and A. F. M. Smith
Novel approach to non-linear/non-Gaussian Bayesian state estimation,
IEE Proceedings-F Radar and Signal Processing, vol. 140, no. 2, pp. 107–113, Apr. 1993.

[13] F. Gustafsson
Particle filter theory and practice with positioning applications,
IEEE Aerospace and Electronics Systems Magazine, vol. 25, no. 7, pp. 53–82, Jul. 2010.

[14] S. Julier, J. Uhlmann, and H. Durrant-Whyte
A new approach for filtering nonlinear systems,
in *Proceedings of the American Control Conference*, vol. 3, 1995, pp. 1628–1632.

[15] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte
A new method for the nonlinear transformation of means and covariances in filters and estimators,
IEEE Transactions on Automatic Control, vol. 45, no. 3, pp. 477–482, Mar. 2000.

[16] S. J. Julier and J. K. Uhlmann
Unscented filtering and nonlinear estimation,
Proceedings of the IEEE, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[17] K. Ito and K. Xiong
Gaussian filters for nonlinear filtering problems,
IEEE Transactions on Automatic Control, vol. 45, no. 5, pp. 910–927, May 2000.

[18] I. Arasaratnam and S. Haykin
Cubature Kalman filters,
IEEE Transactions on Automatic Control, vol. 54, no. 6, pp. 1254–1269, Jun. 2009.

[19] T. S. Schei
A finite-difference method for linearization in nonlinear estimation algorithms,
Automatica, vol. 33, no. 11, pp. 2053–2058, Nov. 1997.

[20] M. Nørgaard, N. K. Poulsen, and O. Ravn
New developments in state estimation for nonlinear systems,
Automatica, vol. 36, no. 11, pp. 1627–1638, Nov. 2000.

[21] R. 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, OR, USA, Apr. 2004.

[22] S. Särkkä
Bayesian Filtering and Smoothing.
New York: Cambridge University Press, Oct. 2013.

[23] F. Gustafsson and G. Hendeby
Some relations between extended and unscented Kalman filters,
IEEE Transactions on Signal Processing, vol. 60, no. 2, pp. 545–555, Feb. 2012.

[24] M. Evans and T. Swartz
Methods for approximating integrals in statistics with special emphasis on Bayesian integration problems,
Statistical Science, vol. 10, no. 3, pp. 254–272, Aug. 1995.

[25] ———
Approximating Integrals via Monte Carlo and Deterministic Methods, 1st ed.
New York: Oxford University Press, May 2000.

[26] Y. Wu, D. Hu, M. Wu, and X. Hu
A numerical-integration perspective on Gaussian filters,
IEEE Transactions on Signal Processing, vol. 54, no. 8, pp. 2910–2921, Aug. 2006.

[27] M. Šimandl and J. Duník
Derivative-free estimation methods: New results and performance analysis,
Automatica, vol. 45, no. 7, pp. 1749–1757, Jul. 2009.

[28] P. Stano, Z. Lendek, J. Braaksma, R. Babuska, C. de Keizer, and A. den Dekker
Parametric Bayesian filters for nonlinear stochastic dynamical systems: A survey,
IEEE Transactions on Cybernetics, vol. 43, no. 6, pp. 1607–1624, Dec. 2013.

[29] T. Lefebvre, H. Bruyninckx, and J. De Schutter
Kalman filters for non-linear systems: a comparison of performance,
International Journal of Control, vol. 77, no. 7, pp. 639–653, 2004.

[30] M. Roth
"Kalman filters for nonlinear systems and heavy-tailed noise,"
Licentiate thesis, Linköping University, Linköping, Sweden, Sep. 2013. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-97544.

[31] O. Cappé, E. Moulines, and T. Ryden
Inference in Hidden Markov Models.
New York, NY: Springer, Dec. 2010.

[32] P. S. Maybeck
Stochastic Models, Estimation, and Control: Volume 2.
Academic Press, Jun. 1982.

[33] T. M. Cover and J. A. Thomas
Elements of information theory, 2nd ed.
Hoboken, N.J.: Wiley-Interscience, 2006.

[34] C. M. Bishop
Pattern Recognition and Machine Learning.
Springer, Aug. 2006.

[35] M. Roth and F. Gustafsson
An efficient implementation of the second order extended Kalman filter,
in *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, Jul. 2011.

[36] J. Sarmavuori and S. Särkkä
Fourier-Hermite Kalman filter,
IEEE Transactions on Automatic Control, vol. 57, no. 6, pp. 1511–1515, 2012.

[37] F. Sandblom and L. Svensson
Moment estimation using a marginalized transform,
IEEE Transactions on Signal Processing, vol. 60, no. 12, pp. 6138–6150, 2012.

[38] M. F. Huber
Chebyshev polynomial Kalman filter,
Digital Signal Processing, vol. 23, no. 5, pp. 1620–1629, Sep. 2013.

[39] J. Steinbring and U. D. Hanebeck
LRKF revisited—the smart sampling Kalman filter (S2KF),
Journal of Advances in Information Fusion, vol. 9, no. 2, pp. 106–123, Dec. 2014.

[40] H. Tanizaki
Nonlinear Filters: Estimation and Applications, 2nd ed.
Springer, Aug. 1996.

[41] I. Arasaratnam, S. Haykin, and R. J. Elliott
Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature,
Proceedings of the IEEE, vol. 95, no. 5, pp. 953–977, 2007.

[42] B. Jia, M. Xin, and Y. Cheng
High-degree cubature Kalman filter,
Automatica, vol. 49, no. 2, pp. 510–518, Feb. 2013.

[43] J. Dunik, O. Straka, and M. Simandl
Stochastic integration filter,
IEEE Transactions on Automatic Control, vol. 58, no. 6, pp. 1561–1566, 2013.

[44] J. Dunik, M. Simandl, and O. 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.

[45] S. Blackman and R. Popoli
Design and Analysis of Modern Tracking Systems.
Artech House, Aug. 1999.

[46] G. Hendeby, R. Karlsson, and F. Gustafsson
The Rao-Blackwellized particle filter: A filter bank implementation,
EURASIP Journal on Advances in Signal Processing, vol. 2010, no. 1, Dec. 2010.

[47] A. Gut
An Intermediate Course in Probability, 2nd ed.
Springer, Jun. 2009.

[48] M. Abramowitz and I. A. Stegun
Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables.
Dover publications, 1965, vol. 55.

[49] T. W. Anderson
An Introduction to Multivariate Statistical Analysis, 3rd ed.
Wiley-Interscience, 2003.

[50] K.-T. Fang, S. Kotz, and K. W. Ng
Symmetric Multivariate and Related Distributions.
Chapman and Hall/CRC, Nov. 1989.

[51] M. Roth, E. Özkan, and F. Gustafsson
A Student's t filter for heavy tailed process and measurement noise,
in *38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.

[52] S. Saha, P. K. Mandal, Y. Boers, H. Driessen, and A. Bagchi
Gaussian proposal density using moment matching in SMC methods,
Statistics and Computing, vol. 19, no. 2, pp. 203–208, Jun. 2009.

[53] T. Kollo
Advanced Multivariate Statistics with Matrices.
Dordrecht, The Netherlands: Springer, Aug. 2005.

[54] B. Holmquist
Moments and cumulants of the multivariate normal distribution,
Stochastic Analysis and Applications, vol. 6, no. 3, pp. 273–278, Jan. 1988.

[55] J. E. Chacón and T. Duong
Multivariate plug-in bandwidth selection with unconstrained pilot bandwidth matrices,
TEST, vol. 19, no. 2, pp. 375–398, Aug. 2010.

[56] D. Manolakis
Kronecker product based second order approximation of mean value and covariance matrix in nonlinear transformations,
IEEE Signal Processing Letters, vol. 18, no. 1, pp. 43–46, Jan. 2011.

[57] C. P. Robert and G. Casella
Monte Carlo Statistical Methods, 2nd ed., ser. Springer Texts in Statistics.
Springer, 2004.

[58] D. J. C. MacKay
Information Theory, Inference and Learning Algorithms, 1st ed.
Cambridge, UK: Cambridge University Press, Oct. 2003.

[59] M. Roth, C. Fritsche, G. Hendeby, and F. Gustafsson
The ensemble Kalman filter and its relations to other nonlinear filters,
in *European Signal Processing Conference 2015 (EUSIPCO 2015)*, Nice, France, Aug. 2015.

[60] U. D. Hanebeck, M. F. Huber, and V. Klumpp
Dirac mixture approximation of multivariate Gaussian densities,
in *Proceedings of the 48th IEEE Conference on Decision and Control*, 2009, pp. 3851–3858.

[61] R. Gray and D. Neuhoff
Quantization,
IEEE Transactions on Information Theory, vol. 44, no. 6, pp. 2325–2383, 1998.

[62] S. J. Julier
The scaled unscented transformation,
in *Proceedings of the American Control Conference*, vol. 6, 2002, pp. 4555–4559.

[63] T. Lefebvre, H. Bruyninckx, and J. De Schutter
Comment on "A new method for the nonlinear transformation of means and covariances in filters and estimators" [with authors' reply],
IEEE Transactions on Automatic Control, vol. 47, no. 8, pp. 1406–1409, Aug. 2002.

[64] A. H. Stroud
Approximate Calculation of Multiple Integrals.
Prentice-Hall, 1971.

[65] A. H. Stroud and D. Secrest
Approximate integration formulas for certain spherically symmetric regions,
Mathematics of Computation, vol. 17, no. 82, pp. 105–135, Apr. 1963.

[66] C. F. Van Loan
The ubiquitous Kronecker product,
Journal of Computational and Applied Mathematics, vol. 123, no. 1–2, pp. 85–100, Nov. 2000.

[67] P. Closas, C. Fernandez-Prades, and J. Vila-Valls
Multiple quadrature Kalman filtering,
IEEE Transactions on Signal Processing, vol. 60, no. 12, pp. 6125–6137, 2012.

[68] A. Genz and J. Monahan
Stochastic integration rules for infinite regions,
SIAM Journal on Scientific Computing, vol. 19, no. 2, pp. 426–439, Mar. 1998.

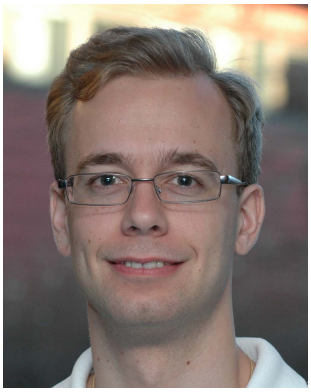
[69] O. Straka, J. Dunik, and M. Simandl
Randomized unscented Kalman filter in target tracking,
in *15th International Conference on Information Fusion*, Jul. 2012, pp. 503–510.

[70] J. Dunik, O. Straka, M. Simandl, and E. Blasch
Random-point-based filters: analysis and comparison in target tracking,
IEEE Transactions on Aerospace and Electronic Systems, vol. 51, no. 2, pp. 1403–1421, Apr. 2015.

[71] M. Nørgaard, N. K. Poulsen, and O. Ravn
"Advances in derivative-free state estimation for nonlinear systems,"
Department of Mathematical Modelling, DTU, Lyngby, Denmark, Tech. Rep. IMM-REP-1998-15, Apr. 2000.

[72] G. H. Golub and C. F. Van Loan
Matrix Computations, 3rd ed.
Baltimore: Johns Hopkins University Press, Oct. 1996.

Michael Roth is a Ph.D. student in the Division of Automatic Control, Department of Electrical Engineering, Linköping University, since 2010. He received his Master's degree (Diplom-Ingenieur) in Electrical Engineering from TU Berlin in 2010. His main research is on algorithms for filtering and smoothing in non-linear, non-Gaussian, and high-dimensional state space models. Moreover, he has been involved in several projects about system identification, control, and signal processing for medical applications.



Gustaf Hendeby is Associate Professor in the Division of Automatic Control, Department of Electrical Engineering, Linköping University. He received his M.Sc. in Applied Physics and Electrical Engineering in 2002 and his Ph.D. in Automatic Control in 2008, both from Linköping University. He worked as Senior Researcher at the German Research Center for Artificial Intelligence (DFKI) 2009–2011, and Senior Scientist at Swedish Defense Research Agency (FOI) and held an adjunct Associate Professor position at Linköping University 2011–2015. Dr. Hendeby's main research interests are stochastic signal processing and sensor fusion with applications to nonlinear problems, target tracking, and simultaneous localization and mapping (SLAM). He has experience of both theoretical analysis as well as implementation aspects.

Fredrik Gustafsson is professor in Sensor Informatics at the Department of Electrical Engineering, Linköping University, since 2005. He received the M.Sc. degree in electrical engineering in 1988 and the Ph.D. degree in Automatic Control in 1992, both from Linköping University. During 1992–1999 he held various positions in automatic control, and 1999–2005 he had a professorship in Communication Systems. His research interests are in stochastic signal processing, adaptive filtering and change detection, with applications to communication, vehicular, airborne, and audio systems. He is a co-founder of the companies NIRA Dynamics (automotive safety systems), Softube (audio effects) and SenionLab (indoor positioning systems).

He was an associate editor for *IEEE Transactions of Signal Processing* 2000–2006, *IEEE Transactions on Aerospace and Electronic Systems* 2010–2012, and EURASIP *Journal on Applied Signal Processing* 2007–2012. He was awarded the Arnberg prize by the Royal Swedish Academy of Science (KVA) 2004, elected member of the Royal Academy of Engineering Sciences (IVA) 2007, and elevated to IEEE Fellow 2011. He was awarded the Harry Rowe Mimno Award 2011 for the tutorial "Particle Filter Theory and Practice with Positioning Applications," which was published in the *AESS Magazine* in July 2010, and was co-author of "Smoothed state estimates under abrupt changes using sum-of-norms regularization" that received the Automatica paper prize in 2014.

The Smart Sampling Kalman Filter with Symmetric Samples

JANNIK STEINBRING
MARTIN PANDER
UWE D. HANEBECK

Nonlinear Kalman Filters (KFs) are powerful and widely-used techniques when trying to estimate the hidden state of a stochastic nonlinear dynamic system. A novel sample-based KF is the Smart Sampling Kalman Filter (S^2KF). It is based on deterministic Gaussian samples which are obtained from an offline optimization procedure. Although this sampling technique is quite effective, it does not preserve the point symmetry of the Gaussian distribution. In this paper, we overcome this issue by extending the S^2KF with a new point-symmetric Gaussian sampling scheme to improve its estimation quality. Moreover, we also improve the numerical stability of the sample computation. This allows us to accurately approximate thousand-dimensional Gaussian distributions using tens of thousands of optimally placed and equally weighted samples. We evaluate the new symmetric S^2KF by computing higher-order moments of standard normal distributions and investigate the estimation quality of the S^2KF when dealing with symmetric measurement equations. Additionally, extended object tracking based on many measurements per time step is considered. This high-dimensional estimation problem shows the advantage of the S^2KF being able to use an arbitrary number of samples independent of the state dimension, in contrast to other state-of-the-art sample-based Kalman Filters. Finally, other estimators also relying on the S^2KF's Gaussian sampling technique, e.g., the Progressive Gaussian Filter (PGF), will benefit from the new point-symmetric sampling as well.

Manuscript received May 18, 2015; revised November 24, 2015; released for publication February 26, 2016.

Refereeing of this contribution was handled by Ondrej Straka.

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

I. INTRODUCTION

Estimating the hidden state of a stochastic dynamic system based on noisy measurements is crucial for many applications in control, object tracking, or robotics. When considering linear systems corrupted by additive Gaussian noise, the Kalman Filter (KF) is the optimal estimator with respect to the mean square error [1]. Unfortunately, most practical problems are nonlinear, making closed-form solutions intractable. Consequently, approximative approaches have to be used. Particle Filters (PFs) [2]–[5] try to approximate the complete, in general multimodal, system state density with a set of weighted particles. This comes at the cost of computational complexity due to the curse of dimensionality. Another problem is sample degeneracy, in particular for high-dimensional state spaces, as a consequence of the particle reweighting using the likelihood function. To reduce computational complexity and circumvent the problem of sample degeneracy, the Progressive Gaussian Filter (PGF) [6], [7] approximates the system state as a Gaussian and moves the particles automatically to the important regions of the state space. Nevertheless, those nonlinear filters are still costly compared to linear filters applied to nonlinear problems.

The Extended Kalman Filter (EKF) explicitly linearizes the underlying models around the current state estimate to be able to apply the standard KF to the considered problem [8]. Iterated variants of the EKF (IEKF) try to improve the EKF approach by iteratively searching for a more suitable point for the model linearization [8]. A more suitable way of model linearization is based on statistical linearization, which can be performed in the best case analytically or, in all other cases, by exploiting samples in the form of Linear Regression Kalman Filters (LRKFs) [9]. LRKFs obtain the required moments by propagating samples through the system and measurement models and computing sample mean and sample covariance matrix, respectively. The most commonly used LRKF is the Unscented Kalman Filter (UKF) [10]. Its samples are, however, limited in number and placement, and several attempts exist to improve the UKF by finding its optimal parameter settings for specific estimation problems [11]. Nevertheless, the additional computational time required to find proper UKF parameters can be used instead to propagate more carefully chosen samples through the models in order to improve the estimation quality. For example, the Gauss-Hermite Kalman Filter (GHKF) introduced in [12] is based on the Gauss-Hermite quadrature rule to generate its samples. Unfortunately, the GHKF also suffers from the curse of dimensionality, and hence, is not well suited for larger state spaces. The fifth-degree Cubature Kalman Filter (CKF) [13] relies on a fifth-degree spherical-radial integration rule to construct its samples. However, by design, the number of samples still grows quadratically in the state dimension making the fifth-degree CKF computational

burdensome when dealing with larger state spaces. A non-deterministic LRKF was proposed with the Randomized UKF (RUKF) [14], [15]. Here, an arbitrary number of randomly scaled and rotated UKF sample sets are combined to a single set of samples. On the one hand this has the advantage of being able to change the employed number of samples. On the other hand it prohibits a reproducible filter behavior and imposes an additional runtime overhead compared to other LRKFs due to the creation of several random orthogonal matrices per prediction and measurement update. The estimation quality of any LRKF, regardless of the sampling it is based on, can be improved by using the iterated statistical linearization approach [16], [17]. A more detailed overview of linear filters and LRKFs can be found in [18], [19].

Recently, the Smart Sampling Kalman Filter (S^2KF) was proposed in [19], [20], and already successfully used for Simultaneous Localization and Mapping (SLAM) in [21]. The S^2KF uses optimal deterministic sampling of a standard normal distribution comprising an arbitrary number of equally weighted samples based on a combination of the Localized Cumulative Distribution (LCD) and a modified Cramér-von Mises distance [22], [23]. The same LCD approach was also extended to approximate arbitrary Gaussian mixture distributions [24].

In this paper, we improve the numerical stability of the LCD approach when dealing with Gaussian densities and, more importantly, extend the S^2KF with a point-symmetric Gaussian sampling. This new sampling approach offers several benefits. First, it reflects the point symmetry of the Gaussian distribution and allows for matching all odd moments of a standard normal distribution exactly, which results in a more accurate state estimation. In this regard, the S^2KF catches up to state-of-the-art LRKFs as all of them also rely on a point-symmetric sampling scheme. Second, due to the improved numerical stability, it is now possible to compute an optimal approximation of thousand-dimensional standard normal distributions comprising tens of thousands of samples. Third, as a minor benefit, the required number of parameters that have to be optimized is reduced by half. Consequently, the samples can be computed faster. However, this is only a minor improvement as the computation is performed offline.

The remainder of the paper is organized as follows. First, we give an overview of nonlinear Kalman filtering and its transition to LRKFs. After that, in Sec. III, we introduce a new point-symmetric version of the S^2KF. In Sec. IV, we evaluate the symmetric S^2KF by computing higher-order moments of multivariate standard normal distributions, showing the advantage of the new point-symmetric sampling scheme when dealing with symmetric measurement equations, and performing extended object tracking. Finally, conclusions are given in Sec. V.

II. SAMPLE-BASED NONLINEAR KALMAN FILTERING

We consider estimating the hidden state $\underline{x}_k$ of a discrete-time stochastic nonlinear dynamic system, where the system model

$$\underline{x}_k = \underline{a}_k(\underline{x}_{k-1}, \underline{w}_k) \tag{1}$$

describes its temporal evolution.[1] Additionally, we receive noisy measurements $\tilde{\underline{y}}_k$ that are assumed to be generated according to the measurement model

$$\underline{y}_k = \underline{h}_k(\underline{x}_k, \underline{v}_k). \tag{2}$$

Thus, the received measurements $\tilde{\underline{y}}_k$ are realizations of the random variable $\underline{y}_k$. The noise variables $\underline{w}_k$ and $\underline{v}_k$ are assumed to be Gaussian and independent of the system state for all time steps. Their densities are given by

$$f_k^w(\underline{w}_k) = \mathcal{N}(\underline{w}_k; \hat{\underline{w}}_k, \mathbf{C}_k^w)$$

and

$$f_k^v(\underline{v}_k) = \mathcal{N}(\underline{v}_k; \hat{\underline{v}}_k, \mathbf{C}_k^v),$$

where $\hat{\underline{w}}_k$ and $\hat{\underline{v}}_k$ denote the mean vectors, and $\mathbf{C}_k^w$ and $\mathbf{C}_k^v$ the covariance matrices.

Our goal is to determine a state estimate of $\underline{x}_k$ in the form of a conditional state density

$$f_k^e(\underline{x}_k) := f(\underline{x}_k \mid \tilde{\underline{y}}_k, \ldots, \tilde{\underline{y}}_1)$$

recursively over time using Bayesian inference. Such a recursive estimator consists of two parts, namely the prediction step and the filter step. On the one hand, the prediction step propagates the state estimate $f_{k-1}^e(\underline{x}_{k-1})$ from time step $k-1$ to the current time step k by employing the system model (1) resulting in the predicted state density

$$f_k^p(\underline{x}_k) := f(\underline{x}_k \mid \tilde{\underline{y}}_{k-1}, \ldots, \tilde{\underline{y}}_1).$$

On the other hand, the filter step incorporates a newly received measurement $\tilde{\underline{y}}_k$ into this propagated state estimate $f_k^p(\underline{x}_k)$ with the aid of the measurement model (2).

In nonlinear Kalman filtering, both state densities are approximated as Gaussian distributions, and the predicted state density is given by

$$f_k^p(\underline{x}_k) \approx \mathcal{N}(\underline{x}_k; \hat{\underline{x}}_k^p, \mathbf{C}_k^p),$$

with predicted state mean

$$\hat{\underline{x}}_k^p = \iint \underline{a}_k(\underline{x}_{k-1}, \underline{w}_k) \cdot f_{k-1}^e(\underline{x}_{k-1}) \cdot f_k^w(\underline{w}_k) d\underline{x}_{k-1} d\underline{w}_k \tag{3}$$

and predicted state covariance matrix

$$\mathbf{C}_k^p = \iint (\underline{a}_k(\underline{x}_{k-1}, \underline{w}_k) - \hat{\underline{x}}_k^p) \cdot (\underline{a}_k(\underline{x}_{k-1}, \underline{w}_k) - \hat{\underline{x}}_k^p)^{\mathrm{T}} \cdot f_{k-1}^e(\underline{x}_{k-1}) \cdot f_k^w(\underline{w}_k) d\underline{x}_{k-1} d\underline{w}_k, \tag{4}$$

[1] The subscript k denotes the discrete time step, matrices are printed bold face, and vectors are underlined.

respectively. Furthermore, the, in general intractable, Bayesian measurement update is also approximated to obtain the posterior Gaussian state density

$$f_k^e(\underline{x}_k) \approx \mathcal{N}(\underline{x}_k; \hat{\underline{x}}_k^e, \mathbf{C}_k^e),$$

with posterior state mean

$$\hat{\underline{x}}_k^e = \hat{\underline{x}}_k^p + \mathbf{C}_k^{x,y} \cdot (\mathbf{C}_k^y)^{-1} \cdot (\tilde{\underline{y}}_k - \hat{\underline{y}}_k) \tag{5}$$

and posterior state covariance matrix

$$\mathbf{C}_k^e = \mathbf{C}_k^p - \mathbf{C}_k^{x,y} \cdot (\mathbf{C}_k^y)^{-1} \cdot (\mathbf{C}_k^{x,y})^{\mathrm{T}}, \tag{6}$$

which are the well-known Kalman Filter formulas [8]. In order to obtain (5) and (6), the measurement mean

$$\hat{\underline{y}}_k = \iint \underline{h}_k(\underline{x}_k, \underline{v}_k) \cdot f_k^p(\underline{x}_k) \cdot f_k^v(\underline{v}_k) d\underline{x}_k d\underline{v}_k, \tag{7}$$

the measurement covariance matrix

$$\mathbf{C}_k^y = \iint (\underline{h}_k(\underline{x}_k, \underline{v}_k) - \hat{\underline{y}}_k) \cdot (\underline{h}_k(\underline{x}_k, \underline{v}_k) - \hat{\underline{y}}_k)^{\mathrm{T}} \cdot f_k^p(\underline{x}_k) \cdot f_k^v(\underline{v}_k) d\underline{x}_k d\underline{v}_k, \tag{8}$$

as well as the cross-covariance matrix of predicted state and measurement

$$\mathbf{C}_k^{x,y} = \iint (\underline{x}_k - \hat{\underline{x}}_k^p) \cdot (\underline{h}_k(\underline{x}_k, \underline{v}_k) - \hat{\underline{y}}_k)^{\mathrm{T}} \cdot f_k^p(\underline{x}_k) \cdot f_k^v(\underline{v}_k) d\underline{x}_k d\underline{v}_k \tag{9}$$

are required.

Unfortunately, computing the above integrals in closed-form is only possible for a small set of system and measurement models, but it yields the best possible Kalman Filter for the given models. In all other cases, numerical integration methods have to be applied. As we aim for an online estimation technique, the employed numerical integration has to possess a *real-time capable computational complexity* and still deliver adequate integration results in order to obtain a good recursive state estimation quality. When looking at the five integrals, it can be seen that the last terms are always a product of two independent Gaussian densities, namely

$$f_{k-1}^e(\underline{x}_{k-1}) \cdot f_k^w(\underline{w}_k) = \mathcal{N}\left(\begin{bmatrix} \underline{x}_{k-1} \\ \underline{w}_k \end{bmatrix}; \begin{bmatrix} \hat{\underline{x}}_{k-1}^e \\ \hat{\underline{w}}_k \end{bmatrix}, \begin{bmatrix} \mathbf{C}_{k-1}^e & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_k^w \end{bmatrix} \right) \tag{10}$$

for the prediction and

$$f_k^p(\underline{x}_k) \cdot f_k^v(\underline{v}_k) = \mathcal{N}\left(\begin{bmatrix} \underline{x}_k \\ \underline{v}_k \end{bmatrix}; \begin{bmatrix} \hat{\underline{x}}_k^p \\ \hat{\underline{y}}_k \end{bmatrix}, \begin{bmatrix} \mathbf{C}_k^p & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_k^v \end{bmatrix} \right) \tag{11}$$

for the measurement update, respectively. By exploiting this fact, an efficient, i.e., fast but still accurate, computation of the integrals is possible. This can be done by replacing the occurring Gaussian distributions (10) and (11) with proper Dirac mixture densities, that is, sample-based density representations, and evaluating the system model (1) and measurement model (2) using these samples. As a result, emphasis is directly put on the important regions of the state space, and the regions covered by only a small portion of the probability mass of the Gaussian densities are neglected. This approach leads to the class of Linear Regression Kalman Filters (LRKFs).

A Dirac mixture approximation of a given probability density function $f_k(\underline{s}_k)$ comprising M_k samples with sample positions $\underline{s}_{k,i}$ and sample weights $\alpha_{k,i}$ is defined as

$$\sum_{i=1}^{M_k} \alpha_{k,i} \cdot \delta(\underline{s}_k - \underline{s}_{k,i}), \tag{12}$$

where $\delta(\cdot)$ denotes the Dirac delta distribution and the sample weights must sum up to one. Such an approximation can be computed in several ways, e.g., by simply using random sampling or deterministic approaches such as done by the UKF.

Now, we assume that an approximation of the Gaussian joint density (10) comprising M_k samples with positions $[\underline{x}_{k-1,i}^{\mathrm{T}}, \underline{w}_{k,i}^{\mathrm{T}}]^{\mathrm{T}}$ and weights $\alpha_{k,i}$ is at hand. By replacing the Gaussian joint density in the integrals (3) and (4) with this Dirac mixture approximation, and using the Dirac sifting property, we obtain an approximation for the predicted state mean

$$\hat{\underline{x}}_k^p \approx \sum_{i=1}^{M_k} \alpha_{k,i} \cdot \underline{a}_k(\underline{x}_{k-1,i}, \underline{w}_{k,i})$$

and the predicted state covariance matrix

$$\mathbf{C}_k^p \approx \sum_{i=1}^{M_k} \alpha_{k,i} \cdot (\underline{a}_k(\underline{x}_{k-1,i}, \underline{w}_{k,i}) - \hat{\underline{x}}_k^p) \cdot (\underline{a}_k(\underline{x}_{k-1,i}, \underline{w}_{k,i}) - \hat{\underline{x}}_k^p)^{\mathrm{T}}.$$

The same procedure is used for computing the integrals required for the measurement update. First, a Dirac mixture approximation of the Gaussian (11) encompassing M_k samples with positions $[\underline{x}_{k,i}^{\mathrm{T}}, \underline{v}_{k,i}^{\mathrm{T}}]^{\mathrm{T}}$ and weights $\alpha_{k,i}$ is computed. Second, by replacing the joint Gaussian with its Dirac mixture approximation in the three integrals (7), (8), and (9), and using once more the Dirac sifting property, we get an approximation for the measurement mean

$$\hat{\underline{y}}_k \approx \sum_{i=1}^{M_k} \alpha_{k,i} \cdot \underline{h}_k(\underline{x}_{k,i}, \underline{v}_{k,i}),$$

the measurement covariance matrix

$$\mathbf{C}_k^y \approx \sum_{i=1}^{M_k} \alpha_{k,i} \cdot (\underline{h}_k(\underline{x}_{k,i}, \underline{v}_{k,i}) - \hat{\underline{y}}_k) \cdot (\underline{h}_k(\underline{x}_{k,i}, \underline{v}_{k,i}) - \hat{\underline{y}}_k)^{\mathrm{T}},$$

and the cross-covariance matrix

$$\mathbf{C}_k^{x,y} \approx \sum_{i=1}^{M_k} \alpha_{k,i} \cdot (\underline{x}_{k,i} - \hat{\underline{x}}_k^p) \cdot (\underline{h}_k(\underline{x}_{k,i}, \underline{v}_{k,i}) - \hat{\underline{y}}_k)^{\mathrm{T}}.$$

It should be noted that the number of samples for the time and the measurement update do not have to be the same. Moreover, both Dirac mixture approximations can be completely different in the way they are obtained, although this is usually not the case.

III. THE SMART SAMPLING KALMAN FILTER WITH SYMMETRIC SAMPLES

In [22], the authors proposed an approach based on the Localized Cumulative Distribution (LCD) to optimally approximate Gaussian distributions with a set of equally weighted samples. This is done by transforming the approximation problem into an optimization problem. Unfortunately, such optimization is very time-consuming, and hence, not suitable for online nonlinear filtering. To enable the LCD approach for online filtering, it is used to optimally sample only a standard normal distribution offline (before filter usage) and transform these samples online (during filter usage) to any required Gaussian with the aid of the Mahalanobis transformation [25]. This is the fundamental basis for the S^2KF [19]. But other nonlinear estimators such as the Progressive Gaussian Filter also make use of this Gaussian sampling technique.

However, the current LCD approach can, and will, arrange the samples in an arbitrary way to optimally approximate a standard normal distribution. More precisely, it does not take the point symmetry of the standard normal distribution explicitly into account so that not all samples have point-symmetric counterparts. Here, we extend the LCD approach to approximate an N-dimensional standard normal distribution with a set of *point-symmetric and equally weighted samples*. Moreover, we improve the numerical stability of the LCD approach when dealing with Gaussian densities to allow approximations of very high dimensions. This new optimal point-symmetric sampling is then used to obtain a symmetric version of the S^2KF.

The use of point-symmetric samples offers several benefits. First, the proposed sampling reflect the point symmetry of the standard normal distribution allowing for more accurate estimation results as will be seen in the evaluation. Second, the used point symmetry makes it possible to capture *all odd moments* of the standard normal distribution exactly (a proof is given in Appendix A). Finally, although not the actual goal of the proposed sampling, the required number of sample positions, i.e., the parameters, that have to be optimized is reduced by half, and hence, speeds up the offline sample computation.

Besides point symmetry, other symmetries such as axial symmetry could also be exploited. However, this would prevent us from using an arbitrary number of samples and would limit the optimizer's control over the sample placement.

In the following, we first define the set of parameters describing point-symmetric Dirac mixtures in Sec. III-A. These parameters have then to be optimized in order to approximate a standard normal distribution in an optimal way. This requires the distance measures between a standard normal distribution and the point-symmetric Dirac mixtures given in Sec. III-B. Subsequently, the gradients of the distance measures are derived in Sec. III-C. Finally, in Sec. III-D, we give a procedure to compute point-symmetric Dirac mixture approximations of standard normal distributions based on the introduced distance measures and their gradients.

A. Point-Symmetric Dirac Mixtures

First, we have to modify the generic Dirac mixture (12) to obtain a point-symmetric one. This is performed by distinguishing between an even and odd number of samples. For the case of $2L$ samples with $L \in \mathbb{N}_+$, that is, the even case, we place the samples point-symmetrically around the state space origin yielding the equally weighted Dirac mixture

$$\frac{1}{2L}\sum_{i=1}^{L}\delta(\underline{s}-\underline{s}_i)+\delta(\underline{s}+\underline{s}_i), \tag{13}$$

with sample positions $\underline{s}_i$ and $-\underline{s}_i$. For $2L+1$ samples, the odd case, we additionally place a sample fixed at the state space origin and obtain the Dirac mixture

$$\frac{1}{2L+1}\left(\delta(\underline{s})+\sum_{i=1}^{L}\delta(\underline{s}-\underline{s}_i)+\delta(\underline{s}+\underline{s}_i)\right). \tag{14}$$

This preserves the desired point symmetry. As the position of the additional sample in the odd case is constant, the set of parameters

$$S := \{\underline{s}_1,\ldots,\underline{s}_L\}$$

is the same for both Dirac mixtures. That is, S describes the entire set of $2L$ or $2L+1$ samples forming a density approximation although S contains only half of the sample positions. Given a set S, the full set of point-symmetric samples is $\{\underline{s}_1,-\underline{s}_1,\ldots,\underline{s}_L,-\underline{s}_L\}$ or $\{\underline{0},\underline{s}_1,-\underline{s}_1,\ldots,\underline{s}_L,-\underline{s}_L\}$, depending on whether the even or the odd case is considered. Moreover, the Dirac mixtures specified in (13) and (14) are always point-symmetric no matter what values S will take.

For example, the UKF sample set comprising $2L$ or $2L+1$ (equally weighted) samples [10] is a special case of these point-symmetric Dirac mixtures. With an even number of samples, it has the parametrization

$$\underline{s}_i = \sqrt{N}\cdot\underline{e}_i \quad \forall i \in \{1,\ldots,N\},$$

where $\underline{e}_i$ denotes the unit vector along the ith dimension. In the odd case, the parametrization is

$$\underline{s}_i = \sqrt{N+0.5}\cdot\underline{e}_i \quad \forall i \in \{1,\ldots,N\},$$

that is, the sample spread is larger due to the additional point mass at the state space origin.

B. Distance Measures

Our goal is to determine the set of parameters S for the above Dirac mixtures so that they approximate a multivariate standard normal distribution in an optimal way. This requires a distance measure between the involved continuous and discrete distributions. As the classical cumulative distribution function is not suitable for the multi-dimensional case [26], we utilize the LCD approach in the same way as the asymmetric S^{2}KF.

DEFINITION III.1 (Localized Cumulative Distribution [19]).

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 $\underline{m} \in \mathbb{R}^N$, $b \in \mathbb{R}_+$, and the symmetric and integrable kernel

$$K(\underline{s}-\underline{m},b) = \exp\left(-\frac{1}{2}\frac{\|\underline{s}-\underline{m}\|_2^2}{b^2}\right).$$

Here, $\underline{m}$ characterizes the location of the kernel and b its size.

The LCD of an N-dimensional standard normal distribution is an integral of a product of two (unnormalized) Gaussians. By using the fact that the product of two Gaussian distributions is also an unnormalized Gaussian and the integral over a probability density equals one, its LCD is obtained by [22]

$$\begin{aligned} F_{\mathcal{N}}(\underline{m},b) &= \int_{\mathbb{R}^N} \mathcal{N}(\underline{s};\underline{0},\mathbf{I}_N) \cdot (2\pi)^{N/2} b^N \mathcal{N}(\underline{s};\underline{m},b^2\mathbf{I}_N) d\underline{s} \\ &= \frac{(2\pi)^{N/2} b^N}{(2\pi)^{N/2}\sqrt{|(1+b^2)\mathbf{I}_N|}} \exp\left(-\frac{1}{2}\frac{\|\underline{m}\|_2^2}{(1+b^2)}\right) \\ &= \left(\frac{b^2}{1+b^2}\right)^{N/2} \exp\left(-\frac{1}{2}\frac{\|\underline{m}\|_2^2}{(1+b^2)}\right), \end{aligned}$$

where $\mathbf{I}_N$ denotes the identity matrix of dimension N. Based on the Dirac sifting property, the LCD of the Dirac mixture comprising an even number of samples is given by

$$\begin{aligned} F_\delta^e(S,\underline{m},b) = \frac{1}{2L}\Bigg(&\sum_{i=1}^{L} \exp\left(-\frac{1}{2}\frac{\|\underline{s}_i-\underline{m}\|_2^2}{b^2}\right) \\ &+ \exp\left(-\frac{1}{2}\frac{\|-\underline{s}_i-\underline{m}\|_2^2}{b^2}\right)\Bigg), \end{aligned}$$

whereas the LCD of the odd Dirac mixture is

$$\begin{aligned} F_\delta^o(S,\underline{m},b) = \frac{1}{2L+1}\Bigg(&\exp\left(-\frac{1}{2}\frac{\|\underline{m}\|_2^2}{b^2}\right) \\ &+ \sum_{i=1}^{L} \exp\left(-\frac{1}{2}\frac{\|\underline{s}_i-\underline{m}\|_2^2}{b^2}\right) \\ &+ \exp\left(-\frac{1}{2}\frac{\|-\underline{s}_i-\underline{m}\|_2^2}{b^2}\right)\Bigg). \end{aligned}$$

To compare the standard normal LCD with a Dirac mixture LCD, we use the modified Cramér-von Mises distance defined as follows.

DEFINITION III.2 (Modified Cramér–von Mises Distance).

The modified Cramér–von Mises (CvM) distance D between two LCDs $F(\underline{m},b)$ and $\tilde{F}(\underline{m},b)$ is given by

$$D(F,\tilde{F}) = \int_0^\infty w(b) \int_{\mathbb{R}^N} (F(\underline{m},b)-\tilde{F}(\underline{m},b))^2 d\underline{m}\, db,$$

with weighting function

$$w(b) = \begin{cases} \pi^{-N/2} b^{1-N}, & b \in (0, b_{\max}] \\ 0, & \text{elsewhere.} \end{cases}$$

The new term $\pi^{-N/2}$ in the weighting function $w(b)$ (in contrast with the definition in [19]) is a consequence of the involved LCDs $F_{\mathcal{N}}$, F_δ^e, and F_δ^o. Without this term, the modified CvM distances between these LCDs would be unbounded for an increasing dimension N, which in turn would make the distances numerically unstable. This improvement now allows the S^{2}KF to compute Dirac mixture approximations for very high state dimensions, e.g., $N > 200$.

Note that the LCD approach is closely related to the concept used in regularized particle filtering [3], [27]. Here, a kernel with a given bandwidth, i.e., kernel size, is convolved with a set of weighted particles, i.e., a Dirac mixture, to improve the sample diversity. Nevertheless, the LCD is also applied to a continuous density to make it comparable with a discrete one. Additionally, we also integrate over all kernel locations $\underline{m}$ and kernel sizes b.

First, we consider the distance between the standard normal distribution and the Dirac mixture comprising an even number of samples, and then extend the results to the odd case. The distance $D(F_{\mathcal{N}}, F_\delta^e)$ can be split into three terms according to

$$D(F_{\mathcal{N}}, F_\delta^e) = D^e(S) = D_1^e - 2D_2^e(S) + D_3^e(S),$$

with the sample-independent part

$$D_1^e = \int_0^{b_{\max}} b \left(\frac{b^2}{1+b^2}\right)^{N/2} db,$$

and the sample-dependent terms

$$D_2^e(S) = \int_0^{b_{\max}} \frac{2b}{2L} \left(\frac{2b^2}{1+2b^2} \right)^{N/2} \cdot \sum_{i=1}^{L} \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{(1+2b^2)} \right) db,$$

and

$$D_3^e(S) = \int_0^{b_{\max}} \frac{2b}{(2L)^2} \sum_{i=1}^{L} \sum_{j=1}^{L} \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b^2} \right) + \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b^2} \right) db.$$

The proof is given in Appendix B. Note that the integration over b is bounded by $b_{\max}$ due to the support of the weighting function $w(b)$. To speed up the distance computation, the following theorem can be applied.

THEOREM III.1. *For a given $b_{\max}$, the following expression for $D_3^e(S)$ can be obtained*

$$\begin{aligned} D_3^e(S) = {} & \frac{2}{(2L)^2} \sum_{i=1}^{L} \sum_{j=1}^{L} \frac{b_{\max}^2}{2} \left(\exp\left(-\frac{1}{2} \frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b_{\max}^2} \right) \right. \\ & \left. + \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b_{\max}^2} \right) \right) \\ & + \frac{1}{8} \left(\|\underline{s}_i - \underline{s}_j\|_2^2 \mathrm{Ei}_0\left(-\frac{1}{2} \frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b_{\max}^2} \right) \right. \\ & \left. + \|\underline{s}_i + \underline{s}_j\|_2^2 \mathrm{Ei}_0\left(-\frac{1}{2} \frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b_{\max}^2} \right) \right), \end{aligned}$$

where $\mathrm{Ei}_0(x)$ *is defined as*

$$\mathrm{Ei}_0(x) := \begin{cases} 0, & \text{if } x = 0 \\ \mathrm{Ei}(x), & \text{elsewhere} \end{cases}$$

and $\mathrm{Ei}(x)$ *denotes the exponential integral*

$$\mathrm{Ei}(x) := \int_{-\infty}^{x} \frac{e^t}{t} dt.$$

PROOF The proof is given in Appendix C.

Now, we consider the case of an odd number of samples. Like in the even case, $D(F_{\mathcal{N}}, F_\delta^o)$ can be split into three terms

$$D(F_{\mathcal{N}}, F_\delta^o) = D^o(S) = D_1^o - 2D_2^o(S) + D_3^o(S).$$

The first part D_1^o is also independent of the samples S and identical to its even counterpart, i.e.,

$$D_1^o = D_1^e.$$

The sample-dependent terms $D_2^o(S)$ and $D_3^o(S)$ can be expressed in terms of the even case plus additional terms due to the fixed sample at the state space origin according to

$$D_2^o(S) = \frac{2L}{2L+1} D_2^e(S) + \int_0^{b_{\max}} \frac{b}{2L+1} \left(\frac{2b^2}{1+2b^2} \right)^{N/2} db$$

and

$$D_3^o(S) = \frac{(2L)^2}{(2L+1)^2} D_3^e(S) + \frac{b_{\max}^2}{2(2L+1)^2} + \int_0^{b_{\max}} \frac{4b}{(2L+1)^2} \sum_{i=1}^{L} \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{2b^2} \right) db.$$

The proof is given in Appendix D. Like for the even case, also the computation of the odd case can be sped up by using the following theorem.

THEOREM III.2. *For a given $b_{\max}$, the following expression for $D_3^o(S)$ can be obtained*

$$\begin{aligned} D_3^o(S) = {} & \frac{(2L)^2}{(2L+1)^2} D_3^e(S) + \frac{b_{\max}^2}{2(2L+1)^2} \\ & + \frac{4}{(2L+1)^2} \sum_{i=1}^{L} \frac{b_{\max}^2}{2} \exp\left(-\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{2b_{\max}^2} \right) \\ & + \frac{1}{8} \|\underline{s}_i\|_2^2 \mathrm{Ei}_0\left(-\frac{1}{2} \frac{\|\underline{s}_i\|_2^2}{2b_{\max}^2} \right), \end{aligned}$$

where $\mathrm{Ei}_0(\cdot)$ *is defined as in Theorem III.1.*

PROOF The proof is given in Appendix E.

The extra terms in $D_2^o(S)$ and $D_3^o(S)$, compared to the even case, reflect the influence of the additional sample, placed at the state space origin, on the distance between the Dirac mixture and the standard normal distribution. The result is that the point mass of the additional sample will cause the other samples to have a slightly larger spread compared to a sample set without the additional sample at the state space origin. Concerning the above mentioned numerical stability, we also give a proof for the boundedness of both distances $D^e(S)$ and $D^o(S)$ in Appendix F. Note also that, like the standard normal distribution, both distance measures $D^e(S)$ and $D^o(S)$ are invariant under rotation/reflection (see Appendix G).

The proposed distance measures can be seen as constrained versions of the asymmetric LCD distance measure. Consequently, a Dirac mixture minimizing the new distance measures can be suboptimal with respect to the asymmetric distance measure. However, the actual goal is to approximate a standard normal distribution as best as possible, not to minimize certain distance measures. More precisely, the term "best" induces the utilized distance measure, and in addition to [22], here "best" also means to preserve the point symmetry of the standard normal distribution. This, in turn, requires new distance measures in the form of the proposed measures $D^e(S)$ and $D^o(S)$.

C. Gradients of the Distance Measures

In order to optimize the parameters S of a given Dirac mixture, we chose to apply a gradient-based iterative optimization procedure. This requires the partial derivatives of the two distance measures $D^e(S)$ and $D^o(S)$ with respect to the set of parameters S. For the even case, the partial derivatives are

$$\frac{\partial D^e(S)}{\partial s_i^{(d)}} = -2\frac{\partial D_2^e(S)}{\partial s_i^{(d)}} + \frac{\partial D_3^e(S)}{\partial s_i^{(d)}} \quad \forall d \in \{1,\ldots,N\},$$

with its two terms

$$\frac{\partial D_2^e(S)}{\partial s_i^{(d)}} = -\frac{s_i^{(d)}}{2L}\int_0^{b_{\max}} \frac{2b}{(1+2b^2)}\left(\frac{2b^2}{1+2b^2}\right)^{N/2} \cdot \exp\left(-\frac{1}{2}\frac{\|\underline{s}_i\|_2^2}{(1+2b^2)}\right) db,$$

and

$$\frac{\partial D_3^e(S)}{\partial s_i^{(d)}} = -\frac{2}{(2L)^2}\int_0^{b_{\max}} \frac{1}{b} \cdot \sum_{j=1}^{L}(s_i^{(d)} - s_j^{(d)})\exp\left(-\frac{1}{2}\frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b^2}\right) + (s_i^{(d)} + s_j^{(d)})\exp\left(-\frac{1}{2}\frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b^2}\right) db. \tag{15}$$

Analogous to $D_3^e(S)$, the following theorem can be used for the computation of its partial derivatives.

THEOREM III.3 *For a given $b_{\max}$, the following expression for $\partial D_3^e(S)/\partial s_i^{(d)}$ can be obtained*

$$\frac{\partial D_3^e(S)}{\partial s_i^{(d)}} = \frac{1}{(2L)^2}\sum_{j=1}^{L}(s_i^{(d)} - s_j^{(d)})\mathrm{Ei}_0\left(-\frac{1}{2}\frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b_{\max}^2}\right) + (s_i^{(d)} + s_j^{(d)})\mathrm{Ei}_0\left(-\frac{1}{2}\frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b_{\max}^2}\right),$$

where $\mathrm{Ei}_0(\cdot)$ *is defined as in Theorem III.1.*

PROOF The proof is given in Appendix H.

As with the distance $D^o(S)$ itself, its partial derivatives

$$\frac{\partial D^o(S)}{\partial s_i^{(d)}} = -2\frac{\partial D_2^o(S)}{\partial s_i^{(d)}} + \frac{\partial D_3^o(S)}{\partial s_i^{(d)}} \quad \forall d \in \{1,\ldots,N\}$$

can be obtained in terms of the even case plus additional terms according to

$$\frac{\partial D_2^o(S)}{\partial s_i^{(d)}} = \frac{2L}{2L+1}\frac{\partial D_2^e(S)}{\partial s_i^{(d)}}$$

and

$$\frac{\partial D_3^o(S)}{\partial s_i^{(d)}} = \frac{(2L)^2}{(2L+1)^2}\frac{\partial D_3^e(S)}{\partial s_i^{(d)}} - \frac{2s_i^{(d)}}{(2L+1)^2}\int_0^{b_{\max}} \frac{1}{b}\exp\left(-\frac{1}{2}\frac{\|\underline{s}_i\|_2^2}{2b^2}\right) db.$$

To ease the computation of the partial derivatives of $D_3^o(S)$, the next theorem can be used.

THEOREM III.4 *For a given $b_{\max}$, the following expression for $\partial D_3^o(S)/\partial s_i^{(d)}$ can be obtained*

$$\frac{\partial D_3^o(S)}{\partial s_i^{(d)}} = \frac{(2L)^2}{(2L+1)^2}\frac{\partial D_3^e(S)}{\partial s_i^{(d)}} + \frac{s_i^{(d)}}{(2L+1)^2}\mathrm{Ei}_0\left(-\frac{1}{2}\frac{\|\underline{s}_i\|_2^2}{2b_{\max}^2}\right),$$

where $\mathrm{Ei}_0(\cdot)$ *is defined as in Theorem III.1.*

PROOF The proof is given in Appendix I.

D. The S²KF with Symmetric Samples

Based on the introduced distance measures $D^e(S)$ and $D^o(S)$, and their partial derivatives, we can compute a Dirac mixture approximation of a standard normal distribution comprising an arbitrary number of optimally placed point-symmetric samples. As this amounts to a simple shape approximation, it does not guarantee an identity sample covariance matrix of the resulting Dirac mixture, a property of key importance in Kalman filtering. Thus, we will additionally constrain the possible resulting Dirac mixtures with this requirement in mind.

For the sample computation, the remaining integrals over b in the distance measures and their gradients are computed with the aid of an adaptive numerical integration scheme, namely the 31-point Gauss-Kronrod quadrature [28]. Furthermore, we utilize the low memory Broyden-Fletcher-Goldfarb-Shanno quasi-Newton optimization (L-BFGS) [29]. The low memory variant is essential here as it avoids the explicit computation and storage of the inverse Hessian matrix of the distance measures. The set of Dirac mixture parameters S encompasses $L \times N$ single parameters to be optimized. Hence, the (inverse) Hessian matrix of $D^e(S)$ or $D^o(S)$ would contain $(L \times N)^2$ entries. When now assuming only a linear increase in the number of samples for an increasing dimension N, that is, $2L = C \cdot N$, with a linear factor C, the size of the Hessian grows with $\mathcal{O}(N^4)$. This problem is illustrated in Fig. 1 for two different linear factors (5 and 10). It can be seen that approximating a 100-dimensional standard normal distribution with a thousand samples would require a Hessian of ≈ 20 gigabytes, and already a Hessian of over 4 gigabytes in case of only 500 samples. Consequently, using the inverse Hessian of $D^e(S)$ or $D^o(S)$ directly in the optimization is intractable.

The computation of the point-symmetric samples works as follows.

1) Choose the desired number of samples M to approximate the N-dimensional standard normal distribution.
2) Depending on the number of samples M, the even distance measure $D^e(S)$ or the odd distance measure $D^o(S)$ is selected.

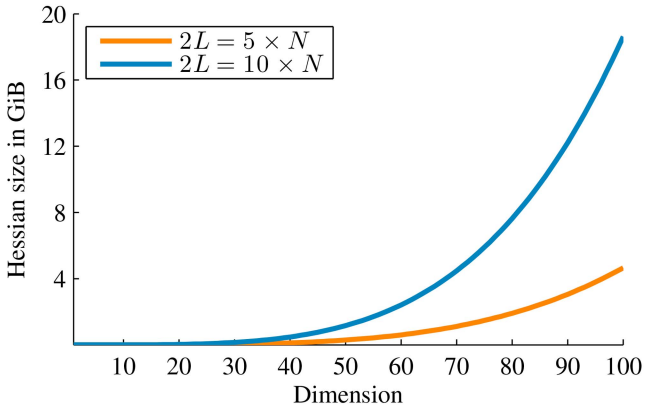


Fig. 1. Size of the Hessian matrix for different dimensions and number of samples.

3) A proper maximum kernel width $b_{\max}$ has to be selected. Generally speaking, the larger the dimension N is the larger $b_{\max}$ has to be in order to consider all sample positions during the optimization, and thus, to get a meaningful approximation. Empirically, we have found that a value of 70 is large enough for up to $N \leq 1000$ dimensions.
4) The initial parameters of the point-symmetric Dirac mixture required by the L-BFGS procedure, i.e., the set S comprising $L = \lfloor M/2 \rfloor$ sample positions, is obtained by drawing L samples randomly from an N-dimensional standard normal distribution.
5) The L-BFGS procedure optimizes the point-symmetric Dirac mixture parameters such that the distance measure is minimized, i.e., it moves the initial L samples (and implicitly their point-symmetric counterparts) in the state space to approximate the standard normal distribution in an optimal way. The point-symmetric Dirac mixture parameters resulting from the L-BFGS procedure are denoted as $\{\underline{z}_i\}_{i=1}^{L}$.
6) These parameters finally have to undergo a transformation so that the resulting Dirac mixture captures the identity covariance matrix of the standard normal distribution as much as possible. The transformation is done by first computing the sample covariance matrix

$$\mathbf{C}^z = \frac{2}{M} \sum_{i=1}^{L} \underline{z}_i \cdot \underline{z}_i^{\mathrm{T}},$$

second computing the Cholesky decomposition

$$\mathbf{C}^z = \mathbf{L}\mathbf{L}^{\mathrm{T}},$$

and third transforming the parameters according to

$$\underline{s}_i = \mathbf{L}^{-1} \cdot \underline{z}_i, \quad \forall i \in \{1, \ldots, L\}.$$

The proof of the transformation is given in Appendix J.
7) The desired set of point-symmetric samples finally approximating the standard normal distribution is either $\{\underline{s}_1, -\underline{s}_1, \ldots, \underline{s}_L, -\underline{s}_L\}$ or $\{\underline{0}, \underline{s}_1, -\underline{s}_1, \ldots, \underline{s}_L, -\underline{s}_L\}$, depending on whether M is even or odd.

Experimentally, we have found that in situations where the covariance matrix was added as an explicit constraint to the optimization procedure, the sample covariance matrix of the resulting Dirac mixture was less accurate compared to the proposed transformation approach. Moreover, the constraint made the optimization procedure much more time-consuming. Consequently, we dropped this approach in favor of the transformation approach.

The results of different LCD-based approximations of a two-dimensional standard normal distribution are depicted in Fig. 2. On the one hand, Figures 2(a) and 2(b) show approximations using the new point-symmetric sampling scheme comprising 12 and 13 samples, respectively. The point-symmetric arrangement around the state space origin can be clearly seen. Note

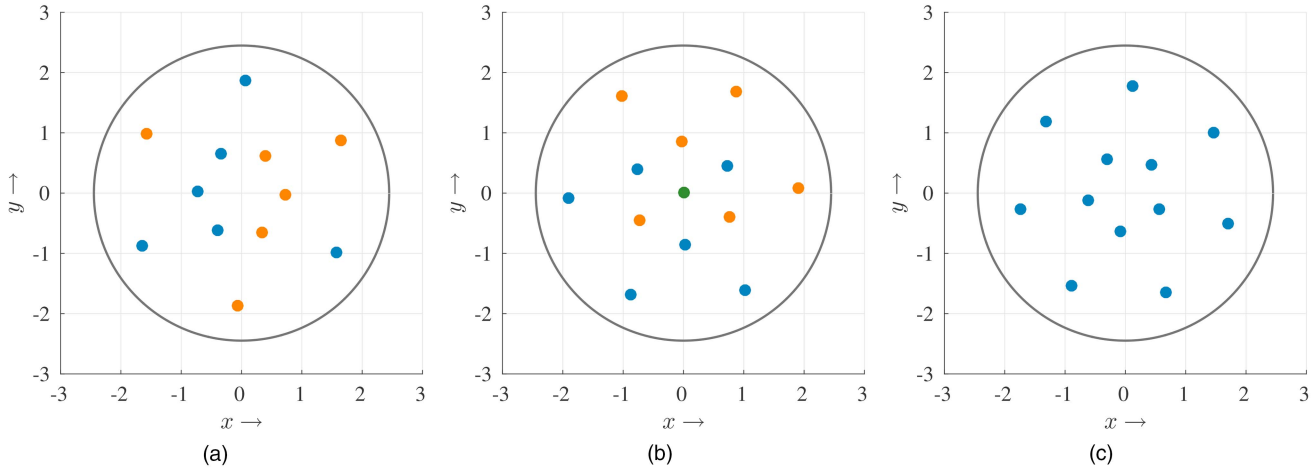


Fig. 2. Different LCD-based approximations of a two-dimensional standard normal distribution with samples $\underline{s}_i$ (blue), point-symmetric counterparts $-\underline{s}_i$ (orange), fixed sample at the state space origin in the odd case (green), and 95% confidence interval of the standard normal distribution (gray). (a) Symmetric approach with 12 samples. (b) Symmetric approach with 13 samples. (c) Asymmetric approach with 12 samples.

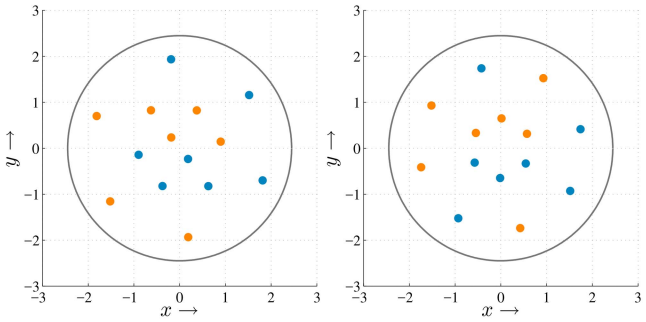


Fig. 3. Two different approximation results with 14 samples of a two-dimensional standard normal distribution.

also the subtle difference in the sample spread of the samples near the state space origin between Fig. 2(a) and Fig. 2(b). This is caused by the additional point mass from the fixed sample at the state space origin. On the other hand, Fig. 2(c) shows an approximation based on the classical asymmetric sampling scheme also comprising 12 samples. Here, the optimization procedure can position all samples individually, and hence, the samples are not necessarily arranged in a point-symmetric way like in the depicted case.

We also have to point out that, due to the random initialization of the L-BFGS procedure and the rotation/reflection invariance of the distance measures, the computed samples are not unique. That is, different approximations with the same number of samples can either differ only in a rotation/reflection or can have a completely different sample placement. For example, Fig. 3 shows two different approximation results for the same number of samples where the difference is not only a rotation or reflection. Concerning the time needed for the computation, on an Intel Core i7-3770 CPU, our implementation in C++ [30] takes about 4 minutes to approximate a 500D standard normal distribution with 10,000 samples and 35 minutes to approximate a 1000D standard normal distribution with 20,000 samples.

Using the above described optimal point-symmetric sampling of a standard normal distribution, we obtain a symmetric version of the S^{2}KF. Furthermore, to avoid a re-computation on every program start, we store any computed Dirac mixture approximation of a standard normal distribution persistent in the file system for later reuse. This mechanism is called the *Sample Cache* and was already used by the asymmetric S^{2}KF.

IV. EVALUATION

In this Section, we want to compare the new point-symmetric sampling scheme of the S^{2}KF with its asymmetric version and other state-of-the-art LRKFs. First, we take a closer look at the approximation of higher-order moments of standard normal distributions. Then, the advantage of using a point-symmetric sampling scheme, and hence, the new version of the S^{2}KF, is discussed by means of a simple symmetric measurement equation. Finally, extended object tracking is performed to compare the recursive state estimation quality of various state-of-the-art LRKFs.

A. Moment Errors of a Standard Normal Distribution

First, we investigate how well the employed sampling schemes of state-of-the-art LRKFs approximate the moments of a standard normal distribution. Thus, we are interested in the expectation values

$$\mathbb{E}[x_1^{n_1} x_2^{n_2} \cdots x_N^{n_N}] = \int_{\mathbb{R}^N} x_1^{n_1} x_2^{n_2} \cdots x_N^{n_N} \mathcal{N}(\underline{x}; \underline{0}, \mathbf{I}_N) d\underline{x},$$

with

$$\sum_{i=1}^{N} n_i = m, \quad 0 \leq n_i \leq m$$

for different dimensions N and moment orders m. This has the advantage of being independent of a concrete system and measurement model. For given N and m, the number of possible combinations $J_{N,m}$ to select the values for n_i is equal to the number of terms in a multinomial sum with N summands raised to the power m, that is,

$$J_{N,m} = \binom{m+N-1}{N-1} = \frac{(m+N-1)!}{(N-1)!m!}.$$

Hence, a moment is characterized by $J_{N,m}$ distinct values.

As all state-of-the-art LRKFs employ a point-symmetric sampling scheme and capture mean and covariance matrix, we focus on higher-order even moments. More precisely, we take a look at the 4th, 6th, and 8th moment, i.e., $m \in \{4,6,8\}$. In many practical applications, 3D and 6D Gaussian distributions are of special interest. For example, the location and orientation in 2D or the position in 3D can be estimated using a three-dimensional system state. When additionally considering velocities in the 2D case or the orientation in the 3D case, a six-dimensional state is required. Thus, we chose to study the approximations of standard normal distributions with these two dimensions, i.e., $N \in \{3,6\}$.

We compare the new point-symmetric S^{2}KF, the UKF with equally weighted samples, the RUKF, the fifth-degree CKF, and the GHKF with two quadrature points. To assess the different LRKF sampling techniques, for each dimension N and moment m we compute a normalized moment error according to

$$\sqrt{\frac{1}{J_{N,m}} \sum_{j=1}^{J_{N,m}} (\mathbb{E}_j^{\text{true}} - \mathbb{E}_j^{\text{LRKF}})^2}, \tag{16}$$

where $\mathbb{E}_j$ denotes one of the $J_{N,m}$ possible combinations for the mth moment, the superscript "true" the true moment value and "LRKF" the LRKF sampling estimate. It is important to note that the moments computed with samples are *not* invariant under rotation. Thus, the same

holds for the normalized moment error (16). To mitigate this, we randomly rotate the sample sets of the UKF, fifth-degree CKF, and GHKF 100 times and build the average moment error. Regarding the S^{2}KF and the RUKF, they do not have unique sample sets, and hence, for both filters 100 sample sets are generated and the average moment errors are computed as well. Moreover, the S^{2}KF and the RUKF are evaluated with different number of samples. Note that we do not compare the asymmetric S^{2}KF here due to its errors in the odd moments. Those errors will likely reduce the errors in even moments. Because of this, a moment-based comparison against the other LRKFs is not meaningful.

The results are depicted in Figures 4 and 5. As the UKF, the fifth-degree CKF, and the GHKF have a fixed number of samples, they are depicted as a bar at their respective employed number of samples. Additionally, for all filters their respective minimum and maximum moment errors are also depicted. The average moment errors of the UKF and the S^{2}KF are nearly identical for the case when both filters use the same number of samples. This is due to the fact that both sample sets are equally weighted and the S^{2}KF places its samples like the UKF (except for the rotation) as this minimize the utilized distance measure. The RUKF, however, scales the utilized UKF sample sets randomly. Consequently, its sample set is not necessarily equally weighted like for the UKF and the S^{2}KF, and hence, their moment errors differ. Considering all average moment errors, the S^{2}KF delivers always smaller errors than the RUKF and the GHKF (for the same number of samples). The sampling of the fifth-degree CKF is the only one that matches the 4th moment exactly. This is based on the fact that the spherical-radial rule of the fifth-degree CKF has a 5th-degree accuracy [13]. Regarding the spread of the moment errors, it can be seen that the S^{2}KF and UKF have nearly the same variability in the errors, and that the maximum moment errors of the GHKF exceed the ones of the S^{2}KF. Furthermore, the S^{2}KF has a much smaller variability in the errors than the RUKF. Most times the errors of the S^{2}KF are as small as the smallest errors of the RUKF or are even smaller, especially for the 6D standard normal distribution.

B. Symmetric Measurement Equations

To illustrate the advantages of using a point-symmetric sampling scheme, we consider the two-dimensional system state

$$\underline{x} = [a, b]^{\mathrm{T}}$$

combined with the scalar and symmetric measurement equation

$$y = h(\underline{x}, v) = \sqrt{a^2 + b^2} + v,$$

where v is zero-mean Gaussian noise with variance $\sigma^2 = 0.01$. Hence, we measure a noisy distance from the system state $\underline{x}$ to the state space origin. Such a symmetric measurement equation arises for example in [31], [32].

We assume that the true system state is

$$\underline{x}_{true} = [1, 2]^{\mathrm{T}},$$

and our goal is to estimate it using a Nonlinear Kalman Filter initialized with mean and covariance matrix

$$\hat{\underline{x}}^p = [0, 0]^{\mathrm{T}}, \quad \mathbf{C}^p = \mathrm{diag}(4, 0.5).$$

The setup is illustrated in Fig. 6. From the the estimator's perspective, the received measurement $\tilde{y}$ could stem from any state located on the gray circle around the prior mean, not only $\underline{x}_{true}$. Hence, a Nonlinear Kalman Filter cannot gain any new information about the hidden system state from the measurement $\tilde{y}$. This situation is reflected in a zero cross-covariance matrix of state and measurement $\mathbf{C}^{x,y}$ in (5) and (6). Consequently, the posterior state estimate (mean and covariance matrix) equals the prior, no matter what prior uncertainty we have.

Now, we try to reproduce this result when using LRKFs. More precisely, we compare the asymmetric S^{2}KF, its new point-symmetric version (both using 11 samples), and the UKF. We perform $R = 100$ Monte Carlo runs. In each run, we reset the initial state estimate, and simulate a noisy measurement $\tilde{y}$ to perform one measurement update. Moreover, both S^{2}KF variants compute a new set of samples approximating a standard normal distribution in every Monte Carlo run. We compute the Root Mean Square Error (RMSE) for the posterior mean

$$\sqrt{\frac{1}{R}\sum_{r=1}^{R} \|\hat{\underline{x}}_r^e - \hat{\underline{x}}^p\|_2^2},$$

where $\hat{\underline{x}}_r^e$ denotes the estimated posterior mean of run r. Additionally, we compute the RMSE of the posterior covariance matrix

$$\sqrt{\frac{1}{R}\sum_{r=1}^{R} \|\mathbf{C}_r^e - \mathbf{C}^p\|^2},$$

where $\mathbf{C}_r^e$ denotes estimated posterior covariance matrix of run r and $\|\cdot\|$ the Frobenius norm.

The results of the evaluation are depicted in Fig. 7. It can be seen that the UKF and the point-symmetric S^{2}KF do not have any errors. This can be explained with their point-symmetric sampling scheme. More precisely, the cross-covariance matrix is computed correctly according to

$$\begin{aligned}\mathbf{C}^{x,y} &= \frac{1}{2L+1}\begin{bmatrix}0\\0\end{bmatrix}\left(\hat{y} - \sqrt{(a_i)^2 + (b_i)^2}\right) \\ &\quad + \frac{1}{2L+1}\sum_{i=1}^{L}\left(\begin{bmatrix}a_i\\b_i\end{bmatrix} - \begin{bmatrix}a_i\\b_i\end{bmatrix}\right)\left(\hat{y} - \sqrt{(a_i)^2 + (b_i)^2}\right) \\ &= \begin{bmatrix}0\\0\end{bmatrix},\end{aligned}$$

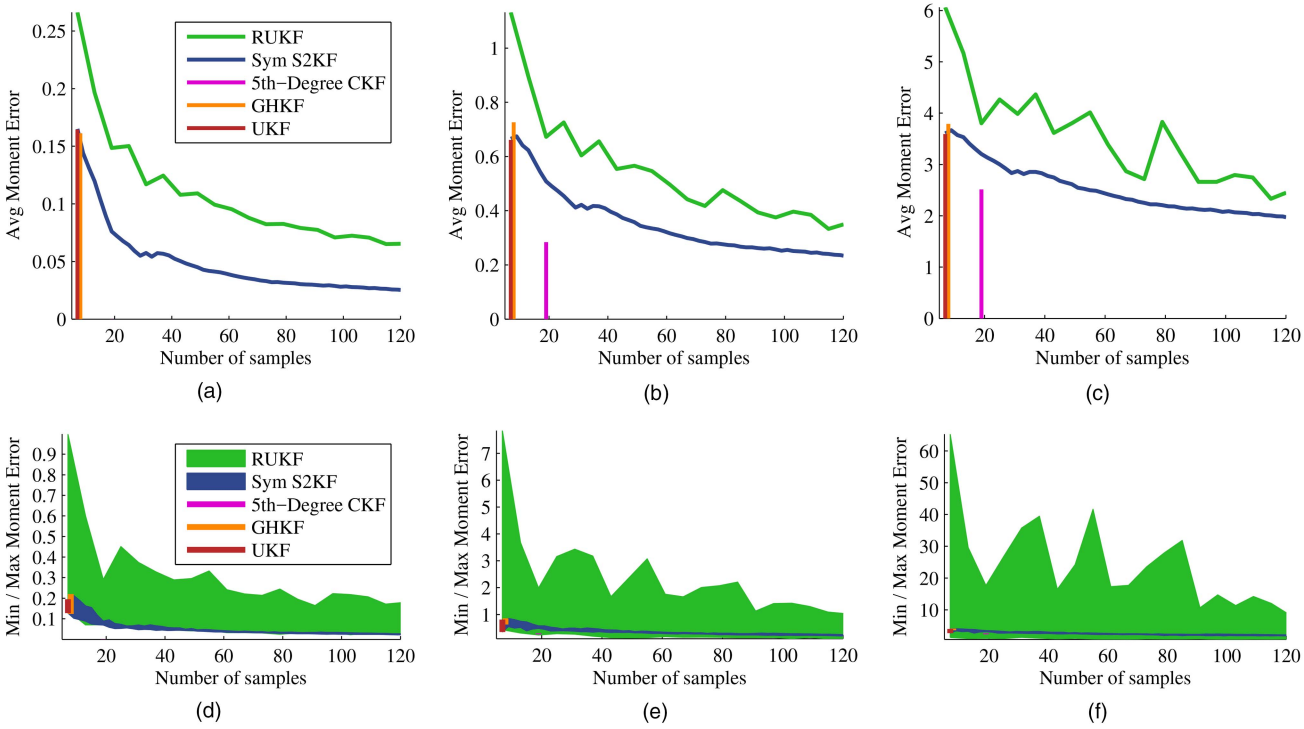


Fig. 4. Moment errors of a 3D standard normal distribution. (a) Average errors 4th moment. (b) Average errors 6th moment. (c) Average errors 8th moment. (d) Min/Max errors 4th moment. (e) Min/Max errors 6th moment. (f) Min / Max errors 8th moment.

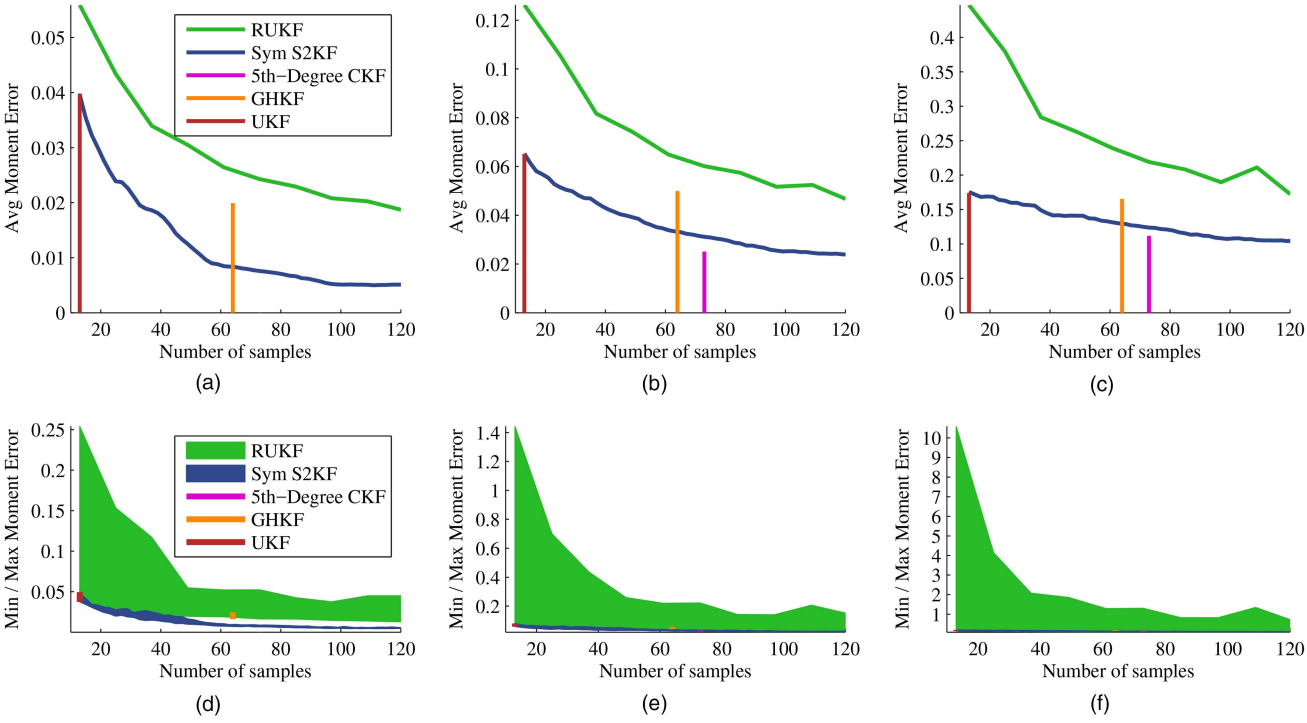


Fig. 5. Moment errors of a 6D standard normal distribution. (a) Average errors 4th moment. (b) Average errors 6th moment. (c) Average errors 8th moment. (d) Min/Max errors 4th moment. (e) Min/Max errors 6th moment. (f) Min/Max errors 8th moment.

where $\hat{y}$ is the measurement mean, $L = 2$ for the UKF, and $L = 5$ for the point-symmetric S^{2}KF. For the asymmetric sampling scheme, however, point-symmetric samples cannot be guaranteed, and hence, the cross-covariance matrix $\mathbf{C}^{x,y}$ do not necessarily evaluates to zero. In such a case, it introduces (theoretically non-existent) correlations between the measurement and the system state. As a consequence, the asymmetric S^{2}KF slightly changes its state estimate mistakenly. Over time, those small errors can accumulate to non-negligible es-

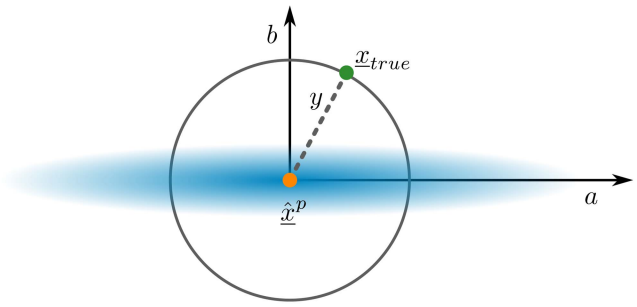


Fig. 6. Symmetric measurement model with prior mean (orange), prior uncertainty (blue), true system state (green).

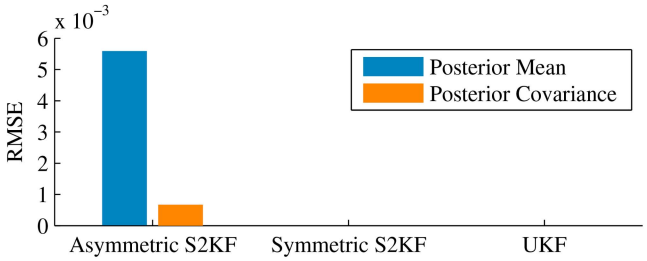


Fig. 7. Estimation errors for symmetric measurement model.

timation errors or even result in filter divergence. The other estimators do not have such a problem due to their point-symmetric sampling. So even such a simple scenario demonstrates the advantages of the new point-symmetric sampling scheme of the S^{2}KF.

C. Extended Object Tracking

Now, we consider estimating the pose and extent of a cylinder in 3D based on a Random Hypersurface Model (RHM) [33], [34]. The system state is composed of position $\underline{c}_k = [c_k^x, c_k^y, c_k^z]^\mathrm{T}$ and velocity $\underline{\nu}_k = [\nu_k^x, \nu_k^y, \nu_k^z]^\mathrm{T}$, rotation angles $\underline{\phi}_k = [\phi_k^x, \phi_k^y]^\mathrm{T}$ and their velocities $\underline{\omega}_k = [\omega_k^x, \omega_k^y]^\mathrm{T}$, as well as the cylinder radius r_k and length l_k according to

$$\underline{x}_k = [\underline{c}_k^\mathrm{T}, \underline{\nu}_k^\mathrm{T}, \underline{\phi}_k^\mathrm{T}, \underline{\omega}_k^\mathrm{T}, r_k, l_k]^\mathrm{T}.$$

The temporal evolution of the cylinder is modeled with a constant velocity model

$$\underline{x}_k = \mathbf{A}\underline{x}_{k-1} + \underline{w},$$

with system matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_2 & \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_2 \end{bmatrix}$$

and zero-mean Gaussian white noise $\underline{w}$ with covariance matrix

$$\mathbf{C}^w = \mathrm{diag}(10^{-6}\mathbf{I}_3, 10^{-4}\mathbf{I}_3, 10^{-10}\mathbf{I}_2, 10^{-5}\mathbf{I}_2, 10^{-4}\mathbf{I}_2).$$

TABLE I
LRKF settings for the measurement update.

LRKF	Number of samples		
Fifth-degree CKF	$2 \cdot 92^2 + 1$	=	16,929
RUKF (with 5 iterations)	$5 \cdot (2 \cdot 92) + 1$	=	921
RUKF (with 20 iterations)	$20 \cdot (2 \cdot 92) + 1$	=	1,841
Asymmetric S^{2}KF	Freely selectable		461
Asymmetric S^{2}KF	Freely selectable		1,841
Symmetric S^{2}KF	Freely selectable		461
Symmetric S^{2}KF	Freely selectable		1,841

This linear model allows to compute the prediction step analytically for all LRKFs.

A measurement is a noisy point

$$\tilde{\underline{y}}_k = [\tilde{y}_k^x, \tilde{y}_k^y, \tilde{y}_k^z]^\mathrm{T}$$

from the cylinder's surface. It is related to the system state by means of the implicit nonlinear measurement equation

$$\underline{0} = \underline{h}(\underline{x}_k, \tilde{\underline{y}}_k, \underline{v}, s) = \begin{bmatrix} (m_k^x)^2 + (m_k^y)^2 - r_k^2 \\ m_k^z - s \cdot l_k \\ (m_k^z - s \cdot l_k)^2 \end{bmatrix}, \qquad (17)$$

where

$$\underline{m}_k = (\mathbf{R}(\phi_k^y) \cdot \mathbf{R}(\phi_k^x))^{-1}(\tilde{\underline{y}}_k - \underline{v} - \underline{c}_k),$$

and zero-mean Gaussian white noise $\underline{v}$ with covariance matrix $\mathbf{C}^v = 0.01 \cdot \mathbf{I}_3$ and multiplicative white noise $s \sim \mathcal{U}(-0.5, 0.5)$.[2] Furthermore, $\mathbf{R}(\cdot)$ denotes a 3D rotation matrix around the respective axis. It is important to note that the measurement equation itself depends on the received measurement $\tilde{\underline{y}}_k$, and the estimator only takes the so-called pseudo measurement $\underline{0}$ as input. The reason for this is that the proposed measurement model tries to minimize the Euclidean distance between the received measurements $\tilde{\underline{y}}_k$ and the cylinder's surface, and thus, generates measurements of value zero in the optimal case. Note also that the quadratic term in the last row of (17) is necessary when dealing with multiplicative noise in combination with Kalman Filters [33], [35].

At each time step, we receive a set of 20 measurements

$$\mathcal{Y}_k = \{\tilde{\underline{y}}_k^{(1)}, \ldots, \tilde{\underline{y}}_k^{(20)}\}.$$

As the order of processing measurements affects the filtered state estimate, we do not process measurements sequentially. More precisely, we process all measurements *at once*, that is, in a single measurement update, by stacking the measurements into a large measurement vector according to

$$\begin{bmatrix} \underline{0} \\ \vdots \\ \underline{0} \end{bmatrix} = \begin{bmatrix} \underline{h}(\underline{x}_k, \tilde{\underline{y}}_k^{(1)}, \underline{v}^{(1)}, s^{(1)}) \\ \vdots \\ \underline{h}(\underline{x}_k, \tilde{\underline{y}}_k^{(20)}, \underline{v}^{(20)}, s^{(20)}) \end{bmatrix}.$$

[2] As LRKFs can only sample Gaussian distributions, the uniform distribution will be approximated as a Gaussian using moment matching.

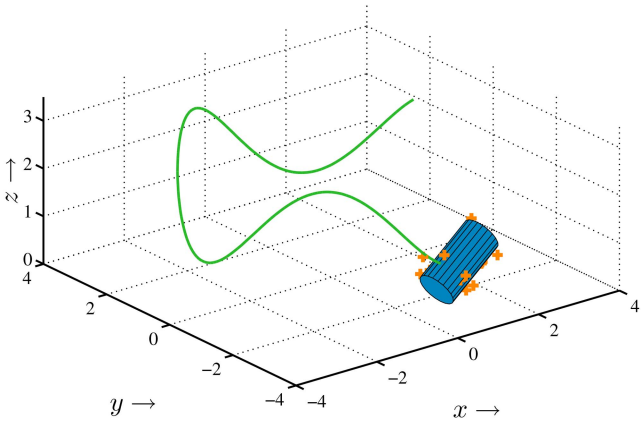


Fig. 8. Cylinder state (blue) after 360 time steps inclusive its trajectory (green line) and 20 noisy measurements (orange crosses).

This, in turn, requires a set of $20 \cdot 4 = 80$ measurement noise variables in total. Together with the twelve-dimensional system state, a LRKF has to sample a 92-dimensional random vector to perform a measurement update. The number of samples used by the investigated LRKFs are summarized in Table I. It should be noted that the GHKF [12] is intractable for the considered scenario as it relies on a Cartesian product and would require at least 2^{92} samples. In addition, the UKF is not able to estimate the cylinder's height as discussed in [20], and hence, it is also not evaluated here.

We simulate a nonlinear trajectory of a cylinder over 500 time steps including rotations in all its three degrees of freedom as depicted in Fig. 8. Additionally, the initial cylinder's length of 1 increases to 1.5 after 200 time steps, and the initial radius of 0.3 increases to 0.4 after further 100 time steps. Finally, at time step 400, the cylinder's length shrinks back to 0.5. We perform 100 Monte Carlo runs. In reach run, we initialize the estimators with

$$\hat{\underline{x}}_0^e = [\hat{\underline{c}}^{\mathrm{T}}, 0, \ldots, 0, 1, 2]^{\mathrm{T}}$$
$$\mathbf{C}_0^e = \mathrm{diag}(\mathbf{C}^c, 10^{-3}\mathbf{I}_3, 10^{-7}\mathbf{I}_4, 10^{-2}\mathbf{I}_2),$$

where $\hat{\underline{c}}$ denotes the mean and $\mathbf{C}^c$ the covariance of the first set of measurements $\mathcal{Y}_0$. For each investigated LRKF, we compute the cylinder position RMSE (Fig. 9(a)), the RMSE of the angle between the true cylinder longitudinal axis and the estimated one (Fig. 9(b)), as well as the cylinder volume RMSE (Fig. 9(c)). Regarding the cylinder position, the RUKF instances were the filters with the largest errors although they used the same or twice the number of samples of the S^{2}KF variants. The asymmetric S^{2}KF was a little bit less accurate than the symmetric S^{2}KF and the fifth-degree CKF. Same results can be observed for the cylinder orientation error. For the cylinder volume error, all estimators had noticeable error peaks at time steps 200, 300, and 400. These can be explained with the abrupt shape changes of the cylinder at the respective time steps. Furthermore, the fifth-degree CKF is not as good as in the other estimation quality criteria, and also the asymmetric S^{2}KF is slightly better than the symmetric S^{2}KF in the beginning.

However, when looking at the runtimes of the respective LRKF measurement updates in Fig. 9(d), the fifth-degree CKF was the slowest filter due to its large amount of samples. The runtimes of the asymmetric and the symmetric S^{2}KF were nearly identical as they used the same number of samples. For the case when the RUKF and the S^{2}KF variants used the same number of samples, the RUKF was slower (11.5 ms compared to 4.5 ms) due to the additional overhead resulting from the creation of several 92-dimensional random orthogonal matrices during each measurement update. All in all, both S^{2}KF variants were the filters yielding the best compromise between runtime performance and estimation accuracy. Moreover, this illustrates the advantage of being able to select the number of samples independently of the state/noise dimensions, in contrast to the fifth-degree CKF.

V. CONCLUSIONS

In this paper, we introduced a new point-symmetric Gaussian sampling scheme for the Smart Sampling Kalman Filter. This reflects the point symmetry of the Gaussian distribution, allows for matching all odd moments of a standard normal distribution exactly, and improves the estimation quality of the S^{2}KF.

After describing the structure of a sample-based Kalman Filter, we extended the general Dirac mixture to a point-symmetric form by distinguishing between an even and an odd number of samples. Then, we adapted the existing LCD distance measure to these new Dirac mixtures and also gave formulas for their respective gradients. These are required by the iterative optimization procedure which optimizes the Dirac mixture parameters to optimally approximate a multi-dimensional standard normal distribution with a set of equally weighted point-symmetric samples. Furthermore, we improved the numerical stability of the optimization, and together with the halved number of Dirac mixture parameters to be optimized, now it is possible to compute optimal approximations of thousand-dimensional standard normal distributions comprising tens of thousands of samples. As the Progressive Gaussian Filter also relies on the S^{2}KF Gaussian sampling technique, it can directly use and benefit from the new point-symmetric sampling scheme.

The evaluations showed that the S^{2}KF can handle symmetric measurement equations now much better when using the new symmetric sampling scheme. It was also shown that the S^{2}KF gave the best compromise between estimation accuracy and filter runtime when dealing with high-dimensional problems such as extended object tracking. Additionally, this illustrated the advantage of the S^{2}KF being able to use an arbitrary number of samples independent of the state/noise dimensions.

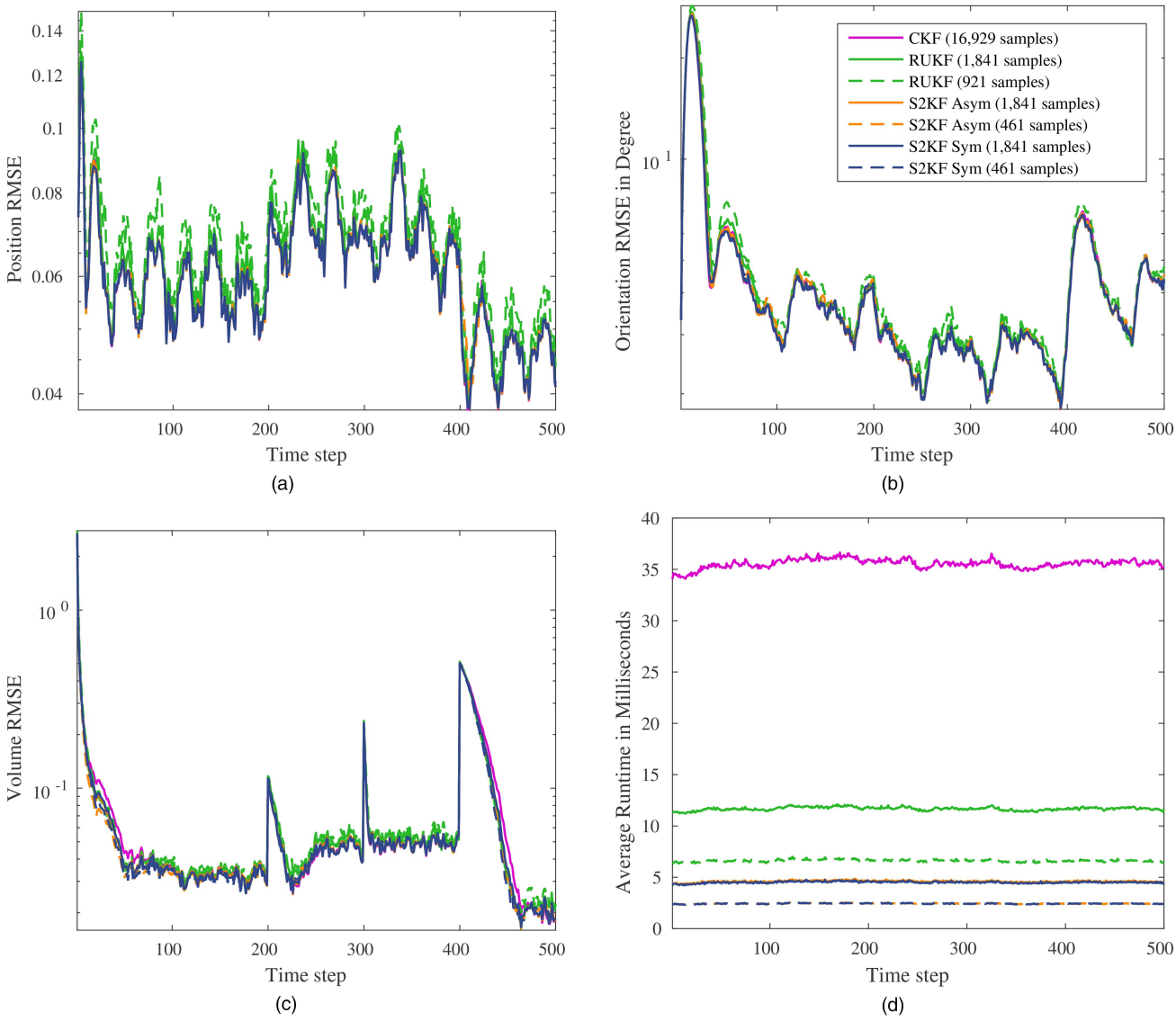


Fig. 9. Cylinder tracking simulation results. (a) Cylinder position error. (b) Cylinder orientation error. (c) Cylinder volume error. (d) Measurement update runtime.

Finally, an open source implementation of the S^{2}KF including both the new point-symmetric Gaussian sampling and the asymmetric Gaussian sampling is available in the Nonlinear Estimation Toolbox [30].

APPENDIX

A. Odd Moments of a Point-Symmetric Dirac Mixture

The odd moments of an arbitrary density function $f(\underline{x})$ with $\underline{x} \in \mathbb{R}^N$ are defined as

$$\mathbb{E}_f\left[\prod_{j=1}^{N} x_j^{n_j}\right] = \int_{\mathbb{R}^N} \prod_{j=1}^{N} x_j^{n_j} \cdot f(\underline{x}) d\underline{x},$$

where

$$\sum_{j=1}^{N} n_j = 2k+1, \quad 0 \leq n_j \leq 2k+1, \; k \in \mathbb{N}.$$

For a standard normal distribution, i.e., $f(\underline{x}) = \mathcal{N}(\underline{x}; \underline{0}, \mathbf{I}_N)$, all odd moments equals zero. Hence, we have to show that this also holds for a point-symmetric Dirac mixture density function comprising $2L$ samples. By replacing the density $f(\underline{x})$ with a point-symmetric Dirac mixture approximation we obtain

$$\begin{aligned}\mathbb{E}_\delta\left[\prod_{j=1}^{N} x_j^{n_j}\right] &= \int_{\mathbb{R}^N} \prod_{j=1}^{N} x_j^{n_j} \frac{1}{2L} \sum_{i=1}^{L} \delta(\underline{x}-\underline{x}_i) + \delta(\underline{x}+\underline{x}_i) d\underline{x} \\ &= \frac{1}{2L} \sum_{i=1}^{L} \left(\prod_{j=1}^{N} x_{i,j}^{n_j} + \prod_{j=1}^{N} (-x_{i,j})^{n_j} \right) \\ &= \frac{1}{2L} \sum_{i=1}^{L} \left(\prod_{j=1}^{N} x_{i,j}^{n_j} - \prod_{j=1}^{N} x_{i,j}^{n_j} \right) = 0.\end{aligned}$$

The same result can be easily obtained for the case of an odd number of samples $2L+1$ where the additional sample is placed at the state space origin.

B. Proof of Distance $D^e(S)$

By using the facts that the distance $D^e(S)$ is composed of sums of products of unormalized Gaussians and their product is also an unnormalized Gaussian as well as the integral over a Gaussian equals always one, the three terms of the distance $D^e(S)$ are obtained according to

$$\begin{aligned}
D_1^e &= \int_0^{b_{\max}} \frac{1}{\pi^{N/2} b^{N-1}} \int_{\mathbb{R}^N} \left(\frac{b^2}{1+b^2}\right)^N \\
&\quad \cdot (2\pi)^N (1+b^2)^N \mathcal{N}(\underline{m}; \underline{0}, (1+b^2)\mathbf{I}_N)^2 d\underline{m}\, db \\
&= \int_0^{b_{\max}} \frac{1}{\pi^{N/2} b^{N-1}} \left(\frac{b^2}{1+b^2}\right)^N \pi^{\frac{N}{2}} (1+b^2)^{N/2} db \\
&= \int_0^{b_{\max}} b \left(\frac{b^2}{1+b^2}\right)^{N/2} db,
\end{aligned}$$

$$\begin{aligned}
D_2^e(S) &= \int_0^{b_{\max}} \frac{1}{\pi^{N/2} b^{N-1}} \int_{\mathbb{R}^N} \left(\frac{b^2}{1+b^2}\right)^{N/2} \\
&\quad \cdot (2\pi)^{N/2} (1+b^2)^{N/2} \\
&\quad \cdot \mathcal{N}(\underline{m}; \underline{0}, (1+b^2)\mathbf{I}_N) \frac{(2\pi)^{N/2} b^N}{2L} \\
&\quad \cdot \sum_{i=1}^{L} \mathcal{N}(\underline{m}; \underline{s}_i, b^2\mathbf{I}_N) + \mathcal{N}(\underline{m}; -\underline{s}_i, b^2\mathbf{I}_N) d\underline{m}\, db \\
&= \int_0^{b_{\max}} \frac{2^N \pi^{N/2} b^{N+1}}{2L} \frac{1}{(2\pi)^{N/2}(1+2b^2)^{N/2}} \\
&\quad \cdot \sum_{i=1}^{L} \exp\left(-\frac{1}{2}\frac{\|\underline{s}_i\|_2^2}{(1+2b^2)}\right) \\
&\quad + \exp\left(-\frac{1}{2}\frac{\|-\underline{s}_i\|_2^2}{(1+2b^2)}\right) db \\
&= \int_0^{b_{\max}} \frac{2b}{2L} \left(\frac{2b^2}{1+2b^2}\right)^{N/2} \\
&\quad \cdot \sum_{i=1}^{L} \exp\left(-\frac{1}{2}\frac{\|\underline{s}_i\|_2^2}{(1+2b^2)}\right) db,
\end{aligned}$$

and

$$\begin{aligned}
D_3^e(S) &= \int_0^{b_{\max}} \frac{1}{\pi^{N/2} b^{N-1}} \int_{\mathbb{R}^N} \left(\frac{(2\pi)^{N/2} b^N}{2L}\right)^2 \\
&\quad \cdot \sum_{i=1}^{L} \mathcal{N}(\underline{m}; \underline{s}_i, b^2\mathbf{I}_N) + \mathcal{N}(\underline{m}; -\underline{s}_i, b^2\mathbf{I}_N) \\
&\quad \cdot \sum_{j=1}^{L} \mathcal{N}(\underline{m}; \underline{s}_j, b^2\mathbf{I}_N) + \mathcal{N}(\underline{m}; -\underline{s}_j, b^2\mathbf{I}_N) d\underline{m}\, db \\
&= \int_0^{b_{\max}} \frac{2^N \pi^{N/2} b^{N+1}}{(2L)^2} \frac{1}{(2\pi)^{N/2}(2b^2)^{N/2}} \sum_{i=1}^{L}\sum_{j=1}^{L} \\
&\quad \exp\left(-\frac{1}{2}\frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b^2}\right) + \exp\left(-\frac{1}{2}\frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b^2}\right) \\
&\quad + \exp\left(-\frac{1}{2}\frac{\|-\underline{s}_i - \underline{s}_j\|_2^2}{2b^2}\right) \\
&\quad + \exp\left(-\frac{1}{2}\frac{\|\underline{s}_j - \underline{s}_i\|_2^2}{2b^2}\right) db \\
&= \int_0^{b_{\max}} \frac{2b}{(2L)^2} \sum_{i=1}^{L}\sum_{j=1}^{L} \exp\left(-\frac{1}{2}\frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b^2}\right) \\
&\quad + \exp\left(-\frac{1}{2}\frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b^2}\right) db.
\end{aligned}$$

C. Proof of Theorem III.1

Like in [22], to compute the term $D_3^e(S)$ we use that for $z > 0$

$$\int_0^{b_{\max}} \frac{2}{b} \exp\left(-\frac{1}{2}\frac{z}{2b^2}\right) db = -\mathrm{Ei}(-\frac{1}{2}\frac{z}{2b_{\max}^2}), \quad (18)$$

where $\mathrm{Ei}(x)$ is the exponential integral defined as

$$\mathrm{Ei}(x) := \int_{-\infty}^{x} \frac{e^t}{t} dt.$$

Moreover, the product rule gives

$$\begin{aligned}
\frac{b_{\max}^2}{2} \exp\left(-\frac{1}{2}\frac{z}{2b_{\max}^2}\right) &= \int_0^{b_{\max}} b \exp\left(-\frac{1}{2}\frac{z}{2b^2}\right) db \\
&\quad + \frac{z}{4} \int_0^{b_{\max}} \frac{1}{b} \exp\left(-\frac{1}{2}\frac{z}{2b^2}\right) db,
\end{aligned}$$

and together with (18) we obtain

$$\begin{aligned}
\int_0^{b_{\max}} b \exp\left(-\frac{1}{2}\frac{z}{2b^2}\right) db &= \frac{b_{\max}^2}{2} \exp\left(-\frac{1}{2}\frac{z}{2b_{\max}^2}\right) \\
&\quad + \frac{z}{8} \mathrm{Ei}(-\frac{1}{2}\frac{z}{2b_{\max}^2}). \quad (19)
\end{aligned}$$

Note that, although $\mathrm{Ei}(x)$ is not defined for $x = 0$, the integral in (19) still converges for $z = 0$ and is equal to $b_{\max}^2/2$. Hence, we introduce the function

$$\mathrm{Ei}_0(x) := \begin{cases} 0, & \text{if } x = 0 \\ \mathrm{Ei}(x), & \text{elsewhere} \end{cases} \quad (20)$$

to also cover the case $z = 0$. By replacing $\mathrm{Ei}(x)$ in (19) with $\mathrm{Ei}_0(x)$, we get the closed-form expression

$$D_3^e(S) = \frac{2}{(2L)^2}\sum_{i=1}^{L}\sum_{j=1}^{L}\frac{b_{\max}^2}{2}\left(\exp\left(-\frac{1}{2}\frac{\|\underline{s}_i-\underline{s}_j\|_2^2}{2b_{\max}^2}\right)\right.$$
$$\left.+\exp\left(-\frac{1}{2}\frac{\|\underline{s}_i+\underline{s}_j\|_2^2}{2b_{\max}^2}\right)\right)$$
$$+\frac{1}{8}\left(\|\underline{s}_i-\underline{s}_j\|_2^2\mathrm{Ei}_0\left(-\frac{1}{2}\frac{\|\underline{s}_i-\underline{s}_j\|_2^2}{2b_{\max}^2}\right)\right.$$
$$\left.+\|\underline{s}_i+\underline{s}_j\|_2^2\mathrm{Ei}_0\left(-\frac{1}{2}\frac{\|\underline{s}_i+\underline{s}_j\|_2^2}{2b_{\max}^2}\right)\right).$$

D. Proof of Distance $D^o(S)$

The distance $D^o(S)$ differs from its even counterpart due to the additional sample placed fixed at the state space origin. This does not effect D_1^o, and hence, it equals D_1^e. The other two terms are sums of their reweighted even counterparts (due to the changed sample weight) and terms comprising also products of unnormalized Gaussians. Hence, they are given as

$$\begin{aligned}
&D_2^o(S)\\
&= \frac{2L}{2L+1}D_2^e(S)+\int_0^{b_{\max}}\frac{1}{\pi^{N/2}b^{N-1}}\int_{\mathbb{R}^N}\left(\frac{b^2}{1+b^2}\right)^{N/2}\\
&\quad\cdot(2\pi)^{N/2}(1+b^2)^{N/2}\mathcal{N}(\underline{m};\underline{0},(1+b^2)\mathbf{I}_N)\\
&\quad\cdot\frac{(2\pi)^{N/2}b^N}{2L+1}\mathcal{N}(\underline{m};\underline{0},b^2\mathbf{I}_N)d\underline{m}\,db\\
&= \frac{2L}{2L+1}D_2^e(S)\\
&\quad+\int_0^{b_{\max}}\frac{2^N\pi^{N/2}b^{N+1}}{2L+1}\frac{1}{(2\pi)^{N/2}(1+2b^2)^{N/2}}db\\
&= \frac{2L}{2L+1}D_2^e(S)+\int_0^{b_{\max}}\frac{b}{2L+1}\left(\frac{2b^2}{1+2b^2}\right)^{N/2}db
\end{aligned}$$

and

$$\begin{aligned}
D_3^o(S) &= \frac{(2L)^2}{(2L+1)^2}D_3^e(S)+\int_0^{b_{\max}}\frac{1}{\pi^{N/2}b^{N-1}}\\
&\quad\cdot\int_{\mathbb{R}^N}\left(\frac{(2\pi)^{N/2}b^N}{2L+1}\right)^2\left(2\cdot\mathcal{N}(\underline{m};\underline{0},b^2\mathbf{I}_N)\right.\\
&\quad\cdot\left(\sum_{i=1}^{L}\mathcal{N}(\underline{m};\underline{s}_i,b^2\mathbf{I}_N)+\mathcal{N}(\underline{m};-\underline{s}_i,b^2\mathbf{I}_N)\right)\\
&\quad\left.+\mathcal{N}(\underline{m};\underline{0},b^2\mathbf{I}_N)^2\right)d\underline{m}\,db\\
&= \frac{(2L)^2}{(2L+1)^2}D_3^e(S)+\int_0^{b_{\max}}\frac{2^N\pi^{N/2}b^{N+1}}{(2L+1)^2}\\
&\quad\cdot\frac{1}{(2\pi)^{N/2}(2b^2)^{N/2}}\left(2\cdot\sum_{i=1}^{L}\exp\left(-\frac{1}{2}\frac{\|\underline{s}_i\|_2^2}{2b^2}\right)\right.\\
&\quad\left.+\exp\left(-\frac{1}{2}\frac{\|-\underline{s}_i\|_2^2}{2b^2}\right)+1\right)db\\
&= \frac{(2L)^2}{(2L+1)^2}D_3^e(S)+\frac{b_{\max}^2}{2(2L+1)^2}\\
&\quad+\int_0^{b_{\max}}\frac{4b}{(2L+1)^2}\sum_{i=1}^{L}\exp\left(-\frac{1}{2}\frac{\|\underline{s}_i\|_2^2}{2b^2}\right)db.
\end{aligned}$$

E. Proof of Theorem III.2

A closed-form expression for $D_3^o(S)$ can directly be obtained by using again (19) and (20) as well as the closed-form expression for $D_3^e(S)$ resulting in

$$\begin{aligned}
D_3^o(S) &= \frac{(2L)^2}{(2L+1)^2}D_3^e(S)+\frac{b_{\max}^2}{2(2L+1)^2}\\
&\quad+\frac{4}{(2L+1)^2}\sum_{i=1}^{L}\frac{b_{\max}^2}{2}\exp\left(-\frac{1}{2}\frac{\|\underline{s}_i\|_2^2}{2b_{\max}^2}\right)\\
&\quad+\frac{1}{8}\|\underline{s}_i\|_2^2\mathrm{Ei}_0\left(-\frac{1}{2}\frac{\|\underline{s}_i\|_2^2}{2b_{\max}^2}\right).
\end{aligned}$$

F. Boundedness of $D^e(S)$ and $D^o(S)$

We show the boundedness of the distances $D^e(S)$ and $D^o(S)$ for an increasing dimension N. For a given $b_{\max}$ it holds

$$\lim_{N\to\infty}D_1^e=\lim_{N\to\infty}\int_0^{b_{\max}}b\underbrace{\left(\frac{b^2}{1+b^2}\right)^{N/2}}_{\to 0\text{ for }N\to\infty}db=0,$$

$$\begin{aligned}
\lim_{N\to\infty}D_2^e(S) &= \lim_{N\to\infty}\int_0^{b_{\max}}\frac{2b}{2L}\left(\frac{2b^2}{1+2b^2}\right)^{N/2}\\
&\quad\cdot\underbrace{\sum_{i=1}^{L}\exp\left(-\frac{1}{2}\frac{\|\underline{s}_i\|_2^2}{(1+2b^2)}\right)}_{\le L}db\\
&\le \lim_{N\to\infty}\int_0^{b_{\max}}b\underbrace{\left(\frac{2b^2}{1+2b^2}\right)^{N/2}}_{\to 0\text{ for }N\to\infty}db=0,
\end{aligned}$$

and

$$\lim_{N\to\infty} D_3^e(S) = \lim_{N\to\infty} \int_0^{b_{\max}} \frac{2b}{(2L)^2} \cdot \sum_{i=1}^{L}\sum_{j=1}^{L} \exp\left(-\frac{1}{2}\frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b^2}\right) \underbrace{+\exp\left(-\frac{1}{2}\frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b^2}\right)}_{\leq 2L^2} db \leq \lim_{N\to\infty} \int_0^{b_{\max}} b\, db = \frac{b_{\max}^2}{2}.$$

Hence, the distance $D^e(S)$ is bounded by $b_{\max}$ according to

$$\lim_{N\to\infty} D^e(S) = \lim_{N\to\infty} D_1^e - 2D_2^e(S) + D_3^e(S) \leq \frac{b_{\max}^2}{2}.$$

In a similar manner, the same result can be obtained for the distance $D^o(S)$.

G. Invariance under Rotation/Reflection

We want to proof that the distance measures $D^e(S)$ and $D^o(S)$ are invariant under rotation/reflection. Let $\mathbf{R} \in \mathbb{R}^{N\times N}$ be an orthogonal matrix and $\underline{a}, \underline{b} \in \mathbb{R}^N$. Then, it holds

$$\|\mathbf{R}\underline{a}\|_2^2 = \|\underline{a}\|_2^2$$
$$\|\mathbf{R}\underline{a} \pm \mathbf{R}\underline{b}\|_2^2 = \|\underline{a} \pm \underline{b}\|_2^2.$$

Hence, given two point-symmetric Dirac mixtures parameterized by the sets

$$A = \{\underline{s}_1, \ldots, \underline{s}_L\}$$

and

$$B = \{\mathbf{R}\underline{s}_1, \ldots, \mathbf{R}\underline{s}_L\},$$

we directly see that $D^e(A) = D^e(B)$ and $D^o(A) = D^o(B)$.

H. Proof of Theorem III.3

With the aid of (18), the terms

$$\int_0^{b_{\max}} \frac{1}{b}(s_i^{(d)} \pm s_j^{(d)}) \exp\left(-\frac{1}{2}\frac{\|\underline{s}_i \pm \underline{s}_j\|_2^2}{2b^2}\right) db \qquad (21)$$

of (15) can be computed according to

$$-\frac{1}{2}(s_i^{(d)} \pm s_j^{(d)}) \mathrm{Ei}\left(-\frac{1}{2}\frac{\|\underline{s}_i \pm \underline{s}_j\|_2^2}{2b_{\max}^2}\right). \qquad (22)$$

For the special case of $\|\underline{s}_i \pm \underline{s}_j\|_2^2 = 0$ also $s_i^{(d)} \pm s_j^{(d)}$ equals zero. Consequently, the integral (21) converges to zero as well. Like in the closed-form expression for $D_3^e(S)$, we can replace $\mathrm{Ei}(x)$ in (22) with (20) to handle such cases and obtain a closed-form expression for $\partial D_3^e(S)/\partial s_i^{(d)}$ according to

$$\frac{\partial D_3^e(S)}{\partial s_i^{(d)}} = \frac{1}{(2L)^2}\sum_{j=1}^{L}(s_i^{(d)} - s_j^{(d)})\mathrm{Ei}_0\left(-\frac{1}{2}\frac{\|\underline{s}_i - \underline{s}_j\|_2^2}{2b_{\max}^2}\right) + (s_i^{(d)} + s_j^{(d)})\mathrm{Ei}_0\left(-\frac{1}{2}\frac{\|\underline{s}_i + \underline{s}_j\|_2^2}{2b_{\max}^2}\right).$$

I. Proof of Theorem III.4

A closed-form expression for $\partial D_3^o(S)/\partial s_i^{(d)}$ can analogously be obtained by exploiting (18) and (22) as well as the closed-form expression for $\partial D_3^e(S)/\partial s_i^{(d)}$ resulting in

$$\frac{\partial D_3^o(S)}{\partial s_i^{(d)}} = \frac{(2L)^2}{(2L+1)^2}\frac{\partial D_3^e(S)}{\partial s_i^{(d)}} + \frac{s_i^{(d)}}{(2L+1)^2}\mathrm{Ei}_0\left(-\frac{1}{2}\frac{\|\underline{s}_i\|_2^2}{2b_{\max}^2}\right).$$

J. Sample Covariance Matrix Correction

Given a point-symmetric Dirac mixture parameterized by $\{\underline{z}_i\}_{i=1}^L$. Our goal is to find a matrix $\mathbf{T}$ to transform these parameters according to

$$\underline{s}_i = \mathbf{T}\cdot\underline{z}_i, \quad \forall i \in \{1,\ldots,L\},$$

such that the sample covariance matrix of the point-symmetric Dirac mixture given by $\{\underline{s}_i\}_{i=1}^L$ equals the identity, i.e.,

$$\mathbf{C}^s = \frac{2}{M}\sum_{i=1}^{L}\underline{s}_i \cdot \underline{s}_i^{\mathrm{T}} = \mathbf{I}_N,$$

where $M = 2L+1$ or $M = 2L$, depending on whether an additional sample is placed at the origin or not. Hence, we set

$$\mathbf{C}^s = \frac{2}{M}\sum_{i=1}^{L}(\mathbf{T}\underline{z}_i)\cdot(\mathbf{T}\underline{z}_i)^{\mathrm{T}} = \mathbf{T}\mathbf{C}^z\mathbf{T}^{\mathrm{T}} \stackrel{!}{=} \mathbf{I}_N.$$

With the matrix decomposition $\mathbf{C}^z = \mathbf{A}\mathbf{A}^{\mathrm{T}}$, we see that $\mathbf{I}_N = (\mathbf{T}\mathbf{A})(\mathbf{T}\mathbf{A})^{\mathrm{T}}$ can be satisfied with $\mathbf{T} = \mathbf{A}^{-1}$. $\mathbf{A}$ can be computed, for example, with the eigendecomposition or Cholesky decomposition of $\mathbf{C}^z$.

REFERENCES

[1] 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.

[2] 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.

[3] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon
Beyond the Kalman Filter: Particle Filters for Tracking Applications.
Artech House Publishers, 2004.

[4] 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.

[5] Jayesh H. Kotecha and Petar M. Djuric
Gaussian Particle Filtering,
IEEE Transactions on Signal Processing, vol. 51, no. 10, pp. 2592–2601, Oct. 2003.

[6] 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.

[7] ———
GPU-Accelerated Progressive Gaussian Filtering with Applications to Extended Object Tracking,
in *Proceedings of the 18th International Conference on Information Fusion (Fusion 2015)*, Washington D. C., USA, Jul. 2015, pp. 1038–1045.

[8] Dan Simon
Optimal State Estimation,
1st ed. Wiley & Sons, 2006.

[9] 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.

[10] Simon J. Julier and Jeffrey K. Uhlmann
Unscented Filtering and Nonlinear Estimation,
in *Proceedings of the IEEE*, vol. 92, Mar. 2004, pp. 401–422.

[11] Ondřej Straka, Jindřich Duník, and Miroslav Šimandl
Unscented Kalman Filter with Advanced Adaptation of Scaling Parameter,
Automatica, vol. 50, no. 10, pp. 2657–2664, Oct. 2014.

[12] 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.

[13] Bin Jia, Ming Xin, and Yang Cheng
High-Degree Cubature Kalman Filter,
Automatica, vol. 49, no. 2, pp. 510–518, Feb. 2013.

[14] Jindřich Duník, Ondřej Straka, and Miroslav Šimandl
The Development of a Randomised Unscented Kalman Filter,
in *Proceedings of the 18th IFAC World Congress*, Milano, Italy, Aug. 2011, pp. 8–13.

[15] Ondřej Straka, Jindřich Duník, Miroslav Šimandl, and Erik Blasch
Randomized Unscented Transform in State Estimation of non-Gaussian Systems: Algorithms and Performance,
in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012, pp. 2004–2011.

[16] Ángel F. García-Fernández, Lennart Svensson, and Mark Morelande
Iterated Statistical Linear Regression for Bayesian Updates,
in *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, Jul. 2014.

[17] Ángel F. García-Fernández, Lennart Svensson, Mark Morelande, and Simo Särkkä
Posterior Linearisation Filter: Principles and Implementation Using Sigma Points,
IEEE Transactions on Signal Processing, vol. 63, no. 20, pp. 5561–5573, Oct. 2015.

[18] Pawe Stano, Zsófia Lendek, Jelmer Braaksma, Robert Babuska, Cees de Keizer, and Arnold J. den Dekker
Parametric Bayesian Filters for Nonlinear Stochastic Dynamical Systems: A Survey,
IEEE Transactions on Cybernetics, vol. 43, no. 6, pp. 1607–1624, Dec. 2013.

[19] Jannik Steinbring and Uwe D. Hanebeck
LRKF Revisited: The Smart Sampling Kalman Filter (S^2KF),
Journal of Advances in Information Fusion, vol. 9, no. 2, pp. 106–123, Dec. 2014.

[20] ———
S^2KF: The Smart Sampling Kalman Filter,
in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013, pp. 2089–2096.

[21] Cihan Ulas and Hakan Temeltas
Planar Features and 6d-SLAM Based on Linear Regression Kalman Filters with N-Dimensional Approximated Gaussians,
in *Proceedings of the 19th IFAC World Congress*, Cape Town, South Africa, Aug. 2014, pp. 10,194–10,199.

[22] 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.

[23] 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.

[24] Igor Gilitschenski, Jannik Steinbring, Uwe D. Hanebeck, and Miroslav Šimandl
Deterministic Dirac Mixture Approximation of Gaussian Mixtures,
in *Proceedings of the 17th International Conference on Information Fusion (Fusion 2014)*, Salamanca, Spain, Jul. 2014.

[25] Wolfgang Härdle and Léopold Simar
Applied Multivariate Statistical Analysis,
2nd ed. Berlin Heidelberg: Springer, 2008.

[26] Uwe D. Hanebeck and Vesa Klumpp
Localized Cumulative Distributions and a Multivariate Generalization of the Cramér-von Mises Distance,
in *Proceedings of the 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008)*, Seoul, Republic of Korea, Aug. 2008, pp. 33–39.

[27] N. Oudjane and C. Musso
Progressive Correction for Regularized Particle Filters,
in *Proceedings of the 3rd International Conference on Information Fusion (Fusion 2000)*, Paris, France, Jul. 2000.

[28] R. Piessens, E. de Doncker-Kapenga, C.W. Überhuber, and D.K. Kahaner
QUADPACK: A Subroutine Package for Automatic Integration,
1st ed., ser. Springer Series in Computational Mathematics. Berlin Heidelberg: Springer, 1983.

[29] Jorge Nocedal and Stephen J. Wright
Numerical Optimization,
2nd ed., ser. Springer Series in Operations Research and Financial Engineering. Springer, 2006.

[30] Jannik Steinbring
"Nonlinear Estimation Toolbox."
[Online]. Available: https://bitbucket.org/nonlinearestimation/toolbox.

[31] Marcus Baum and Uwe D. Hanebeck
The Kernel-SME Filter for Multiple Target Tracking,
in *Proceedings of the 16th International Conference on Information Fusion (Fusion 2013)*, Istanbul, Turkey, Jul. 2013.

[32] Christof Chlebek and Uwe D. Hanebeck
Bayesian Approach to Direct Pole Estimation,
in *Proceedings of the 2014 European Control Conference (ECC 2014)*, Strasbourg, France, Jun. 2014.

[33] Florian Faion, Marcus Baum, and Uwe D. Hanebeck
Tracking 3d Shapes in Noisy Point Clouds with Random Hypersurface Models,
in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Singapore, Jul. 2012.

[34] Marcus Baum and Uwe D. Hanebeck
Extended Object Tracking with Random Hypersurface Models,
IEEE Transactions on Aerospace and Electronic Systems, vol. 50, no. 1, pp. 149–159, Jan. 2014.

[35] 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.

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.

Martin Pander received his Dipl.-Math. techn. in mathematics from the Karlsruhe Institute of Technology (KIT), Germany, in 2015. 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 field of nonlinear state estimation with special interest in progressive methods.

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). From 2005 to 2015, he was 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. He is author and coauthor of more than 380 publications in various high-ranking journals and conferences. 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 is the General Chair of the "19th International Conference on Information Fusion (Fusion 2016)" and the General Chair of the "2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 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).

Sigma-Point Set Rotation for Derivative-Free Filters in Target Tracking Applications

JINDŘICH DUNÍK
ONDŘEJ STRAKA
MIROSLAV ŠIMANDL
ERIK BLASCH

The paper focuses on the state estimation of the nonlinear discrete-time stochastic dynamic systems by the derivative-free filters. In particular the impact of the σ-point set rotation on the performance of the unscented transform and the unscented Kalman filter (UKF) is analysed. It is shown that the σ-point set rotation is an additional user-defined parameter closely tied with the covariance matrix decomposition technique used in σ-point computation that significantly affects the estimation performance. Analysis, algorithms, and recommendations for computations of the optimal σ-point set rotation are provided to determine either the rotation prior to the estimation experiment (off-line) or during the estimation experiment (on-line). Further, two approaches for a reduction of optimisation computational costs are presented. The proposed algorithms, namely the on-line adaptive-sigma-point-set-UKF (AUKF) and off-line trained-sigma-point-set-UKF (TUKF), are illustrated and verified in a numerical study considering two static and two dynamic examples. The TUKF improves the UKF performance, while the computational complexity is preserved. The AUKF further improves the estimate accuracy with increased computational burden.

Manuscript received February 16, 2015; revised December 22, 2015; released for publication April 2, 2016.

Refereeing of this contribution was handled by U. Hanebeck.

Authors' addresses: J. Duník, O. Straka, and M. Šimandl, Department of Cybernetics and European Centre of Excellence NTIS, Faculty of Applied Sciences, University of West Bohemia, Univerzitn´ 8, 306 14 Pilsen, Czech Republic, tel.: +420377632549, fax: +420377632502 (E-mail: {dunikj,straka30,simandl}@kky.zcu.cz), E. Blasch, Air Force Research Lab, Information Directorate, Rome, NY 13441 (E-mail: erik.blasch.1@us.af.mil).

This work was supported by the grant GA15-12068S by the Czech Science Foundation (GACR).

LIST OF ABBREVIATIONS

AUKF	Adaptive-sigma-point-set Unscented Kalman Filter
BRRs	Bayesian Recursive Relations
CKF	Cubature Kalman Filter
DDFs	Divided Difference Filters
KF	Kalman Filter
MC	Monte-Carlo
TUKF	Trained-sigma-point-set Unscented Kalman Filter
PDF	Probability Density Function
RMSE	Root Mean Square Error
STD	Standard Deviation
SVD	Singular Value Decomposition
TE	Taylor series Expansion
UKF	Unscented Kalman Filter
UT	Unscented Transformation

1. INTRODUCTION

State estimation of nonlinear discrete-time stochastic systems plays an important role in many fields such as adaptive and optimal control, fault detection, and signal processing in many applications such as navigation and target tracking.

The state estimation can be solved by various techniques among them the Bayesian and optimisation approaches are widely preferred. The Bayesian approach stems from the solution to the Bayesian recursive relations (BRRs) computing the probability density functions (PDFs) of the state conditioned by the available measurements. The conditional PDF provides a complete description of the immeasurable state, which is valid almost over the whole state space. Therefore, the BRRs-based methods are usually called global. In contrast, the solution to the estimation problem provided by the optimisation approach is in the form of conditional moments of the state, which do not represent a complete description of the estimated state. Therefore, the optimisation-based methods are usually called local[1] as the estimate is valid in a small vicinity of the working point only. The local methods are based on specification and (usually) minimisation of a criterion (most often the mean squared error) under assumption of a certain estimator structure.

The closed-form solution to the state estimation problem is available only for a limited set of the systems. Among these systems, the linear ones are the most important [21], [1]. In other cases [5], [29], [30], if the closed-form solution is not available, approximate methods have to be used. The approximate global methods are based on various approximations to the BRRs solution and are represented for example by the particle filter [5], point-mass method [29], the Gaussian sum method [30], or their combinations [19]. The application

[1] In literature besides the term "local filters," terms "Gaussian filters" or "Kalman filters" can be found as well.

of the global methods is usually limited by their computational complexity especially for high-dimensional systems. Therefore, in most practical applications the approximate local methods are often preferred.

The approximate local filters use the algorithm structure of the Kalman filter[2] (KF) for solving the state estimation problem of nonlinear systems. The local filters can further be divided into two groups; derivative and derivative-free. Derivative filters approximate nonlinear functions in a system description by derivative-based expansions, for example by the Taylor or the Fourier-Hermite series expansions which lead to the extended Kalman filter, the second-order filter [1], or the Fourier-Hermite Kalman filter [27]. Derivative-free local filters are based on differential polynomial interpolations, the unscented transform, or various numerical integration rules. These approximations might be viewed as approximations of the state estimate description by a set of weighted points while the nonlinear functions in the system description remains unaffected. The filters within this group are represented by the divided difference filters (DDFs) based on the Stirling polynomial interpolation [23], the unscented Kalman filter (UKF) based on various versions of the unscented transformation (UT) [16], or the quadrature [13], cubature [2], and stochastic integration [9] based filters utilizing deterministic and stochastic integration rules.[3]

Derivative-free filters (contrary to the derivative ones) evaluate the nonlinear functions in the system description at multiple points (often called σ-points). Placement of the σ-points in the state-space is determined by i) inherent parameters, which are the current mean and associated covariance matrix of the estimate error (i.e., the approximation point) and ii) user-defined parameters affecting the quality of the approximation and subsequently the filter performance. The user-defined parameters might include specification of the σ-point set rotation (also tied to the selection of the covariance matrix decomposition) or the σ-point set scaling (if applicable).

Scaling of the σ-point set by specification of a scaling parameter or parameters has been widely studied in the last decade and recommendations for both fixed and adaptive parameter settings have been proposed. Recommendations for fixed parameters setting stem from a term-by-term comparison of the Taylor series expansion (TE) of the true mean and covariance matrix of a random variable transformed through a nonlinear function with the TE of the point-approximated statistics [23], [17]. In this case, the scaling parameter is a function of the state-space dimension only. On the other hand, an adaptive parameter setting takes an advantage of possibly different parameter values over time instants reflecting the actual working (or approximation) point. Various off-line and on-line adaptive strategies were proposed e.g., in [26], [10], [36], [11], resulting in a non-negligible estimation performance improvement.

Rotation of the σ-point set has been recently directly analysed in [4], [7], [6] and indirectly via the covariance matrix decomposition[4] in [24], [34]. In [7], an adaptive selection of the σ-point set rotation has been studied and illustrated using a simplistic bearings-only tracking example. It was shown that the impact of the σ-point set rotation on the transformation and subsequently on the filter performance is comparable with the σ-point set scaling, which is worth for a deeper analysis.

The goal of the paper is therefore twofold; to summarise the recent results related to the σ-point set rotation in a unified local filter design framework and to thoroughly analyse and explain the impact of the σ-point set rotation on a filter performance. Special emphasis is also placed on a numerical illustration and on reduction of the computational costs of the optimal σ-point set rotation specification.

The rest of the paper is organised as follows. In Section 2 the system description and the introduction to the state estimation with the stress on the UKF are given. Rotation of the σ-point set, its influence, parametrisation and optimisation are discussed in Section 3. Section 4 focuses on two techniques to reduce optimisation costs of σ-point set rotation. In Section 5 a general algorithm for the UKF with rotated σ-point set is summarised. Section 6 compares the proposed algorithms in a numerical study. Conclusions are drawn in Section 7.

2. SYSTEM DESCRIPTION AND STATE ESTIMATION BY UKF

A discrete-time nonlinear stochastic dynamic system is given as

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) + \mathbf{w}_k, \quad k = 0,1,2,\ldots, \tag{1}$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k, \quad k = 0,1,2,\ldots, \tag{2}$$

where the vectors $\mathbf{x}_k \in \mathbb{R}^{n_x}$ and $\mathbf{z}_k \in \mathbb{R}^{n_z}$ represent the immeasurable state of the system and measurement at time instant k, respectively, $\mathbf{f}_k : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ and $\mathbf{h}_k : \mathbb{R}^{n_x} \to \mathbb{R}^{n_z}$ are known vector functions, and $\mathbf{w}_k \in \mathbb{R}^{n_x}$ and $\mathbf{v}_k \in \mathbb{R}^{n_z}$ are the state and measurement white noises. The noises are assumed to be zero-mean with known

[2]The Kalman filter is an optimal (in the mean squared error sense) linear estimator for linear systems.

[3]Although the approximations used in the derivative-free filters originate from quite different basic ideas, the resulting filter algorithms are in many cases identical. As examples, the analysis and analytic comparison of the DDFs and UKF are given in [28], cubature and quadrature filters and UKF in [14], cubature and stochastic integration filters in [9], and the stochastic integration filter and Monte-Carlo filter are in [8].

[4]The particular matrix decompositions lead to the same σ-point sets up to their rotation (or reflection), i.e., all the sets lies on the surface of a hyper-ellipsoid determined by the covariance matrix [34].

covariance matrices $\mathbf{\Sigma}_k^w = \mathrm{cov}[\mathbf{w}_k]$ and $\mathbf{\Sigma}_k^v = \mathrm{cov}[\mathbf{v}_k]$, respectively. The first two moments of the initial state $\mathbf{x}_0$ are assumed to be known as well, i.e., $\mathsf{E}[\mathbf{x}_0] = \bar{\mathbf{x}}_0$, $\mathrm{cov}[\mathbf{x}_0] = \mathbf{P}_0$.

The local state estimation methods provide the point estimate $\hat{\mathbf{x}}_{k|k}$ approximating the conditional mean $\mathsf{E}[\mathbf{x}_k \mid \mathbf{z}^k]$, in which $\mathbf{z}^k = [\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_k]$, and the corresponding covariance matrix of the estimation error $\mathbf{P}_{k|k}^{\mathbf{xx}}$.

The first two moments can be realised as a Gaussian approximation of the conditional PDF, i.e., $p(\mathbf{x}_k \mid \mathbf{z}^k) \approx \mathcal{N}\{\mathbf{x}_k : \hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}^{\mathbf{xx}}\}$ [2], [9], depending on the type of employed approximation.

For calculation of predictive statistics of the state and measurement the UKF utilises the UT.

2.1. Unscented transformation

The UT [16] is a tool for approximate computation of the mean, covariance matrix, and cross-covariance matrix of a transformed random variable

$$\mathbf{y} = \mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x}), \ldots, g_{n_y}(\mathbf{x})]^{\mathrm{T}}, \tag{3}$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$ and $\mathbf{y} \in \mathbb{R}^{n_y}$, under the assumption of known vector function $\mathbf{g}(\cdot)$ and known mean $\hat{\mathbf{x}} = \mathsf{E}[\mathbf{x}]$ and covariance matrix $\mathbf{P}^{\mathbf{xx}} = \mathrm{cov}[\mathbf{x}]$ of $\mathbf{x}$. The UT is based on computation of a symmetric set of deterministic σ-points $\{\mathcal{X}_i\}_{i=0}^{2n_x}$ with appropriate weights $\{\mathcal{W}_i\}_{i=0}^{2n_x}$ according to

$$\mathcal{X}_0 = \hat{\mathbf{x}}, \qquad \mathcal{W}_0 = \frac{\kappa}{n_x + \kappa}, \tag{4}$$

$$\mathcal{X}_j = \hat{\mathbf{x}} + \sqrt{(n_x + \kappa)}\mathbf{S}^{\mathbf{xx}}\mathbf{e}_j, \quad \mathcal{W}_j = \frac{1}{2(n_x + \kappa)}, \tag{5}$$

$$\mathcal{X}_{n_x+j} = \hat{\mathbf{x}} - \sqrt{(n_x + \kappa)}\mathbf{S}^{\mathbf{xx}}\mathbf{e}_j, \quad \mathcal{W}_{n_x+j} = \mathcal{W}_j, \tag{6}$$

where $j = 1, \ldots, n_x$, term $\mathbf{e}_j$ is the jth column of the n_x-dimensional identity matrix $\mathbf{I}_{n_x}$, and $\mathbf{S}^{\mathbf{xx}}$ is a decomposition of the covariance matrix $\mathbf{P}^{\mathbf{xx}}$ such that $\mathbf{S}^{\mathbf{xx}}(\mathbf{S}^{\mathbf{xx}})^{\mathrm{T}} = \mathbf{P}^{\mathbf{xx}}$. The variable κ is a scaling parameter. To get an approximate characteristic of $\mathbf{y}$, each point is transformed through the nonlinear function

$$\mathcal{Y}_i = \mathbf{g}(\mathcal{X}_i), \quad \forall i. \tag{7}$$

The resulting approximate characteristics calculated by the UT are given by

$$\hat{\mathbf{y}}^{\mathrm{UT}} = \sum_{i=0}^{2n_x} \mathcal{W}_i \mathcal{Y}_i, \tag{8}$$

$$\mathbf{P}^{\mathbf{yy},\mathrm{UT}} = \sum_{i=0}^{2n_x} \mathcal{W}_i (\mathcal{Y}_i - \hat{\mathbf{y}}^{\mathrm{UT}})(\cdot)^{\mathrm{T}}, \tag{9}$$

$$\mathbf{P}^{\mathbf{xy},\mathrm{UT}} = \sum_{i=0}^{2n_x} \mathcal{W}_i (\mathcal{X}_i - \hat{\mathbf{x}})(\mathcal{Y}_i - \hat{\mathbf{y}}^{\mathrm{UT}})^{\mathrm{T}}, \tag{10}$$

where the notation $(\mathbf{a})(\cdot)^{\mathrm{T}}$ stands for $(\mathbf{a})(\mathbf{a})^{\mathrm{T}}$. Now, having the UT introduced, the UKF algorithm can be stated.

2.2. Unscented Kalman filter

The UKF algorithm has the following structure [16]:

ALGORITHM 1: *Unscented Kalman Filter*

Step 1: (initialisation) Set the time instant $k = 0$ and define a priori initial condition by the predictive mean $\hat{\mathbf{x}}_{0|-1} = \mathsf{E}[\mathbf{x}_0] = \bar{\mathbf{x}}_0$ and the predictive covariance matrix $\mathbf{P}_{0|-1}^{\mathbf{xx}} = \mathrm{cov}[\mathbf{x}_0] = \mathbf{P}_0$. Set the scaling parameter κ and compute the σ-point weights

$$\mathcal{W}_0 = \frac{\kappa}{n_x + \kappa}, \tag{11}$$

$$\mathcal{W}_i = \frac{1}{2(n_x + \kappa)}, \quad i = 1, \ldots, 2n_x. \tag{12}$$

Step 2: (filtering step) The state predictive estimate is updated with respect to the last measurement $\mathbf{z}_k$ according to

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}), \tag{13}$$

$$\mathbf{P}_{k|k}^{\mathbf{xx}} = \mathbf{P}_{k|k-1}^{\mathbf{xx}} - \mathbf{K}_k \mathbf{P}_{k|k-1}^{\mathbf{zz}} \mathbf{K}_k^{\mathrm{T}}, \tag{14}$$

where $\mathbf{K}_k = \mathbf{P}_{k|k-1}^{\mathbf{xz}}(\mathbf{P}_{k|k-1}^{\mathbf{zz}})^{-1}$ is the filter gain,

$$\hat{\mathbf{z}}_{k|k-1} = \sum_{i=0}^{2n_x} \mathcal{W}_i \mathcal{Z}_{i,k|k-1}, \tag{15}$$

$$\mathbf{P}_{k|k-1}^{\mathbf{zz}} = \sum_{i=0}^{2n_x} \mathcal{W}_i (\mathcal{Z}_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1})(\cdot)^{\mathrm{T}} + \mathbf{\Sigma}_k^v, \tag{16}$$

$$\mathbf{P}_{k|k-1}^{\mathbf{xz}} = \sum_{i=0}^{2n_x} \mathcal{W}_i (\mathcal{X}_{i,k|k-1} - \hat{\mathbf{x}}_{k|k-1})(\mathcal{Z}_{i,k|k-1} - \hat{\mathbf{z}}_{k|k-1})^{\mathrm{T}}, \tag{17}$$

$$\mathcal{Z}_{i,k|k-1} = \mathbf{h}_k(\mathcal{X}_{i,k|k-1}), \tag{18}$$

and the predictive state σ-points are computed according to

$$\mathcal{X}_{0,k|k-1} = \hat{\mathbf{x}}_{k|k-1}, \tag{19}$$

$$\mathcal{X}_{j,k|k-1} = \hat{\mathbf{x}}_{k|k-1} + \sqrt{(n_x + \kappa)}\mathbf{S}_{k|k-1}^{\mathbf{xx}}\mathbf{e}_j, \tag{20}$$

$$\mathcal{X}_{n_x+j,k|k-1} = \hat{\mathbf{x}}_{k|k-1} - \sqrt{(n_x + \kappa)}\mathbf{S}_{k|k-1}^{\mathbf{xx}}\mathbf{e}_j, \tag{21}$$

with $j = 1, \ldots n_x$ and $\mathbf{S}_{k|k-1}^{\mathbf{xx}}$ being a decomposition of $\mathbf{P}_{k|k-1}^{\mathbf{xx}}$.

Step 3: (predictive step) The predictive statistics are given by the relations

$$\hat{\mathbf{x}}_{k+1|k} = \sum_{i=0}^{2n_x} \mathcal{W}_i \mathcal{X}_{i,k+1|k}, \tag{22}$$

$$\mathbf{P}_{k+1|k}^{\mathbf{xx}} = \sum_{i=0}^{2n_x} \mathcal{W}_i (\mathcal{X}_{i,k+1|k} - \hat{\mathbf{x}}_{k+1|k})(\cdot)^{\mathrm{T}} + \mathbf{\Sigma}_k^w, \tag{23}$$

$$\mathcal{X}_{i,k+1|k} = \mathbf{f}_k(\mathcal{X}_{i,k|k}), \tag{24}$$

where the filtering state σ-points are computed according to

$$\mathcal{X}_{0,k|k} = \hat{\mathbf{x}}_{k|k}, \tag{25}$$

$$\mathcal{X}_{j,k|k} = \hat{\mathbf{x}}_{k|k} + \sqrt{(n_x + \kappa)}\mathbf{S}^{\mathbf{xx}}_{k|k}\mathbf{e}_j, \tag{26}$$

$$\mathcal{X}_{n_x+j,k|k} = \hat{\mathbf{x}}_{k|k} - \sqrt{(n_x + \kappa)}\mathbf{S}^{\mathbf{xx}}_{k|k}\mathbf{e}_j, \tag{27}$$

with $j = 1,\dots n_x$ and $\mathbf{S}^{\mathbf{xx}}_{k|k}$ being a decomposition of $\mathbf{P}^{\mathbf{xx}}_{k|k}$.

Let $k = k + 1$. The algorithm continues by **Step 2**.

2.3. σ-point set

The σ-point set of the UT (4)–(6) ((19)–(21) and (25)–(27) in the UKF) is determined by the inherent parameters given by the mean and covariance matrix of $\mathbf{x}$ (i.e., by $\hat{\mathbf{x}}$, $\mathbf{P}^{\mathbf{xx}}$) and by the user-defined parameters. The latter include the *scaling parameter* κ influencing the set scaling (the area of the state-space covered by the points) and the *decomposition technique* which is inevitably tied with the σ-point set rotation [34]. However, independently of the selected user-defined parameter, the weighted σ-point set has at least the same mean and covariance matrix as the original to-be-transformed random variable, i.e.,

$$\hat{\mathbf{x}}^{\mathrm{UT}} = \sum_{i=0}^{2n_x} \mathcal{W}_i\mathcal{X}_i = \hat{\mathbf{x}}, \tag{28}$$

$$\begin{aligned}\mathbf{P}^{\mathbf{xx},\mathrm{UT}} &= \sum_{i=0}^{2n_x} \mathcal{W}_i(\mathcal{X}_i - \hat{\mathbf{x}}^{\mathrm{UT}})(\mathcal{X}_i - \hat{\mathbf{x}}^{\mathrm{UT}})^{\mathrm{T}} \\ &= \mathbf{S}^{\mathbf{xx}}[\mathbf{S}^{\mathbf{xx}}]^{\mathrm{T}} = \mathbf{P}^{\mathbf{xx}}.\end{aligned} \tag{29}$$

This is the ultimate property of any σ-point set.

2.4. Scaling parameter of σ-point set

In literature, three main options for the selection of the scaling parameter can be identified, namely:

- constant parameter (specified prior to the estimation experiment),
- off-line computed time-varying parameter,
- on-line computed time-varying parameter.

The standard setting of the constant scaling parameter follows the recommendation $\kappa = 3 - n_x$ [16], [17], which minimises the error of the TE of the true mean and its UT approximation.[5]

The strategies for time-varying setting of the scaling parameter take into account not only the state dimension but also the actual filter working point (mean and covariance matrix) and the particular nonlinear functions in the state-space model. The strategies for off-line setting can be found e.g., in [26] and for on-line setting in [11], [35], [33]. The on-line strategies compute κ (usually in the filtering step) at every time instant to minimise or maximise a chosen criterion.

2.5. Covariance matrix decomposition

The choice of the covariance matrix decomposition technique was discussed in [34], where it was shown that all decomposition techniques provide the same σ-point set up to the σ-point set rotation or reflection.[6] The consequence is that there are infinitely many decompositions which can be parametrised by a rotation matrix. The analysis also indicated that the impact of the rotation is increasing for covariance matrices with non-negligible off-diagonal elements and with difference in magnitude of the matrix eigenvalues. Illustration of three different σ-point sets based on the Cholesky decomposition, singular value decomposition (SVD), and matrix square root and three different sets obtained by the SVD and consequently rotated by 0, 30, and 60 degrees can be found in Fig. 1.

In [4], [6], and [7] the discussion was further extended by introducing arbitrary rotation and reflection matrices in the σ-point set computation and a significant impact of the rotation was illustrated by a set of numerical examples without a theoretical analysis. As a consequence, the provided recommendations for the σ-point set rotation were rather ad hoc without theoretical justification.

The aim of the following section is to provide a thorough theoretical analysis of the impact of the σ-point rotation on the UT and subsequently on the UKF performance and the justification for on-line and off-line rotation matrix optimisation. The theoretical results are then illustrated by static and dynamic numerical examples.

3. OPTIMISATION OF σ-POINT SET ROTATION IN UT

The σ-points lie on an hyper-ellipsoid with its size determined by the scaling parameter and the semi-axes given by eigenvectors of the covariance matrix $\mathbf{P}^{\mathbf{xx}}$. Specific position of the σ-points on the hyper-ellipsoid are determined by the chosen decomposition. Different decompositions can be obtained by right-multiplying a decomposition by a rotation matrix $\mathbf{C}$. To avoid confusion, in the rest of the paper the SVD will be used to generate the decomposition $\mathbf{S}^{\mathbf{xx}}$ if not stated otherwise. The SVD decomposes the symmetric and positive definite covariance matrix $\mathbf{P}^{\mathbf{xx}}$ as $\mathbf{P}^{\mathbf{xx}} = \mathbf{U}\mathbf{D}\mathbf{U}^{\mathrm{T}}$, where $\mathbf{U}$ is a unitary matrix and $\mathbf{D}$ is a diagonal matrix. Then, the decomposition $\mathbf{S}^{\mathbf{xx}}$ is given by $\mathbf{S}^{\mathbf{xx}} = \mathbf{U}\sqrt{\mathbf{D}}$, where $\sqrt{\mathbf{D}}$ is a diagonal matrix with elements given by square root of elements on the diagonal of $\mathbf{D}$. Note that using the SVD generates the σ-points on the principal axes of a hyper-ellipsoid.

[5] Note that, if $n_x > 3$, the covariance matrices may lose positive semi-definiteness as $\kappa < 0$. Then, it is better to choose $\kappa = 0$ [17], which ensures positive definiteness of covariance matrices. For such a choice, the UKF converts into the cubature Kalman filter (CKF) [2].

[6] Contrary to the rotation, the reflection changes the order of the points besides their rotation. If a symmetric σ-point set is considered, the rotation and reflection are interchangeable.

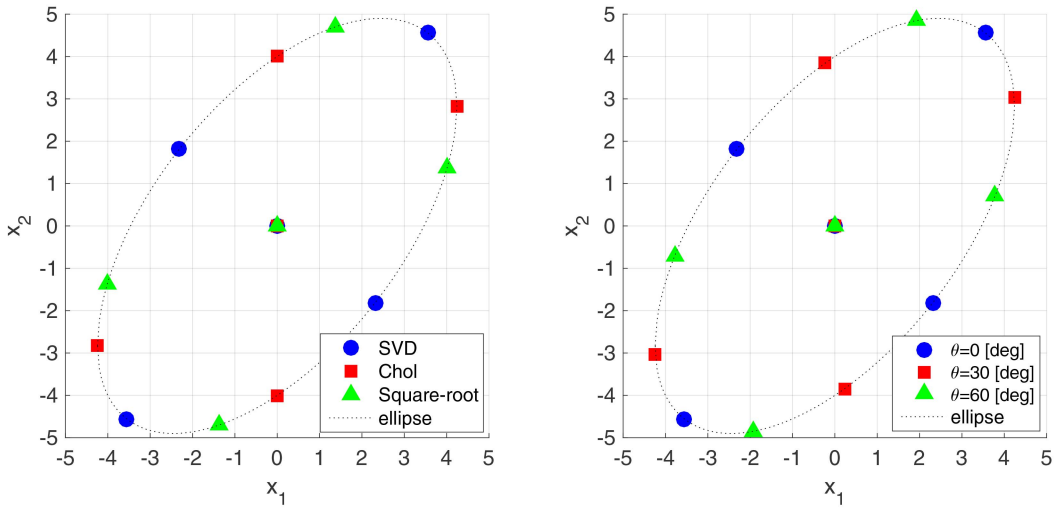


Fig. 1. Illustration of σ-point sets based on different decompositions and with different rotations.

3.1. Rotation of UT σ-point set

The σ-point set is typically computed by (4)–(6). The σ-point set with explicit consideration of the rotation can be computed according to

$$\mathcal{X}_0^r = \hat{\mathbf{x}}, \tag{30}$$

$$\mathcal{X}_j^r = \hat{\mathbf{x}} + \sqrt{(n_x+\kappa)}\mathbf{S}^{\mathbf{xx},r}\mathbf{e}_j, \tag{31}$$

$$\mathcal{X}_{n_x+j}^r = \hat{\mathbf{x}} - \sqrt{(n_x+\kappa)}\mathbf{S}^{\mathbf{xx},r}\mathbf{e}_j, \tag{32}$$

where $j = 1,\dots n_x$ and $\mathbf{S}^{\mathbf{xx},r}$ is a decomposition of $\mathbf{P}^{\mathbf{xx}}$ parametrised by a rotation matrix $\mathbf{C}$

$$\mathbf{S}^{\mathbf{xx},r} = \mathbf{S}^{\mathbf{xx}}\mathbf{C}, \tag{33}$$

where $\mathbf{C}$ is a rotation matrix in n_x dimensional space. Because of the orthogonal property of the rotation matrix, the rotated covariance matrix decomposition $\mathbf{S}^{\mathbf{xx},r}$ also produces the covariance matrix $\mathbf{P}^{\mathbf{xx}}$ as well as the original decomposition $\mathbf{S}^{\mathbf{xx}}$:

$$\mathbf{P}^{\mathbf{xx}} = \mathbf{S}^{\mathbf{xx},r}[\mathbf{S}^{\mathbf{xx},r}]^{\mathrm{T}} = \mathbf{S}^{\mathbf{xx}}\mathbf{C}\mathbf{C}^{\mathrm{T}}[\mathbf{S}^{\mathbf{xx}}]^{\mathrm{T}} = \mathbf{S}^{\mathbf{xx}}\mathbf{I}_{n_x}[\mathbf{S}^{\mathbf{xx}}]^{\mathrm{T}}. \tag{34}$$

As the rotated σ-point set (30)–(32) is still symmetric and the relation (34) is valid, the rotated set preserves the mean and covariance matrix of the original variable $\mathbf{x}$ similarly to the "unrotated" set in (28), (29). The σ-point set rotation, therefore, does not affect the weights $\{\mathcal{W}_i\}_{i=0}^{2n_x}$.

Hence, the rotation matrix $\mathbf{C}$ can be seen as another design parameter of the σ-point set (besides the scaling parameter).

3.2. Effect of σ-points rotation

Let us focus on the UT based approximation of $\hat{\mathbf{y}} = \mathsf{E}[\mathbf{y}]$ by $\hat{\mathbf{y}}^{\mathrm{UT}}$ in (8). The approximation uses the weights $\mathcal{W}_i$ and σ-points $\mathcal{Y}_i(\mathbf{C}) = \mathbf{g}(\mathcal{X}_i^r)$, where the notation emphasises dependence of the σ-points on the rotation matrix $\mathbf{C}$.

The error of approximation denoted as $\tilde{\mathbf{y}}(\mathbf{C})$ is given by

$$\tilde{\mathbf{y}}(\mathbf{C}) = \hat{\mathbf{y}} - \hat{\mathbf{y}}^{\mathrm{UT}}(\mathbf{C}). \tag{35}$$

The optimum rotation will be sought to minimise the weighted squared error of approximation

$$J(\mathbf{C}) = \tilde{\mathbf{y}}(\mathbf{C})^{\mathrm{T}}\mathbf{W}\tilde{\mathbf{y}}(\mathbf{C}), \tag{36}$$

where $\mathbf{W}$ can be any symmetric positive definite matrix weighting individual elements of $\tilde{\mathbf{y}}(\mathbf{C})$. The weight matrix $\mathbf{W}$ can for example be chosen as $\mathbf{W} = (\mathbf{P}^{\mathbf{yy}})^{-1}$, in which case the criterion puts stress on the elements of $\tilde{\mathbf{y}}(\mathbf{C})$ corresponding to small diagonal values of $\mathbf{P}^{\mathbf{yy}}$.

The true mean $\hat{\mathbf{y}}$ can be expressed using the TE of $\mathbf{g}$ around $\hat{\mathbf{x}}$ as

$$\begin{aligned}\hat{\mathbf{y}} = \mathsf{E}[\mathbf{y}] = \mathsf{E}[\mathbf{g}(\mathbf{x})] &= \mathsf{E}[\sum_{i=0}^{\infty}\frac{1}{i!}\mathbf{g}^{(i)}(\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^{\otimes i}] \\ &= \sum_{i=0}^{\infty}\frac{1}{i!}\mathbf{g}^{(i)}(\hat{\mathbf{x}})\mathbf{M}^{\mathbf{x}(i)},\end{aligned} \tag{37}$$

where $\otimes$ is the Kronecker product and the jth row of $\mathbf{g}^{(i)}(\hat{\mathbf{x}}) \in \mathbb{R}^{n_y\times(n_x)^i}$ is given by

$$\mathbf{g}_j^{(i)}(\hat{\mathbf{x}}) = \sum_{l_1,\dots l_i}^{n_x,\dots,n_x}\frac{\partial^i g_j(\mathbf{x})}{\partial\mathbf{x}_{l_1}\cdots\partial\mathbf{x}_{l_i}}\bigg|_{\mathbf{x}=\hat{\mathbf{x}}}(\mathbf{e}_{l_1}\otimes\mathbf{e}_{l_2}\otimes\cdots\otimes\mathbf{e}_{l_i}) \tag{38}$$

with $\mathbf{x}_{l_i}$ being the l_ith element of $\mathbf{x}$. Further, $(\mathbf{x}-\hat{\mathbf{x}})^{\otimes i} = (\mathbf{x}-\hat{\mathbf{x}})\otimes(\mathbf{x}-\hat{\mathbf{x}})^{\otimes i-1}$, $(\mathbf{x}-\hat{\mathbf{x}})^{\otimes 0} = \mathbf{1}_{n_x\times 1}$, $\mathbf{M}^{\mathbf{x}(i)} = \mathsf{E}[(\mathbf{x}-\hat{\mathbf{x}})^{\otimes i}]$, and $\mathbf{M}^{\mathbf{x}(i)} \in \mathbb{R}^{(n_x)^i\times 1}$ is the ith central moments of $\mathbf{x}$ stacked column-wise.

Similarly, by expansion of each transformed σ-point $\mathcal{Y}_i(\mathbf{C}) = \mathbf{g}(\mathcal{X}_i^r)$ around the mean $\hat{\mathbf{x}}$, the mean $\hat{\mathbf{y}}^{\mathrm{UT}}(\mathbf{C})$ can

be written as

$$\hat{\mathbf{y}}^{\mathrm{UT}}(\mathbf{C}) = \sum_{j=0}^{2n_x} \mathcal{W}_j \sum_{i=0}^{\infty} \frac{1}{i!} \mathbf{g}^{(i)}(\hat{\mathbf{x}})(\mathcal{X}_j^r - \hat{\mathbf{x}})^{\otimes i}$$
$$= \sum_{i=0}^{\infty} \frac{1}{i!} \mathbf{g}^{(i)}(\hat{\mathbf{x}}) \sum_{j=0}^{2n_x} \mathcal{W}_j (\mathcal{X}_j^r - \hat{\mathbf{x}})^{\otimes i}$$
$$= \sum_{i=0}^{\infty} \frac{1}{i!} \mathbf{g}^{(i)}(\hat{\mathbf{x}}) \mathbf{M}^{\mathbf{x}(i),\mathrm{UT}}(\mathbf{C}), \tag{39}$$

where $\mathbf{M}^{\mathbf{x}(i),\mathrm{UT}}(\mathbf{C})$ is an UT approximation of $\mathbf{M}^{\mathbf{x}(i)}$. Note that

$$\mathbf{M}^{\mathbf{x}(1)} = \mathbf{M}^{\mathbf{x}(1),\mathrm{UT}}(\mathbf{C}) = \mathbf{0}_{n_x \times 1}, \tag{40}$$

$$\mathbf{M}^{\mathbf{x}(2)} = \mathbf{M}^{\mathbf{x}(2),\mathrm{UT}}(\mathbf{C}) = \mathrm{vec}(\mathbf{P}^{\mathbf{xx}}), \tag{41}$$

where $\mathrm{vec}(\mathbf{A})$ is a column vector obtained by stacking the columns of $\mathbf{A}$. Hence, the first two moments are independent of the σ-point rotation $\mathbf{C}$.

Assuming $\mathbf{x}$ being Gaussian and the σ-point set being symmetric, the odd terms are zero $\mathbf{M}^{\mathbf{x}(2i+1)} = \mathbf{0}$, $\mathbf{M}^{\mathbf{x}(2i+1),\mathrm{UT}} = \mathbf{0}$, $\forall i \in \mathbb{Z}^*$, and considering (40) and (41), the error of approximation is equal to

$$\tilde{\mathbf{y}}(\mathbf{C}) = \sum_{i=2}^{\infty} \frac{1}{(2i)!} \mathbf{g}^{(2i)}(\hat{\mathbf{x}})(\mathbf{M}^{\mathbf{x}(2i)} - \mathbf{M}^{\mathbf{x}(2i),\mathrm{UT}}(\mathbf{C})). \tag{42}$$

This means that the first non-zero term of the error $\tilde{\mathbf{y}}(\mathbf{C})$ expansion (42) is the fourth one, i.e., $\frac{1}{4!}\mathbf{g}^{(4)}(\hat{\mathbf{x}})(\mathbf{M}^{\mathbf{x}(4)} - \mathbf{M}^{\mathbf{x}(4),\mathrm{UT}}(\mathbf{C}))$. For a scalar function $\mathbf{g}$ the fourth term may be zeroed by a choice of the scaling parameter κ [32] and the first non-zero term is the sixth term. Nevertheless, for general vector functions $\mathbf{g}$ the fourth term is non-zero and depends on the rotation matrix $\mathbf{C}$. This implies that the value of the criterion (36) also depends on the rotation matrix $\mathbf{C}$, of which appropriate choice may lead to a minimum value of the criterion and consequently to the best $\hat{\mathbf{y}}^{\mathrm{UT}}$.

3.3. Selection of the optimum rotation

A direct utilisation of the TE of the mean error (42) for the rotation matrix $\mathbf{C}$ optimisation purposes is impossible due to the infinite number of terms. Assuming the magnitude of the ith term in (42) is decreasing with $i \to \infty$, the criterion (36) can be replaced by

$$J^4(\mathbf{C}) = [\tfrac{1}{4!}\mathbf{g}^{(4)}(\hat{\mathbf{x}})\tilde{\mathbf{M}}^{\mathbf{x}(4)}(\mathbf{C})]^{\mathrm{T}}\mathbf{W}[\tfrac{1}{4!}\mathbf{g}^{(4)}(\hat{\mathbf{x}})\tilde{\mathbf{M}}^{\mathbf{x}(4)}(\mathbf{C})]. \tag{43}$$

where $\tilde{\mathbf{M}}^{\mathbf{x}(4)}(\mathbf{C}) = \mathbf{M}^{\mathbf{x}(4)} - \mathbf{M}^{\mathbf{x}(4),\mathrm{UT}}(\mathbf{C})$. Hence, $J^4(\mathbf{C})$ in (43) contains the first non-zero term of (42).

The criterion $J^4(\mathbf{C})$ can be used in the UT for finding an optimal rotation matrix $\mathbf{C}$. However, its utilisation is not very practical as the criterion requires calculation of fourth derivatives of $\mathbf{g}$, which might be limiting for the UT and subsequently for the UKF designed to be derivative-free.

An alternative of (36) more suitable for optimisation of the rotation matrix $\mathbf{C}$ to achieve the smallest error is based on a replacement of the mean $\hat{\mathbf{y}}$ by a sample $\mathbf{y}_s$ of the variable $\mathbf{y}$. This leads to the criterion

$$J^{\mathrm{S}}(\mathbf{C}) = [\mathbf{y}_s - \hat{\mathbf{y}}^{\mathrm{UT}}(\mathbf{C})]^{\mathrm{T}}\mathbf{W}[\mathbf{y}_s - \hat{\mathbf{y}}^{\mathrm{UT}}(\mathbf{C})], \tag{44}$$

which is suitable for an application in the UKF.

Note that the criterion (36) and its consequent simplifications (43) and (44) aim to minimise the approximation error of the mean $\hat{\mathbf{y}}^{\mathrm{UT}}$. Other criteria also can be proposed for minimising approximation error of other quantities, such as the covariance matrix $\mathbf{P}^{\mathbf{yy},\mathrm{UT}}$. In that case a modification analogous to (43) can be proposed while no modification analogous to (44) exists as there is no sample of $\mathbf{P}^{\mathbf{yy}}$ available.

3.4. Choice of the weight matrix

It has been suggested above to choose the weight matrix $\mathbf{W}$ as the inverse of the covariance matrix $\mathbf{P}^{\mathbf{yy}}$ to properly weight individual elements of the mean error. This choice is however usually impossible as the covariance matrix $\mathbf{P}^{\mathbf{yy}}$ is usually unknown and is approximated by the UT. Its approximation also depends on the choice of the rotation matrix $\mathbf{C}$. So it may be replaced by its estimate $\mathbf{P}^{\mathbf{yy},\mathrm{UT}}(\mathbf{C})$, in which case the criterion would be

$$J^{\mathrm{S}}(\mathbf{C}) = [\mathbf{y}_s - \hat{\mathbf{y}}^{\mathrm{UT}}(\mathbf{C})]^{\mathrm{T}}(\mathbf{P}^{\mathbf{yy},\mathrm{UT}}(\mathbf{C}))^{-1}[\mathbf{y}_s - \hat{\mathbf{y}}^{\mathrm{UT}}(\mathbf{C})]. \tag{45}$$

Hence, by minimising (45), the difference $\mathbf{y}_s - \hat{\mathbf{y}}^{\mathrm{UT}}(\mathbf{C})$ and $(\mathbf{P}^{\mathbf{yy},\mathrm{UT}}(\mathbf{C}))^{-1}$ are minimised and thus $\mathbf{P}^{\mathbf{yy},\mathrm{UT}}(\mathbf{C})$ is maximised.

If $\mathbf{W}$ in (44) is selected as $\mathbf{W} = (\mathbf{P}^{\mathbf{yy},\mathrm{UT}}(\mathbf{C}))^{-1}$, then $J^{\mathrm{S}}(\mathbf{C})$ is χ^2 distributed random variable with n_y degrees of freedom with the mean $\mathsf{E}[J^{\mathrm{S}}(\mathbf{C})] = n_y$. Also note that $\mathsf{E}[\mathbf{y}_s - \hat{\mathbf{y}}^{\mathrm{UT}}(\mathbf{C})] = \hat{\mathbf{y}} - \hat{\mathbf{y}}^{\mathrm{UT}}(\mathbf{C})$. Consequently, the criterion can be formulated as

$$J^{\mathrm{MS}}(\mathbf{C}) = |[\mathbf{y}_s - \hat{\mathbf{y}}^{\mathrm{UT}}(\mathbf{C})]^{\mathrm{T}}(\mathbf{P}^{\mathbf{yy},\mathrm{UT}}(\mathbf{C}))^{-1} \times [\mathbf{y}_s - \hat{\mathbf{y}}^{\mathrm{UT}}(\mathbf{C})] - n_y|. \tag{46}$$

Minimising (46) results in $J^{\mathrm{S}}(\mathbf{C})$ (45), computed on the basis of predicted measurement statistics, which is closest to its mean value, i.e., to n_y.

3.5. Rotation parametrisation

Minimisation of the criterion (44) w.r.t. the rotation matrix $\mathbf{C}$ directly is quite difficult and should be avoided. The reason is twofold; the rotation matrix has to be an orthogonal matrix, thus it is difficult to perform minimisation while the structure is respected, and n_x^2 elements need to be found. Instead the rotation matrix could be parametrised by a set of

$$n_\theta = n_x(n_x - 1)/2 \tag{47}$$

rotation angles $\theta_1, \ldots, \theta_{n_\theta}$. Therefore, if $J(\mathbf{C})$ is minimised with respect to the rotation matrix parametrised by the rotation angles, then n_θ is the dimension of the optimisation space. Additionally, the rotation angles correspond to a minimal representation of a general rotation in $\mathbb{R}^{n_x}$.

As an example, parametrisation of a rotation matrix by six consecutive rotations $\theta_1, \theta_2, \ldots, \theta_6$ in four dimensional space is given by [22], [12]

$$\begin{aligned}\mathbf{C} = &\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos(\theta_6) & -\sin(\theta_6) \\ 0 & 0 & \sin(\theta_6) & \cos(\theta_6) \end{bmatrix} \\ &\times \begin{bmatrix} \cos(\theta_5) & 0 & 0 & -\sin(\theta_5) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \sin(\theta_5) & 0 & 0 & \cos(\theta_5) \end{bmatrix} \\ &\times \begin{bmatrix} \cos(\theta_4) & 0 & -\sin(\theta_4) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta_4) & 0 & \cos(\theta_4) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_3) & 0 & -\sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & \sin(\theta_3) & 0 & \cos(\theta_3) \end{bmatrix} \\ &\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_2) & -\sin(\theta_2) & 0 \\ 0 & \sin(\theta_2) & \cos(\theta_2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &\times \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{48}$$

where the last right-hand side matrix rotates a vector through an angle θ_1 in the $x-y$ plane (about the axis perpendicular to the plane) with $\mathbf{x} = [x, y, u, z]^{\mathrm{T}} \in \mathbb{R}^4$.

4. REDUCTION OF OPTIMISATION COSTS

Due to quadratic dependence of the number of rotation angles n_θ (47) on the state dimension it is convenient to pay attention to a possible costs reduction. In this section two approaches to optimisation costs reduction will be described. The reduction is achieved by decreasing the dimension n_θ of the optimisation space.

The first approach is applicable only for a class of functions $\mathbf{g}$ and offers a costs reduction that does not affect the smallest attainable approximation error of $\hat{\mathbf{y}}^{\mathrm{UT}}(\mathbf{C})$ and $\mathbf{P}^{\mathbf{yy},\mathrm{UT}}(\mathbf{C})$. Hence, the approach will be called lossless.

The second approach is applicable for general functions $\mathbf{g}$ and suggests optimising over the angles with a significant effect only. Application of this approach leads a suboptimal solution, i.e., to the approximation error probably higher than if the optimisation over the full optimisation space is done. Hence, the approach will be called lossy.

4.1. Lossless optimisation space dimension decreasing

If the variable $\mathbf{y}$ is a function of only some elements of $\mathbf{x}$, then the optimisation over some rotation angles does not affect the mean approximation $\hat{\mathbf{y}}^{\mathrm{UT}}(\mathbf{C})$ and the covariance matrix approximation $\mathbf{P}^{\mathbf{yy},\mathrm{UT}}(\mathbf{C})$. Hence, these rotations are useless in the optimisation and the dimension of the optimisation space can be decreased. This is for example the case of tracking applications where only some elements (usually positional) of the state $\mathbf{x}_k$ are directly observed by the measurement $\mathbf{z}_k$, while other elements (velocity, acceleration) are not.

Suppose, the variable $\mathbf{x}$ can be split into two parts, $\mathbf{x}_a \in \mathbb{R}^{n_a}$ and $\mathbf{x}_b \in \mathbb{R}^{n_b}$, $n_b \geq 2$, i.e., $\mathbf{x} = [\mathbf{x}_a^{\mathrm{T}}, \mathbf{x}_b^{\mathrm{T}}]^{\mathrm{T}}$ with $n_a + n_b = n_x$, where only the part $\mathbf{x}_a$ is directly observable through $\mathbf{y}$. More specifically,

$$\begin{aligned}\mathbf{y} &= \mathbf{g}\left(\left[\begin{array}{c} \mathbf{x}_a \\ \hline \mathbf{x}_{b,1} \end{array}\right]\right) \\ &= \mathbf{g}\left(\left[\begin{array}{c} \mathbf{x}_a \\ \hline \mathbf{x}_{b,2} \end{array}\right]\right), \quad \forall \mathbf{x}_{b,1} \in \mathbb{R}^{n_b}, \quad \mathbf{x}_{b,2} \in \mathbb{R}^{n_b}. \end{aligned} \tag{49}$$

To show, that some rotation matrices do not affect the σ-points $\mathcal{Y}_i = \mathbf{g}(\mathcal{X}_i^r)$, $i = 0, \ldots 2n_x$, it is convenient to consider $\mathbf{S}^{\mathbf{xx}}$ being obtained by the Cholesky decomposition for which the independence will be proven. Later, the rotations that do not affect the σ-points $\mathcal{Y}_i$ will be specified for a general decomposition.

The decomposition $\mathbf{S}^{\mathbf{xx}}$ obtained by the Cholesky decomposition has the following form

$$\mathbf{S}_{\mathrm{Ch}}^{\mathbf{xx}} = \left[\begin{array}{c|c} \mathbf{L}_a & \mathbf{0}_{n_a \times n_b} \\ \hline \mathbf{F} & \mathbf{L}_b \end{array}\right], \tag{50}$$

where $\mathbf{L}_a \in \mathbb{R}^{n_a \times n_a}$ and $\mathbf{L}_b \in \mathbb{R}^{n_b \times n_b}$ are lower triangular matrices and $\mathbf{F} \in \mathbb{R}^{n_b \times n_a}$ is a full-rank matrix. Note that the notation $\mathbf{S}_{\mathrm{Ch}}^{\mathbf{xx}}$ emphasises that it is the Cholesky decomposition. The σ-points $\mathcal{X}_{i,a}^r$ related to $\mathbf{x}_a$ that do affect the σ-points $\mathcal{Y}_i$ are given by

$$\mathcal{X}_{0,a}^r = \hat{\mathbf{x}}_a, \tag{51}$$

$$\mathcal{X}_{j,a}^r = \hat{\mathbf{x}}_a + \sqrt{(n_x + \kappa)}[\mathbf{L}_a, \mathbf{0}_{n_a \times n_b}]\mathbf{e}_j, \tag{52}$$

$$\mathcal{X}_{n_x+j,a}^r = \hat{\mathbf{x}}_a - \sqrt{(n_x + \kappa)}[\mathbf{L}_a, \mathbf{0}_{n_a \times n_b}]\mathbf{e}_j, \tag{53}$$

where $j = 1, \ldots, n_x$, $\mathbf{e}_j$ is the jth column of $\mathbf{I}_{n_x}$ and $\hat{\mathbf{x}} = [\hat{\mathbf{x}}_a^{\mathrm{T}}, \hat{\mathbf{x}}_b^{\mathrm{T}}]^{\mathrm{T}}$.

Now, consider an arbitrary n_x-dimensional rotation matrix $\mathbf{C}'$ that rotates in $\mathbb{R}^{n_b}$ while the space $\mathbf{R}^{n_a}$ is unaffected. Such rotation matrix can be written as

$$\mathbf{C}' = \left[\begin{array}{c|c} \mathbf{I}_{n_a} & \mathbf{0}_{n_a \times n_b} \\ \hline \mathbf{0}_{n_b \times n_a} & \mathbf{C}_b \end{array}\right]. \tag{54}$$

Note that the rotation matrix of this form is the first rotation matrix in (48) with rotation angle θ_6.

The Cholesky decomposition $\mathbf{S}_{\mathrm{Ch}}^{\mathbf{xx},r}$ rotated by $\mathbf{C}'$ is given by

$$\mathbf{S}_{\mathrm{Ch}}^{\mathbf{xx},r} = \mathbf{S}_{\mathrm{Ch}}^{\mathbf{xx}}\mathbf{C}' = \left[\begin{array}{c|c} \mathbf{L}_a & \mathbf{0}_{n_a\times n_b} \\ \hline \mathbf{F} & \mathbf{L}_b \end{array}\right] \left[\begin{array}{c|c} \mathbf{I}_{n_a} & \mathbf{0}_{n_a\times n_b} \\ \hline \mathbf{0}_{n_b\times n_a} & \mathbf{C}_b \end{array}\right] = \left[\begin{array}{c|c} \mathbf{L}_a & \mathbf{0}_{n_a\times n_b} \\ \hline \mathbf{F} & \mathbf{L}_b\mathbf{C}_b \end{array}\right]. \quad (55)$$

From (55) it can be seen that the rotation matrix $\mathbf{C}'$ in (54) does not affect the first n_a rows of $\mathbf{S}_{\mathrm{Ch}}^{\mathbf{xx}}$. Hence, the rotation matrix $\mathbf{C}'$ does not affect the σ-points $\mathcal{X}_{i,a}^r$ and consequently the σ-points $\mathcal{Y}_i$, the mean $\hat{\mathbf{y}}^{\mathrm{UT}}(\mathbf{C}')$ and the covariance matrix $\mathbf{P}^{\mathbf{yy},\mathrm{UT}}(\mathbf{C}')$. This implies that rotation matrices in the form (54) do not have to be considered for optimisation. This gives $n_b(n_b-1)/2$ rotation angles that do not have to be optimised. Hence, the dimension of the optimisation space is now reduced to

$$n_\theta = [n_x(n_x-1)/2] - [n_b(n_b-1)/2]. \quad (56)$$

A general decomposition $\mathbf{S}^{\mathbf{xx}}$ can be written as $\mathbf{S}^{\mathbf{xx}} = \mathbf{S}_{\mathrm{Ch}}^{\mathbf{xx}}\bar{\mathbf{C}}$, where $\bar{\mathbf{C}}$ is an n_x dimensional rotation matrix. Then, the rotation matrices unnecessary for optimisation are given by $\bar{\mathbf{C}}^{\mathrm{T}}\mathbf{C}'$ as the rotated decomposition $\mathbf{S}^{\mathbf{xx},r}$ is given by

$$\mathbf{S}^{\mathbf{xx},r} = \mathbf{S}^{\mathbf{xx}}\bar{\mathbf{C}}^{\mathrm{T}}\mathbf{C}' = \mathbf{S}_{\mathrm{Ch}}^{\mathbf{xx}}\bar{\mathbf{C}}\bar{\mathbf{C}}^{\mathrm{T}}\mathbf{C}' = \mathbf{S}_{\mathrm{Ch}}^{\mathbf{xx}}\mathbf{I}_{n_x}\mathbf{C}', \quad (57)$$

in which case again the σ-points $\mathcal{X}_{i,a}^r$ are unaffected by the rotation $\bar{\mathbf{C}}^{\mathrm{T}}\mathbf{C}'$.

However, for the considered decrease of n_θ, a direct usage of the Cholesky decomposition $\mathbf{S}_{\mathrm{Ch}}^{\mathbf{xx}}$ is convenient due to parametrisation of the rotation matrix $\mathbf{C}$ by angles $\boldsymbol{\theta}$ as is shown in Section 3.5.

The proposed lossless decrease of optimisation space dimension does not affect the smallest error of $\hat{\mathbf{y}}^{\mathrm{UT}}(\mathbf{C})$ and $\mathbf{P}^{\mathbf{yy},\mathrm{UT}}(\mathbf{C})$ attainable by the optimisation. However, it does affect the cross-covariance matrix $\mathbf{P}^{\mathbf{xy}}$ (10). Therefore, when applied in the UKF, the predictive state σ-points $\mathcal{X}_{i,k|k-1}^r$ affect through $\mathbf{P}_{k|k-1}^{\mathbf{xz}}$ the gain $\mathbf{K}_k$ of the filter.

4.2. Lossy optimisation space dimension decreasing

For reduction of the computational costs, it is advisable to consider only the rotation angles that significantly affect the criterion value. For such analysis the usage of the criterion (43) is opportune. To set up an order of elements of $\mathbf{x}$ according to their influence on (43), the value $J_i^4(\mathbf{C})$ is calculated for $i = 1,\ldots,n_x$

$$J_i^4(\mathbf{C}) = [\tfrac{1}{4!}\mathbf{g}^{(4)}(\hat{\mathbf{x}})\tilde{\mathbf{M}}^{\mathbf{x}(4)}(\mathbf{C})\mathbf{J}_i]^{\mathrm{T}}\mathbf{W}[\tfrac{1}{4!}\mathbf{g}^{(4)}(\hat{\mathbf{x}})\tilde{\mathbf{M}}^{\mathbf{x}(4)}(\mathbf{C})\mathbf{J}_i] \quad (58)$$

where $\mathbf{J}_i = \mathrm{diag}[(\mathbf{e}_i)^{\otimes 4}]$ is a matrix selecting the element of $\tilde{\mathbf{M}}^{\mathbf{x}(4)}(\mathbf{C})$ corresponding to the ith element of $\mathbf{x}$. Then, the values $J_i^4(\mathbf{C})$, $i = 1,\ldots,n_x$ are sorted and the order and individual values provide information of magnitude of influence of $\mathbf{x}$ elements on the criterion. Using this information the user can decide over which rotation angles the chosen criterion should be optimised.

Note that for calculation of (58) any rotation matrix $\mathbf{C}$ can be used, e.g., $\mathbf{C} = \mathbf{I}$.

5. UKF WITH ROTATED σ-POINT SET

The concept of the σ-point set rotation can directly be extended from the UT to the UKF. The UKF with the rotated σ-point set has the same structure as the UKF given by Algorithm 1, where the filtering state σ-points and the predictive state σ-points respect the optimal rotation matrices.

When finding an optimal rotation matrix for the predictive state σ-points in the filtering step both criteria (43) and (44) can be used as the measurement $\mathbf{z}_k$ is available and can be used in place of the sample $\mathbf{y}_s$ in (44). To find an optimal rotation matrix for the filtering state σ-points in the predictive step, only the criterion (43) can be used.

The UKF with optimised rotation of the σ-point set is given by Algorithm 2.

ALGORITHM 2: *Unscented Kalman Filter with Rotated σ-point Set*

Step 1: Identical with **Step 1** of Algorithm 1.
Step 2: Find an optimal rotation matrix $\mathbf{C}_{k|k-1}$.
Step 3: Similar to **Step 2** of Algorithm 1 with $\mathbf{S}_{k|k-1}^{\mathbf{xx}}$ replaced by $\mathbf{S}_{k|k-1}^{\mathbf{xx},r} = \mathbf{S}_{k|k-1}^{\mathbf{xx}}\mathbf{C}_{k|k-1}$.
Step 4: Find an optimal rotation matrix $\mathbf{C}_{k|k}$.
Step 5: Similar to **Step 3** of Algorithm 1 with $\mathbf{S}_{k|k}^{\mathbf{xx}}$ replaced by $\mathbf{S}_{k|k}^{\mathbf{xx},r} = \mathbf{S}_{k|k}^{\mathbf{xx}}\mathbf{C}_{k|k}$.

Let $k = k+1$. The algorithm continues by **Step 2**.

The rotation matrices $\mathbf{C}^{*k|k}$ and $\mathbf{C}^{*k|k-1}$ can be in principle selected on on-line or off-line basis.

5.1. Off-line computed time-invariant and time-varying rotation matrix

The simplest choice of the rotation matrices is $\mathbf{C}_{k|k} = \mathbf{C}_{k|k-1} = \mathbf{I}_{n_x}$, $\forall k$. This corresponds to utilisation of a constant rotation of the σ-points that is determined by the selected covariance matrix decomposition technique.

Alternatively, a constant rotation matrix (or matrices different for the filtering and prediction step) might be found by e.g., a prior analysis of a considered problem as outlined in [6].

Another possibility is to find a sequence (or sequences) of the rotation matrices for a considered scenario based on a prior Monte-Carlo (MC) analysis. This approach is suitable for systems with set-up allowing periodical repetition of their dynamic behaviour. As an example, during the landing phase all aircrafts of a given category follow the same path (trajectory). Nevertheless, once the set-up is changed (either trajectory, sensor location, etc.) the procedure of finding the rotation matrices must be repeated. This is illustrated in numerical simulations in Section 6.

For the prior analysis either criterion (36), (43) or (44) may be used. Note that the rotation matrices $\mathbf{C}_{k|k-1}$, $\mathbf{C}_{k|k}$, if computed off-line, are not conditioned by the

past measurements of considered experiment as the notation indicates. The notation is, however, kept to be sufficiently general to cover also the possibility for the on-line computation of the matrices.

The UKF given by Algorithm 2 with rotation matrices $\mathbf{C}_{k|k-1}$ and $\mathbf{C}_{k|k}$ optimised off-line by means of a prior analysis will be denoted as trained-sigma-point-set-UKF (TUKF).

5.2. On-line computed time-varying rotation matrix

The on-line computation of the rotation matrix (or matrices) respects the current status of the filter. This procedure can be understood as a rotation matrix optimisation and in fact, a similar approach, in principle, has been used for the scaling parameter adaptation [33], [4] and is adopted here.

The rotation matrices used in the filtering and prediction steps, i.e., $\mathbf{C}_{k|k-1}$ and $\mathbf{C}_{k|k}$, can be computed by minimising (43) or (44).

The UKF given by Algorithm 2 with rotation matrices $\mathbf{C}_{k|k-1}$ and $\mathbf{C}_{k|k}$ optimised on-line using either (43) or (44) will be denoted as adaptive-sigma-point-set-UKF (AUKF).

5.3. Notes

Note 1: The on-line optimisation of the rotation matrix is clearly the most computationally demanding part of the algorithm. Moreover, the dimension of the optimisation space n_θ grows quadratically with the state dimension (see (47)). Selection of the rotation angles for optimisation and the optimisation technique is thus crucial. Two techniques for optimisation space dimension n_θ decrease have been proposed in Section 4. A brief discussion regarding the suitable optimisation technique can be found in [10]. In this paper, the grid method is preferred due to its simplicity. The method is summarised in Appendix A.

Note 2: The order of the successive rotations affects the overall rotation matrix. However, if a sufficiently dense grid is assumed, then the impact of the rotation order in the optimisation can be neglected.

Note 3: The rotation matrix optimisation might be performed independently in the predictive and the filtering steps. However, the impact of the adaptation in one step is often dominant and the adaptation in the second step can be skipped to reduce computational costs.

Note 4: The σ-point set rotation does not affect the UT performance if the function $\mathbf{g}(\cdot)$ in (3) is linear. Therefore, it is reasonable to supplement the UKF adaptively rotating of the σ-point set with an algorithm evaluating the severity of the nonlinearities at the actual working points. If the nonlinearity is mild at a given time, the adaptive selection of the rotation matrix might be skipped without any significant impact on the estimation performance. Such nonlinearity measures were proposed and integrated with local filters in [33].

Note 5: The paper considered the system (1) and (2) with additive noises. However, the concept can readily be extended for systems with non-additive noises. Just the optimisation space for rotations would be higher dimensional due to the state augmentation by the noises.

Note 6: Although the concept of the σ-point set rotation was discussed in the UKF framework, the σ-point set rotation adaptation can be used with any deterministic σ-point set based local filter.

6. NUMERICAL ILLUSTRATION

The performance of the UT and the UKF is affected by the σ-point set rotation and scaling. In this section, the UT and UKF are evaluated each using two examples mainly with respect to the σ-point set rotation influence.

6.1. Static example I: Fourth-order polynomial

Consider a random variable nonlinear transformation (as defined by (3)–(10)) with

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathbf{P}^{\mathbf{xx}} = \begin{bmatrix} 4 & 0.8 \\ 0.8 & 10 \end{bmatrix} \tag{59}$$

and

$$y = g(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{x}\mathbf{x}^{\mathrm{T}}\mathbf{x}. \tag{60}$$

In Fig. 2, the impact of the σ-point set rotation through an angle θ on the approximate characteristics of y are illustrated. The characteristics are computed using the UT (8)–(10) with $\mathbf{S}^{\mathbf{xx}}$ computed by the SVD,[7] scaling parameters $\kappa = 1$ and $\kappa = 2$, and the rotation matrix used in (30)–(33) of the form

$$\mathbf{C} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}. \tag{61}$$

Besides these characteristics, the true ones and the ones approximated using the UT with

- a recommended fixed $\kappa = 1$,
- a fixed $\kappa = 2$,

both with $\theta = 0$ [deg], are plotted in Fig. 2. Here, it should be noted, that the true and fixed UT characteristics are not functions of the rotation parameter θ, they are plotted for the whole range of the parameter just for ease of comparison. The figure shows a significant dependency of the UT performance on the σ-point set rotation. As shown in (42) under assumption of the Gaussian PDF of $\mathbf{x}$, the set rotation affects fourth- and higher-order even terms of the TE of the UT-based mean computation. Considering the mean value computation, i.e., $\hat{y}^{\mathrm{UT}}$ (8), in this polynomial example, the fourth-order term of the TE is the only term depending on the rotation. Therefore, the error (42) of $\hat{y}^{\mathrm{UT}}$ is equal to

$$\tilde{y}(\mathbf{C}) = \tfrac{1}{4!}\mathbf{g}^{(4)}(\hat{\mathbf{x}})(\mathbf{M}^{\mathbf{x}(4)} - \mathbf{M}^{\mathbf{x}(4),\mathrm{UT}}(\mathbf{C})), \tag{62}$$

[7] Throughout this section, the SVD of the covariance matrix $\mathbf{P}^{\mathbf{xx}}$ is considered. In this case, the σ-points lie on the principal axes of the covariance matrix ellipsoid [34].

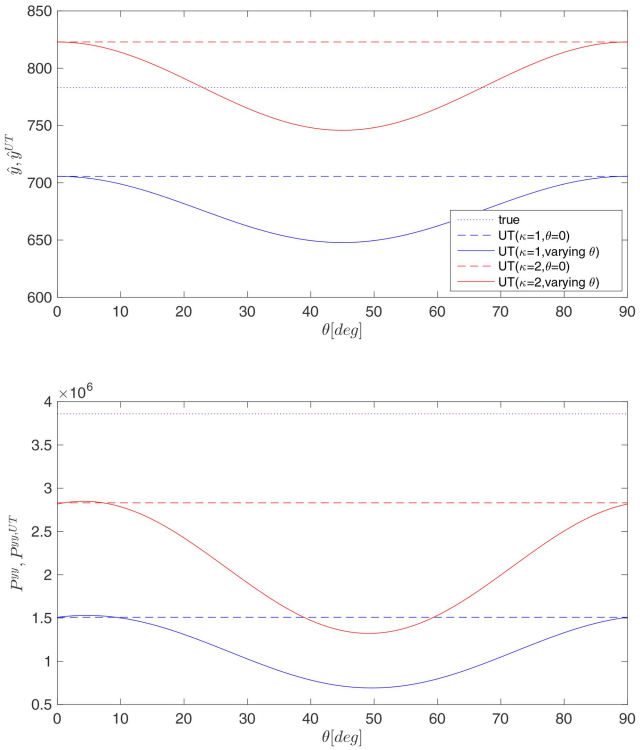


Fig. 2. Static example I: Dependence of the UT-based approximate characteristics on the rotation of σ-point sample set.

where

$$\mathbf{g}^{(4)}(\hat{\mathbf{x}}) = \left[\frac{\partial^4 g(\mathbf{x})}{\partial x_1^4}, \frac{\partial^4 g(\mathbf{x})}{\partial x_1^3 \partial x_2}, \frac{\partial^4 g(\mathbf{x})}{\partial x_1^3 \partial x_2}, \frac{\partial^4 g(\mathbf{x})}{\partial x_1^2 \partial x_2^2}, \frac{\partial^4 g(\mathbf{x})}{\partial x_1^3 \partial x_2}, \dots \right.$$
$$\frac{\partial^4 g(\mathbf{x})}{\partial x_1^2 \partial x_2^2}, \frac{\partial^4 g(\mathbf{x})}{\partial x_1^2 \partial x_2^2}, \frac{\partial^4 g(\mathbf{x})}{\partial x_1 \partial x_2^3}, \frac{\partial^4 g(\mathbf{x})}{\partial x_1^3 \partial x_2}, \frac{\partial^4 g(\mathbf{x})}{\partial x_1^2 \partial x_2^2}, \dots$$
$$\left. \frac{\partial^4 g(\mathbf{x})}{\partial x_1^2 \partial x_2^2}, \frac{\partial^4 g(\mathbf{x})}{\partial x_1 \partial x_2^3}, \frac{\partial^4 g(\mathbf{x})}{\partial x_1^2 \partial x_2^2}, \frac{\partial^4 g(\mathbf{x})}{\partial x_1 \partial x_2^3}, \frac{\partial^4 g(\mathbf{x})}{\partial x_1 \partial x_2^3}, \frac{\partial^4 g(\mathbf{x})}{\partial x_2^4} \right]$$
$$= [24,0,0,8,0,8,8,0,0,8,8,0,8,0,0,24]. \quad (63)$$

Hence the criterion (43) is a function of the fourth moments only and $\mathbf{g}^{(4)}(\hat{\mathbf{x}})$. Thus, it is equal to (36).

Value of the criterion (43) with $\mathbf{W} = 1$ as a function of the rotation θ is shown in Fig. 3 for both setting of κ.

6.2. Static example II: Arctangent

Arctangent is a nonlinear transformation appearing in tracking[8] applications for a conversion between Cartesian and polar coordinates. Contrary to the previous polynomial function, arctangent cannot by expressed by the TE with a finite number of terms.

Let a random variable nonlinear transformation with

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = \begin{bmatrix} 10 \\ 1 \end{bmatrix}, \quad \mathbf{P}^{\mathbf{xx}} = \begin{bmatrix} 4 & 0.8 \\ 0.8 & 10 \end{bmatrix} \quad (64)$$

and

$$y = g(\mathbf{x}) = \operatorname{atan}(x_2/x_1). \quad (65)$$

[8]The four-quadrant inverse tangent function (atan2) is considered.

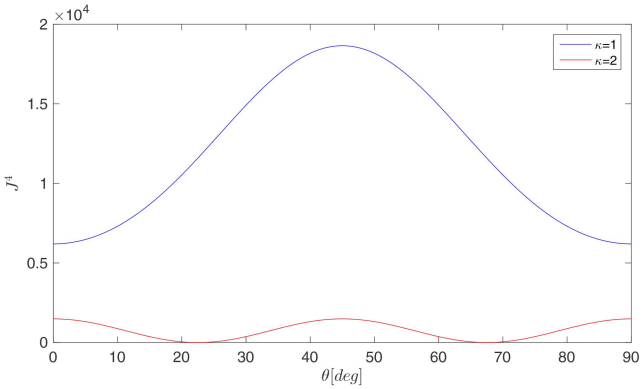


Fig. 3. Static example I: Dependence of the UT-based mean error on the rotation of σ-point sample set (for $\kappa = 1$).

be considered. In Fig. 4, the impact of both parameters affecting the UT performance, i.e., κ and θ, is illustrated for the mean $\hat{y}^{\mathrm{UT}}$ and variance $P^{yy,\mathrm{UT}}$ computation together with the true characteristics of y and the ones approximated using the UT with fixed $\kappa = 1$, and $\theta = 0$ [deg].

Fig. 4 indicates that both user-defined parameters (scaling and rotation) heavily impact the UT-approximated mean value $\hat{y}^{\mathrm{UT}}$ and the variance $P^{yy,\mathrm{UT}}$. By a suitable rotation, it is possible to get the true values of the mean and variance of y (for mean $\theta \approx 31$ and 79 [deg], for variance $\theta \approx 10$ and 58 [deg]; the values are highlighted by the vertical red lines). On the other hand, in this example it is not possible to get the true values of the statistics by any selection of the scaling parameter κ.

The value of the criterion $J(\mathbf{C}(\theta))$ (36) with $\mathbf{W} = 1$ is plotted in Fig. 5. The value is slightly different from the fourth-order moment based criterion $J^4(\mathbf{C}(\theta))$ (43) as, in this case, the rotation affects not only the fourth-order term of the TE but also all remaining higher-order even terms. However, the criterion $J^4(\mathbf{C}(\theta))$ (which can be computed without the knowledge of the true mean $\hat{y}$) still represents a reasonable approximation of the criterion $J(\mathbf{C}(\theta))$ (which cannot be computed without the knowledge of the true mean $\hat{y}$).

6.3. Dynamic example I: Bearings-only tracking

The impact of the σ-point set rotation either fixed or adaptive on the UKF performance is illustrated using the bearings-only tracking example where a manoeuvring object is tracked by a radar platform [25]. The object follows a course of -140 [deg] (the angles are referenced clockwise positive to the y axis) starting 12.2 [km] away from the platform at a constant speed of 4 [knots]. The platform follows a course of 140 [deg] at a constant speed of 5 [knots] and at the time interval $k = \langle 13,17 \rangle$ executes a manoeuvre to reach a new course of 18 [deg]. The initial positions are [12,2] [km] for the object and [0,0] [km] for the platform. The geometry of the motion is depicted in Fig. 6. The object

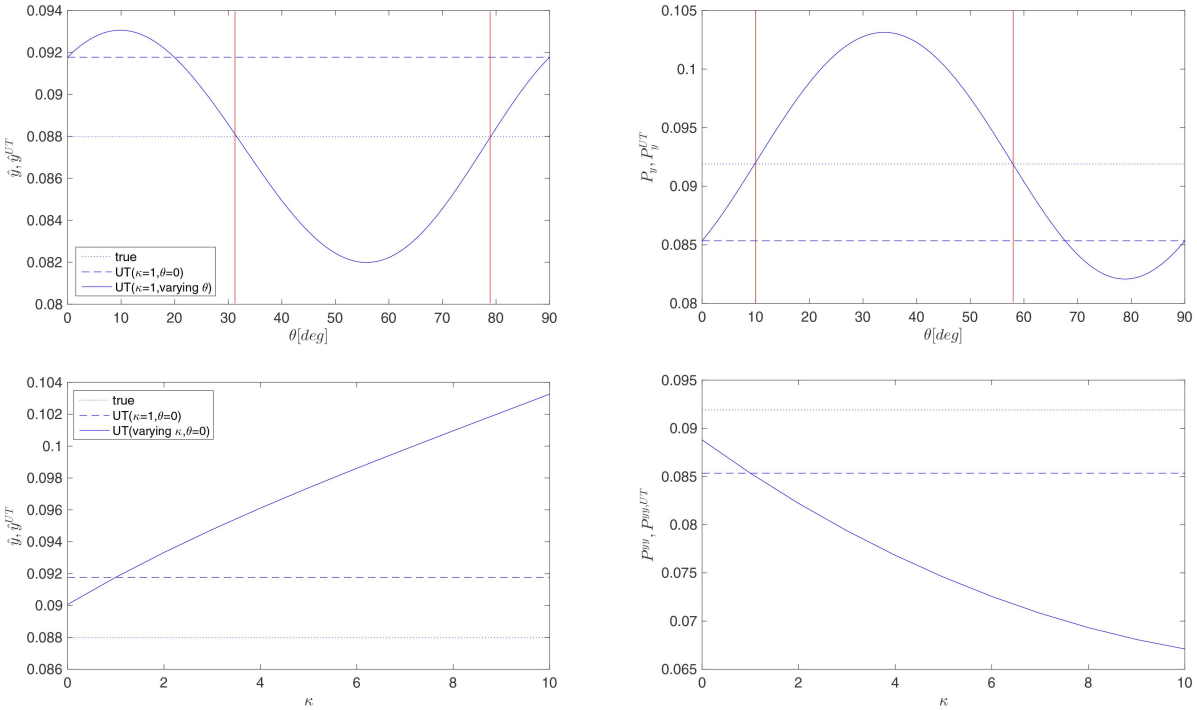


Fig. 4. Static example II: Dependence of the UT-based approximate characteristics on the rotation and scaling of σ-point sample set (together with the true and fixed UT characteristics).

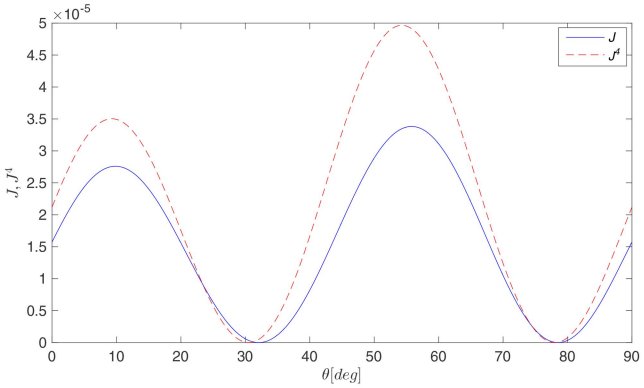


Fig. 5. Static example II: Dependence of the UT-based mean error on the rotation of σ-point sample set (for $\kappa = 1$).

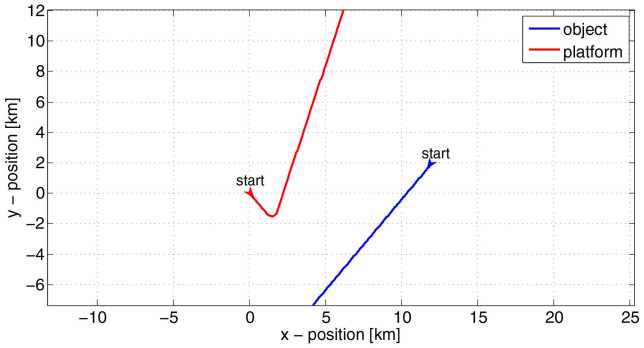


Fig. 6. Geometry of bearings-only tracking example.

motion (relative to the platform) is modelled by a continuous white noise acceleration model [3]. The state of the model $\mathbf{x}_k = [x_{1,k}, x_{2,k}, x_{3,k}, x_{4,k}]^{\mathrm{T}}$ consists of the positions in x and y-directions ($[x_{1,k}, x_{2,k}]$) and respective velocities ($[x_{3,k}, x_{4,k}]$) which evolves as

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{w}_k, \tag{66}$$

with

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 0.5T^2 & 0 \\ 0 & 0.5T^2 \\ T & 0 \\ 0 & T \end{bmatrix}, \tag{67}$$

where $T = 1$ [min] is the sampling interval, $k = 0, 1, \ldots, K = 100$, and $\mathbf{w}_k$ is the Gaussian zero-mean state noise with covariance matrix $\mathbf{\Sigma}^w$ with $\mathbf{\Sigma}^w = 10^{-4}\mathbf{I}_2$ $\times[\mathrm{km}^2/\mathrm{sec}^4]$.

The measurement z_k providing the relative angle of the object w.r.t. the platform at time k is

$$z_k = \arctan \frac{x_{1,k} - x^p_{1,k}}{x_{2,k} - x^p_{2,k}} + v^\theta_k, \tag{68}$$

where $[x^p_{1,k}, x^p_{2,k}]$ is the known platform position and the variance of the measurement noise is $\Sigma^v = (3\ [\mathrm{deg}])^2$.

In total $M = 10^3$ MC simulations were carried out and the UKF with different settings of the user-defined parameters were simulated. In particular, the following UKFs and CKFs are considered

- UKF(svd)—UKF with $\mathbf{S}^{\mathbf{xx}}$ computed by the SVD,
- UKF(chol)—UKF with $\mathbf{S}^{\mathbf{xx}}$ computed by the Cholesky decomposition,

- AUKF(θ_{12},$J^4(\mathbf{C})$)—AUKF with the Cholesky decomposition and rotation matrix computed on the basis of θ_{12} with the criterion $J^4(\mathbf{C})$ (43),
- AUKF(θ_{12})—AUKF with the Cholesky decomposition and rotation matrix computed on the basis of θ_{12} with the criterion $J^{\mathrm{MS}}(\mathbf{C})$ (46),
- AUKF(θ_{34})—AUKF with the Cholesky decomposition and rotation matrix computed on the basis of θ_{34} with the criterion $J^{\mathrm{MS}}(\mathbf{C})$ (46),
- AUKF($\boldsymbol{\theta}_{12,23,34}$)—AUKF with the Cholesky decomposition and rotation matrix computed on the basis of $\boldsymbol{\theta}_{12,23,34}$ with the criterion $J^{\mathrm{MS}}(\mathbf{C})$ (46),
- AUKF($\boldsymbol{\theta}_{all}$)—AUKF with the Cholesky decomposition and rotation matrix computed on the basis of $\boldsymbol{\theta}_{all}$ with the criterion $J^{\mathrm{MS}}(\mathbf{C})$ (46),
- TUKF—UKF with the Cholesky decomposition and off-line identified sequence of rotation matrices,
- CKF5(svd)—CKF of the fifth-order proposed in [15] with $\mathbf{S}^{\mathbf{xx}}$ computed by the SVD,
- CKF5(rot)—CKF5 with the SVD and fixed rotation of the σ-points through 30 [deg],
- ACKF5(θ_{12})—CKF with the SVD and rotation matrix computed on the basis of θ_{12} with the criterion $J^{\mathrm{MS}}(\mathbf{C})$ (46),

where θ_{12} denotes the rotation in the plane of the first two state vector elements, i.e., the respective rotation matrix is of the form

$$\mathbf{C}(\theta_{12}) = \begin{bmatrix} \cos(\theta_{12}) & -\sin(\theta_{12}) & 0 & 0 \\ \sin(\theta_{12}) & \cos(\theta_{12}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{69}$$

etc., and $\boldsymbol{\theta}_{all} = [\theta_{23},\theta_{12},\theta_{13},\theta_{14},\theta_{24}]$. It means that rotation θ_{12} rotates the part of sigma points relative to the position, where as θ_{34} rotates relative to the velocity components. Note that all the considered filters were run with the recommended scaling parameter $\kappa = 0$.

Hence, the filters AUKF(θ_{12},$J^4(\mathbf{C})$), AUKF(θ_{12}), AUKF(θ_{34}), and AUKF($\boldsymbol{\theta}_{12,23,34}$) used the lossy decrease of n_θ proposed in Section 4.2 with $n_\theta = 1$, $n_\theta = 1$, $n_\theta = 1$, and $n_\theta = 3$, respectively. The filter AUKF($\boldsymbol{\theta}_{all}$) used the lossless decrease of n_θ proposed in Section 4.1 and thus ignored rotation with the angle θ_{34}.

Note that the filter AUKF(θ_{34}) was used to demonstrate that the rotation computed on the basis of θ_{34}, i.e., in the plane defined by the state elements that are not directly measurable, has almost no effect on the estimation quality.

The AUKFs use the grid optimisation technique [10] with the grid defined as $\theta_{ij} = \{0,15,\ldots,75\}$ [deg], $\forall i,j$ which results in 6 different values used in optimisation for the AUKF(θ_{12}), AUKF(θ_{34}), in $6^3 = 216$ values for the AUKF($\theta_{12,23,34}$), and in $6^5 = 7776$ for the AUKF($\boldsymbol{\theta}_{all}$). The grid optimisation method is further discussed in Appendix A.

The TUKF takes advantage of the optimal rotation θ_{12} found prior to the estimation experiment on the basis of a set of MC simulations. In particular 10^3 simulations of the AUKF(θ_{12}) were carried out and the average (sample-based) optimal rotation angle was computed as

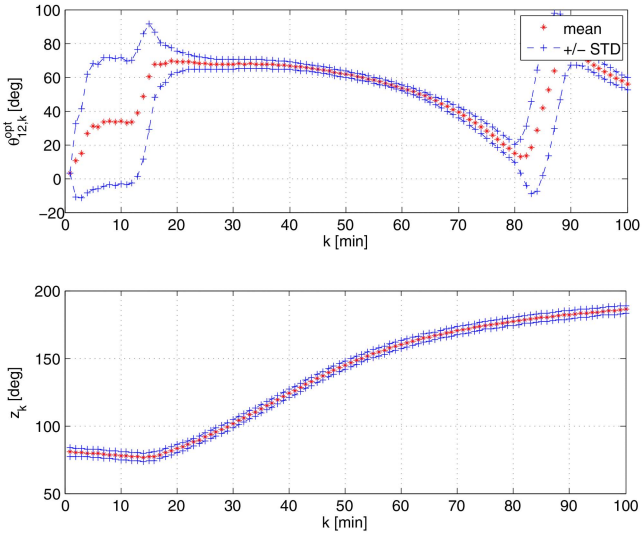


Fig. 7. Dynamic example I: Averaged optimal rotation angle θ_{12} used in TUKF and the averaged measurement sequence.

$$\theta^{\mathrm{opt}}_{12,k|k-1} = \frac{1}{10^3}\sum_{i=1}^{10^3}\theta^{(i)*}_{12,k|k-1}, \quad \forall k, \tag{70}$$

where $\theta^{(i)*}_{12,k|k-1}$ is the optimal rotation at time k of ith MC simulation. The average optimal rotation was used for the rotation matrix $\mathbf{C}^{\mathrm{opt}}_{k|k-1}(\theta_{12})$ computation in the TUKF. The optimal rotation angle is plotted in Fig. 7 together with the respective sample standard deviation (STD). In Fig. 7 the sample statistics of the measurement z_k, i.e., the mean and STD, over the MC simulations are given as well.

The filters were initialised according to [25] with initial range pdf $p(r) = \mathcal{N}\{r;\sqrt{12^2+2^2},4^2\}$ and a speed pdf $p(s) = \mathcal{N}\{s;\bar{s},4^2\}$, where $\bar{s}$ is the true speed.

The filter results were compared using the root mean square error (RMSE) defined as

$$\mathrm{RMSE}^{\mathrm{pos}}_k = \sqrt{\frac{1}{M}\sum_{m=1}^{M}(\hat{x}^{(m)}_{1,k|k} - x^{(m)}_{1,k})^2 + (\hat{x}^{(m)}_{2,k|k} - x^{(m)}_{2,k})^2} \tag{71}$$

for the position error and

$$\mathrm{RMSE}^{\mathrm{vel}}_k = \sqrt{\frac{1}{M}\sum_{m=1}^{M}(\hat{x}^{(m)}_{3,k|k} - x^{(m)}_{3,k})^2 + (\hat{x}^{(m)}_{4,k|k} - x^{(m)}_{4,k})^2} \tag{72}$$

for the velocity error and using the non-credibility index (NCI) [20] defined as

$$\mathrm{NCI}_k = \frac{1}{M}\sum_{m=1}^{M}[10\log_{10}((\tilde{\mathbf{x}}^{(m)}_{k|k})^{\mathrm{T}}(\mathbf{P}^{\mathbf{xx}(m)}_{k|k})^{-1}\tilde{\mathbf{x}}^{(m)}_{k|k}) - 10\log_{10}((\tilde{\mathbf{x}}^{(m)}_{k|k})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}_k\tilde{\mathbf{x}}^{(m)}_{k|k})], \tag{73}$$

TABLE I
Dynamic example I: RMSE, NCI, and time results of bearings-only tracking.

	UKF(svd)	UKF(chol)	AUKF(θ_{12})	AUKF(θ_{12}, $J^4(\mathbf{C})$)	AUKF(θ_{34})	AUKF($\theta_{12,13,24}$)	AUKF($\boldsymbol{\theta}_{all}$)	TUKF
RMSE^{pos} [km]	3.30	3.40	2.90	2.80	3.30	2.70	2.70	3.10
RMSE^{vel} [km/sec]	0.16	0.16	0.15	0.15	0.15	0.15	0.14	0.16
NCI	4.60	4.80	3.80	3.80	4.60	3.40	2.90	4.40
time [msec]	12.8	9.5		86		3175	104191	10.6

	CKF5(svd)	CKF5(rot)	ACKF5(θ_{12})
RMSE^{pos} [km]	3.10	3.50	2.80
RMSE^{vel} [km/sec]	0.15	0.16	0.14
NCI	4.15	5.10	3.50
time [msec]	13.30		93

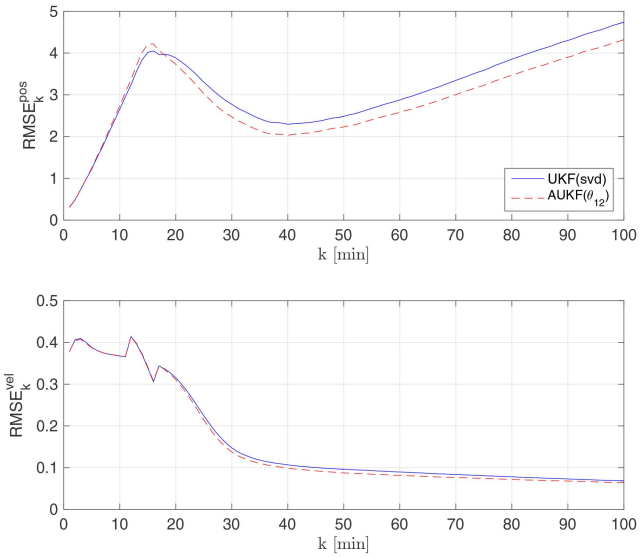


Fig. 8. Dynamic example I: Time behaviour of the RMSE for UKF(svd) and the AUKF(θ_{12}).

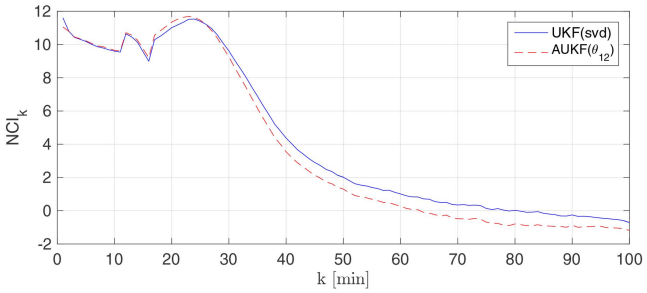


Fig. 9. Dynamic example I: Time behaviour of the NCI for UKF(svd) and the AUKF(θ_{12}).

where $\tilde{\mathbf{x}}_{k|k}^{(m)} \triangleq (\mathbf{x}_k^{(m)} - \hat{\mathbf{x}}_{k|k}^{(m)})$, $\mathbf{x}_k^{(m)}$ is the true state, $\hat{\mathbf{x}}_{k|k}^{(m)}$ and $\mathbf{P}_k^{\mathbf{xx}(m)}$ are the filtering mean and covariance matrix of the estimate provided by the filter at the mth MC run and $\boldsymbol{\Sigma}_k$ is the mean square error. Whereas the RMSE compares just the quality of the mean estimate (the lower RMSE value, the better performance), the NCI assesses the credibility of the estimate, i.e., whether the error covariance matrix of the filter corresponds to the real mean square error of the state estimate. The NCI value should be ideally zero. Negative NCI values indicate pessimistic estimates of the filter, positive NCI values, on the other hand, imply optimistic estimates. The averaged values of the RMSE and the NCI over all time instants are given in Table I. Illustration of time behaviour of the RMSE and the NCI for the UKF(svd) and the AUKF(θ_{12}) is depicted in Fig. 8 and Fig. 9.

It can be seen that the UKF(svd) slightly outperforms the UKF(chol) at the cost of mild increase of the computational complexity. The further improvement can be reached by using the off-line found averaged optimal rotation $\theta_{12,k|k-1}^{\text{opt}}$ with negligible computational requirements increase.

The results for the AUKFs confirm the theoretical analysis of Section 4.1 that the largest impact on the estimation quality is tied with the rotation in the planes defined by the state components which are "directly" measured, i.e., the positions in this example. The major improvement is caused by the rotation in the position plane, i.e., through the angle θ_{12}. Then, the additional rotations through the angles θ_{23} and θ_{13} reduces the RMSE and the NCI further. Adding the adaptive selection of the optimal rotation also in the remaining directions (five in total) brings almost no benefit. This is also confirmed by the results of the AUKF(θ_{34}) which indicates that the rotation in the plane defined by the velocities in x and y directions does not lead to any performance improvement.

Concerning the optimisation criteria, there is no significant difference in using the criterion $J^4(\mathbf{C})$ (43) or the criterion $J^{\text{MS}}(\mathbf{C})$ (46). With respect to the fact that the criterion $J^4(\mathbf{C})$ requires computation and evaluation of the fourth-order derivatives, the criterion $J^{\text{MS}}(\mathbf{C})$ seems to be more suitable in the considered set-up.

From the results it is evident that the lossy decrease of n_θ may save a considerable amount of computational costs. If only the directly observable state elements are considered in the lossy decrease of n_θ, the decrease of estimate quality is negligible in this illustration.

TABLE II
Dynamic example II: RMSE and NCI of two-dimensional example.

	UKF	AUKF(15[deg])	AUKF(30[deg])
RMSE	14.70	8.90	8.95
NCI	11.10	11.00	11.10

For the sake of completeness, the fifth-order CKF, the CKF5 [15], was tested as well. Compared to the UKF, the CKF5 processes $2n_x^2 + 1$ σ-points. Therefore, it provides estimates of better quality, in this example, in both monitored criteria at the costs of slightly higher computational complexity. The performance of the CKF5 is, however, still affected by the rotation of the σ-point set. Fixed σ-point set rotation may improve but also worsen the estimation performance (as can be seen for CKF5(rot)). In principle, the same rotation adaptation criteria may be used as for the UKF. The CKF5 with the adaptation, the ACKF5, according to the criterion (46), then offers further improvement to the estimation performance.

6.4. Dynamic example II: Nonlinear two-dimensional system

The second dynamic example considers a nonlinear system introduced in [37] described by

$$\mathbf{x}_{k+1} = \begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \end{bmatrix} = \begin{bmatrix} 3\sin(5x_{2,k}^2) \\ x_{1,k} + e^{-0.05x_{2,k}} + 10 \end{bmatrix} + \mathbf{w}_k, \tag{74}$$

$$z_k = \cos(x_{1,k}) + x_{2,k}^2 + v_k, \tag{75}$$

with $p(\mathbf{w}_k) = \mathcal{N}\{\mathbf{w}_k : \mathbf{0}, 6\mathbf{I}_2\}$, $p(v_k) = \mathcal{N}\{v_k : 0, 1\}$, $p(\mathbf{x}_0) = \mathcal{N}\{\mathbf{x}_0 : [-0.7, 1]^{\mathrm{T}}, \mathbf{I}_2\}$, and $k = 0, 1, \ldots, 100$.

Three filters, namely

- UKF(svd),
- AUKF(15 [deg])—AUKF with the SVD with the criterion $J^{\mathrm{MS}}(\mathbf{C})$ (46) and grid-based optimisation with step of 15 [deg],
- AUKF(30 [deg])—AUKF with the SVD with the criterion $J^{\mathrm{MS}}(\mathbf{C})$ (46) and grid-based optimisation with step of 30 [deg],

all with $\kappa = 1$, were compared in terms of the averaged RMSE (71) and NCI (73). The simulation results are summarised in Table II.

Adaptation in the σ-point set rotation significantly improves the UKF performance in terms of the RMSE. The NCI remains almost unaffected, which means that the optimisation reduces the estimation error and estimated covariance matrix proportionally. The results also reveal that the AUKF is not heavily dependent on the optimisation grid density in this example. Doubling the optimisation grid points does not have almost and impact on the estimation performance while the computational requirements of AUKF(30[deg]) are half of the AUKF(15[deg]) requirements.

7. CONCLUDING REMARKS

The paper dealt with an analysis of the σ-point set rotation in the derivative-free approximations used in the local filter design, namely in the unscented transform being a cornerstone of the unscented Kalman filter. It was shown that the covariance matrix decomposition, used in σ-point computation, can be multiplied by an arbitrary rotation matrix. The matrix can be then viewed as the user-defined parameter significantly impacting the filter performance. In principle, two different approaches for selection of the appropriate rotation were proposed; off-line approach determining the optimal rotation prior to the estimation experiment (in certain recurrent scenarios even time-varying) leading to the TUKF, or an on-line approach computing the optimal rotation during the experiment respecting the actual conditions leading to the AUKF. To avoid excessive increase of computational costs due to the optimisation of the rotation angles, two techniques were proposed to decrease dimension of the optimisation space. The proposed approaches for rotation matrix optimisation were illustrated and compared in terms of the estimation performance and the computational complexity in several numerical examples.

Note that algorithms of the UKF with rotated σ-point set are part of the Nonlinear Estimation Framework available at http://nft.kky.zcu.cz/.

APPENDIX A GRID-BASED OPTIMISATION METHOD USED IN AUKF

Computationally efficient adaptation of the σ-point set rotation matrix is the key enabler allowing effective usage of the AUKFs.

Any optimisation technique used for the solution to (43) and (44) requires specification of the range for the parameters being optimised as the task is generally nonlinear. If the rotation matrix parameterisation by a sequence of the subsequent rotations is selected, then the simplest way is to define the range for each particular rotation (omitting time indices) as

$$\theta_i = \langle 0, 360 \rangle, \quad i = 1, 2, \ldots, n_\theta, \tag{76}$$

where n_θ is the number of subsequent rotations (47). However, because of the symmetry of the considered σ-point set, the interval can be significantly reduced to

$$\theta_i = \langle 0, 90 \rangle, \quad \forall i, \tag{77}$$

without the loss of generality. This is illustrated for a two- and a three-dimensional case below assuming the grid-based optimisation method used in the filtering step.

The grid-based optimisation method simply covers the interval θ_i (77) by a grid of equidistantly placed points, i.e.,

$$\theta_i^{\mathrm{grid}} = \{\theta_i^{(1)}, \theta_i^{(2)}, \ldots, \theta_i^{(G)}\}, \quad \forall i. \tag{78}$$

If $n_\theta > 1$, then the final (multidimensional) grid is given by the Cartesian product of the particular grids as

$$\boldsymbol{\theta}^{\text{grid}} = \theta_1^{\text{grid}} \times \theta_2^{\text{grid}} \times \cdots \times \theta_{n_\theta}^{\text{grid}}. \tag{79}$$

The cardinality of the grid-points set (79) is

$$n_{\text{grid}} = G^{n_\theta}, \tag{80}$$

thus exponentially growing with the number of the rotation angles θ_i.

The criterion function is then evaluated at all points of the set $\boldsymbol{\theta}^{\text{grid}}$ (79) and the optimum point is chosen according to

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}^{\text{grid}}} J(\mathbf{C}(\boldsymbol{\theta})). \tag{81}$$

The chosen vector $\boldsymbol{\theta}^*$ is subsequently used for the rotation matrix computation as illustrated by (48).

1.1. Two dimensional case

Considering $n_x = 2$, $\hat{\mathbf{x}} = \mathbf{0}_{2\times 1}$, $\mathbf{P}^{\mathbf{xx}} = \mathbf{I}_2 = [\mathbf{e}_1, \mathbf{e}_2]$, and $(n_x + \kappa) = 1$, the σ-point set computed according to (4)–(6) is

$$\mathcal{X}_0 = \mathbf{0}_{2\times 1},\ \mathcal{X}_1 = \mathbf{e}_1,\ \mathcal{X}_2 = \mathbf{e}_2,\ \mathcal{X}_3 = -\mathbf{e}_1,\ \mathcal{X}_4 = -\mathbf{e}_2, \tag{82}$$

where $\mathbf{0}_{n_x\times 1}$ is the zero matrix of the indicated dimension. The number of available rotation angles (47) is $n_\theta = 1$, therefore $\boldsymbol{\theta}^{\text{grid}} = \theta_1^{\text{grid}}$, and the respective covariance matrix $\mathbf{C}$ is computed according to (61).

Then, it is not difficult to see that the σ-point set rotation through an angle $\theta_h \in \langle 90, 180)$ [deg] results in virtually the same σ-point set as for the rotation $\theta_l = (\theta_h - 90) \in \langle 0, 90)$ [deg]. It means that

$$\mathcal{X}_0^r = \mathbf{C}(\theta_h)\mathcal{X}_0 = \mathbf{C}(\theta_l)\mathcal{X}_0 = \mathcal{X}_0, \tag{83}$$

$$\begin{aligned}\mathcal{X}_i^r &= \mathbf{C}(\theta_h)\mathcal{X}_i = \mathbf{C}(\theta_h - 90)\mathbf{C}(90)\mathcal{X}_i = \mathbf{C}(\theta_h - 90)\mathcal{X}_{i+1} \\ &= \mathbf{C}(\theta_l)\mathcal{X}_{i+1},\end{aligned} \tag{84}$$

$$\mathcal{X}_4^r = \mathbf{C}(\theta_h)\mathcal{X}_4 = \mathbf{C}(\theta_l)\mathcal{X}_1, \tag{85}$$

where $\mathbf{C}(90) = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ and $i = 1, 2, 3$.

As the σ-points $\mathcal{X}_1, \ldots, \mathcal{X}_4$ are weighted equally and the UT (8)–(10) does not reflect the σ-point order, the UT with the rotated σ-points (30)–(32) provides the same results for

$$\mathbf{C}(\theta) = \mathbf{C}(\theta + 90) = \mathbf{C}(\theta + 180) = \mathbf{C}(\theta + 270). \tag{86}$$

assuming θ in [deg]. That means the optimisation of the criterion (81) does not bring any benefit if performed on the interval $\theta \in \langle 0, 360)$ [deg] w.r.t. optimisation on $\theta \in \langle 0, 90)$ [deg].

1.2. Three dimensional case

Considering $n_x = 3$, $\hat{\mathbf{x}} = \mathbf{0}_{3\times 1}$, $\mathbf{P}^{\mathbf{xx}} = \mathbf{I}_3 = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$, and $(n_x + \kappa) = 1$, the σ-point set computed according to (4)–(6) is

$$\begin{aligned}&\mathcal{X}_0 = \mathbf{0}_{3\times 1},\ \mathcal{X}_1 = \mathbf{e}_1,\ \mathcal{X}_2 = \mathbf{e}_2,\ \mathcal{X}_3 = \mathbf{e}_3,\ \mathcal{X}_4 = -\mathbf{e}_1, \\ &\mathcal{X}_5 = -\mathbf{e}_2,\ \mathcal{X}_6 = -\mathbf{e}_3.\end{aligned} \tag{87}$$

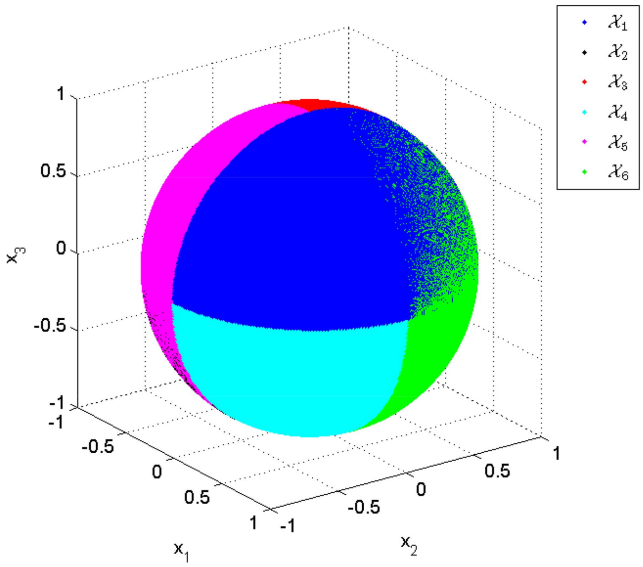


Fig. 10. Illustration of the σ-point set rotation in three dimensional case.

The number of available rotation angles (47) is $n_\theta = 3$, therefore $\boldsymbol{\theta}^{\text{grid}} = \theta_1^{\text{grid}} \times \theta_2^{\text{grid}} \times \theta_3^{\text{grid}}$, and the respective covariance matrix $\mathbf{C}$ is computed according to

$$\begin{aligned}\mathbf{C}(\boldsymbol{\theta}) &= \mathbf{C}_3(\theta_3)\mathbf{C}_2(\theta_2)\mathbf{C}_1(\theta_1) \\ &= \begin{bmatrix} 1 & 0 & 1 \\ 0 & \cos(\theta_3) & -\sin(\theta_3) \\ 0 & \sin(\theta_3) & \cos(\theta_3) \end{bmatrix} \begin{bmatrix} \cos(\theta_2) & 0 & -\sin(\theta_2) \\ 0 & 1 & 0 \\ \sin(\theta_2) & 0 & \cos(\theta_2) \end{bmatrix} \\ &\quad \times \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}.\end{aligned} \tag{88}$$

Similarly to the two dimensional case, the σ-point set rotated through the angle(s) greater than 90° is analysed and it is shown that such rotation results again in virtually the same σ-point set rotated just within the range from 0° to 90°. This is illustrated by the rotation $\theta_{1,h} \in \langle 90, 180)$ [deg] as

$$\mathcal{X}_0^r = \mathbf{C}_{3,2}\mathbf{C}_1(\theta_{1,h})\mathcal{X}_0 = \mathbf{C}_{3,2}\mathbf{C}_1(\theta_{1,l})\mathcal{X}_0 = \mathcal{X}_0, \tag{89}$$

$$\begin{aligned}\mathcal{X}_i^r &= \mathbf{C}_{3,2}\mathbf{C}_1(\theta_{1,h})\mathcal{X}_i = \mathbf{C}_{3,2}\mathbf{C}_1(\theta_{1,h} - 90)\mathbf{C}_1(90)\mathcal{X}_i \\ &= \mathbf{C}_{3,2}\mathbf{C}(\theta_{1,h} - 90)\mathcal{X}_j = \mathbf{C}_{3,2}\mathbf{C}(\theta_{1,l})\mathcal{X}_j,\end{aligned} \tag{90}$$

where $\theta_{1,l} \in \langle 0, 90)$ [deg], $\mathbf{C}_{3,2} = \mathbf{C}_3(\cdot)\mathbf{C}_2(\cdot)$,

$$\mathbf{C}_1(90) = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{91}$$

$i,j = 1,2,\ldots,6$, and

$$\mathbf{C}_1(90)\mathcal{X}_1 = [0,1,0]^{\mathrm{T}} = \mathcal{X}_2, \tag{92}$$

$$\mathbf{C}_1(90)\mathcal{X}_2 = [-1,0,0]^{\mathrm{T}} = \mathcal{X}_4, \tag{93}$$

$$\mathbf{C}_1(90)\mathcal{X}_3 = [0,1,0]^{\mathrm{T}} = \mathcal{X}_3, \tag{94}$$

$$\mathbf{C}_1(90)\mathcal{X}_4 = [0,-1,0]^{\mathrm{T}} = \mathcal{X}_5, \tag{95}$$

$$\mathbf{C}_1(90)\mathcal{X}_5 = [1,0,0]^{\mathrm{T}} = \mathcal{X}_1, \tag{96}$$

$$\mathbf{C}_1(90)\mathcal{X}_6 = [0,0,-1]^{\mathrm{T}} = \mathcal{X}_6. \tag{97}$$

The σ-points $\mathcal{X}_1,\ldots,\mathcal{X}_6$ are again weighted equally and the UT (8)–(10) does not reflect the σ-point order, therefore, the optimisation for an interval $\theta_i \in \langle 0,\theta_{i,max}\rangle$ [deg] with $\theta_{i,max} > 90^\circ$ cannot improve the results.

For illustration, the set of σ-points (87) rotated for the Cartesian product $\boldsymbol{\theta}^{\mathrm{grid}}$ with $\theta_i^{\mathrm{grid}} = \{0,1,\ldots,89\}$, $i = 1,2,3$, is shown in Fig. 10.

REFERENCES

[1] B. D. O. Anderson and J. B. Moore
Optimal Filtering.
New Jersey: Prentice Hall Ins.: Englewood Cliffs, 1979.

[2] I. Arasaratnam and S. Haykin
Cubature Kalman filters.
IEEE Transactions on Automatic Control, vol. 54, no. 6 (2009), 1254–1269.

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

[4] L. Chang, B. Hu, and F. Qin
Unscented type Kalman filter: Limitation and combination.
IET Signal Processing, vol. 7, no. 3 (2013), 167–176.

[5] A. Doucet, N. De Freitas, and N. Gordon
Sequential Monte Carlo Methods in Practice.
Springer, 2001.

[6] J. Duník, O. Straka, and M. Šimandl
On sigma-point set rotation in derivative-free filters.
In *Proceedings of the 17th International Conference on Information Fusion*, Salamanca, Spain, July 2014, 1–8.

[7] J. Duník, O. Straka, and M. Šimandl
Sigma-point set rotation in unscented Kalman filter: Analysis and adaptation.
In *Proceedings of the 19th IFAC World Congress*, Cape Town, South Africa, September 2014, 5951–5956.

[8] J. Duník, O. Straka, M. Šimandl, and E. Blasch
Random-point-based filters: Analysis and comparison in target tracking.
IEEE Transactions on Aerospace and Electronic Systems, vol. 51, no. 2 (2015), 1403–1421.

[9] J. Duník, O. Straka, and M. Šimandl
Stochastic integration filter.
IEEE Transactions on Automatic Control, vol. 58, no. 6 (2013), 1561–1566.

[10] J. Duník, M. Šimandl, and O. Straka
Adaptive choice of scaling parameter in derivative-free local filters.
In *Proceedings of the 2010 International Conference on Information Fusion*, Edinburgh, Great Britain, 2010, 1–8.

[11] J. Duník, M. Šimandl, and O. Straka
Unscented Kalman filter: Aspects and adaptive setting of scaling parameter.
IEEE Transactions on Automatic Control, vol. 57, no. 9 (2012), 2411–2416.

[12] P. D. Groves
Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems (2nd Edition).
Artech House, 2013.

[13] K. Ito and K. Xiong
Gaussian filters for nonlinear filtering problems.
IEEE Transactions on Automatic Control, vol. 45, no. 5 (2000), 910–927.

[14] B. Jia, M. Xin, and Y. Cheng
Sparse Gauss-Hermite quadrature filter with application to spacecraft attitude estimation.
Journal of Guidance, Control, and Dynamics, vol. 34, no. 2 (2011), 367–379.

[15] B. Jia, M. Xin, and Y. Cheng
High-degree cubature Kalman filter.
Automatica, vol. 49, no. 2 (2013), 510–518.

[16] S. J. Julier and J. K. Uhlmann
Unscented filtering and nonlinear estimation.
IEEE Review, vol. 92, no. 3 (2004), 401–421.

[17] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte
A new method for the nonlinear transformation of means and covariances in filters and estimators.
IEEE Transactions on Automatic Control, vol. 45, no. 3 (2000), 477–482.

[18] T. Kollo and D. Von Rosen
Advanced Multivariate Statistics with Matrices, volume 579 of Mathematics and Its Applications.
Springer, 2005.

[19] J. H. Kotecha and P. M. Djurić
Gaussian sum particle filtering.
IEEE Transactions on Signal Processing, vol. 51, no. 10 (2003), 2602–2612.

[20] X. R. Li, Z. Zhao
Measuring estimator's credibility: Noncredibility index.
In *Proceedings of the 2006 International Conference on Information Fusion*, Florence, Italy, 2006.

[21] F. L. Lewis
Optimal Estimation.
New York: John Wiley & Sons, 1986.

[22] C. D. Meyer
Matrix Analysis and Applied Linear Algebra.
SIAM, 2000.

[23] M. Nørgaard, N. K. Poulsen, and O. Ravn
New developments in state estimation for nonlinear systems.
Automatica, vol. 36, no. 11 (2000), 1627–1638.

[24] M. Rhudy, Y. Gu, J. Gross, and M. R. Napolitano
Evaluation of matrix square root operations for UKF within a UAV GPS/INS sensor fusion application.
International Journal of Navigation and Observation, vol. 2011 (2011).

[25] B. Ristic, S. Arulampalam, and N. Gordon
Beyond the Kalman Filter: Particle Filters for Tracking Applications.
Artech House, 2004.

[26] A. Sakai and Y. Kuroda
Discriminatively trained unscented Kalman filter for mobile robot localization.
Journal of Advanced Research in Mechanical Engineering, vol. 1, no. 3 (2010), 153–161.

[27] J. Sarmavuori and S. Särkkä
Fourier-Hermite Kalman filter.
IEEE Transactions on Automatic Control, vol. 57, no. 6 (2012), 1511–1515.

[28] M. Šimandl and J. Duník
Derivative-free estimation methods: New results and performance analysis.
Automatica, vol. 45, no. 7 (2009), 1749–1757.

[29] M. Šimandl, J. Královec, and T. Söderström
Advanced point-mass method for nonlinear state estimation.
Automatica, vol. 42, no. 7 (2006), 1133–1145.

[30] H. W. Sorenson and D. L. Alspach
Nonlinear bayesian estimation using Gaussian sum approximations.
IEEE Transactions of Automatic Control, no. 4 (1972), 439–448.

[31] J. Steinbring and U. Hanebeck
S2KF: The smart sampling Kalman filter.
In *Proceedings of the 16th International Conference on Inference Fusion*, Istanbul, Turkey, July 2013, 2089–2096.

[32] O. Straka, J. Duník, and M. Šimandl
Scaling parameter in unscented transform: Analysis and specification.
In *Proceedings of the 2012 American Control Conference*, 2012, 5550–5555.

[33] O. Straka, J. Duník, and M. Šimandl
Unscented Kalman filter with advanced adaptation of scaling parameter.
Automatica, vol. 5, no. 10 (2014), 2657–2664.

[34] O. Straka, J. Duník, M. Šimandl, and J. Havlík
Aspects and comparison of matrix decompositions in unscented Kalman filter.
In *Proceedings of the 2013 American Control Conference*, Washington, DC, USA, 2013, 3075–3080.

[35] O. Straka, J. Duník, and M. Šimandl
Scaling parameter in unscented transformation: Analysis and specification.
In *Proceedings of the 2012 American Control Conference*, Montreal, Canada, June 2012, 5550–5555.

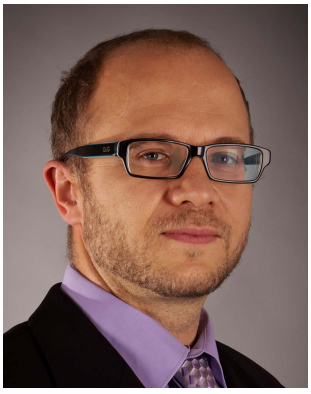
[36] R. Turner and C. E. Rasmussen
Model based learning of sigma points in unscented Kalman filter.
Neurocomputing, vol. 80 (2012), 47–53.

[37] X. Wang, Y. Liang, Q. Pan, and F. Yang
A Gaussian approximation recursive filter for nonlinear systems with correlated noises.
Automatica, vol. 48, no. 9 (2012), 2290–2297.

Jindřich Duník is a scientist at the Department of Cybernetics, Faculty of Applied Sciences, University of West Bohemia (UWB), Czech Republic. He received Ing. (M.Sc.) and Ph.D. degrees in Automatic Control in 2003 and 2008, respectively, both from the UWB. Until 2010, he was with the Department of Cybernetics focusing on the state estimation methods. Then, in the period 2010–2013, he was with Honeywell International, Aerospace Advanced Technology Europe (ATE), working in the areas of inertial and satellite based navigation systems, integrity monitoring methods, and advanced nonlinear estimation techniques. From 2013, he is with the European Centre of Excellence NTIS and Department of Cybernetics, UWB, focusing on the state estimation and system identification. From 2013, he is with Honeywell International, ATE, for part-time.

He is the author or co-author of more than 50 technical papers and patent applications devoted to the nonlinear filtering, system identification, and navigation information integrity monitoring. The papers were published at prestigious conferences (IFAC, IEEE, ISIF, ION) and journals (e.g., *Automatica*, *IEEE Transactions on Automatic Control*, *IEEE Transactions on Aerospace and Electronic Systems*). He is a guest area editor for the *Journal of Advances in Information Fusion* for the special issue on Nonlinear Derivative-Free Filters: Theory and Application and IPC Co-chair of the 12th European Workshop on Advanced Control and Diagnosis. He has also participated in a number of projects of fundamental and applied research (Czech Science Foundation, EU SESAR, ESA).

Jindřich received several Honeywell Bravo Awards and the Technology Achievement Award in 2016 for aircraft navigation system design and Werner von Siemens Excellence Award in 2014 for the basic research New Approaches and Methods of Nonlinear State Estimation and Optimal Decision Making under Uncertainty.



Ondřej Straka is head of Research Group Identification and Decision-Making in the European Centre of Excellence NTIS—New Technologies for the Information Society, where he focuses on nonlinear state estimation, stochastic systems, and system identification.

Dr. Straka focuses his research interests on both local and global nonlinear state estimation methods, performance evaluation and fault detection in navigation systems. He has published over fifty journal and conference papers in journals such as *Automatica*, *IEEE Transactions on Automatic Control*, *IEEE Transactions on Aerospace and Electronic Systems*, *Signal Processing* and at international conferences such as American Control Conference, World Congresses and Symposia of the International Federation of Automatic Control, IFAC, and FUSION Conferences. He has participated in a number of projects of fundamental research and in several project of applied research (e.g., GNSS-based safe train localization, attitude and heading reference system). He was involved in development of several software frameworks for nonlinear state estimation.

Dr. Straka received his Master's degree in Cybernetics and Control Engineering (in 1998) and his Ph.D. degree in Cybernetics (in 2004) both from University of West Bohemia. Since 2015 he has been associate professor at the Department of Cybernetics, University of West Bohemia.

Miroslav Šimandl was born in Ledce, Czechoslovakia, in 1954. He received the M.Sc. (Dipl.Ing.) degree in control engineering in 1978 and the Ph.D. (C.Sc.) degree in technical cybernetics in 1984, both from the Institute of Technology in Pilsen, Czechoslovakia.

In the period 1978–1992, he held various research and teaching positions at the Institute of Technology in Pilsen. In the period 1993–2001, he was associate professor, and since 2002 he has held the position of professor at the Department of Cybernetics, Faculty of Applied Sciences, University of West Bohemia, Pilsen, Czech Republic. Since 2011, he has held the position of vice-rector for research and development at the University of West Bohemia. Currently, he is also with the European Centre of Excellence—New Technologies for the Information Society (NTIS). Within IFAC, he served as the co-chair of the International Programme Committee of the 16th International Federation of Automatic Control (IFAC) World Congress in Prague, 2005, and he has served as a member of two technical committees of IFAC.

Prof. Šimandl is the author or co-author of many journal and conference technical papers. His main research interests are in the fields of nonlinear filtering, active fault detection, nonlinear identification, and adaptive dual control. In 2014 he received the Werner von Siemens Excellence Award for the basic research New Approaches and Methods of Nonlinear State Estimation and Optimal Decision Making under Uncertainty.

Erik P. Blasch is a principal scientist at the the United States Air Force Research Laboratory (AFRL) in the Information Directorate at Rome, NY, USA. From 2009–2012, he was an exchange scientist to Defence Research and Development Canada (DRDC) at Valcartier, Quebec. From 2000–2009, Dr. Blasch was the Information Fusion Evaluation Tech Lead for the AFRL Sensors Directorate—COMprehensive Performance Assessment of Sensor Exploitation (COMPASE) Center supporting design evaluations in Dayton, OH. Dr. Blasch has been an Adjunct Electrical Engineering Professor at Wright State University, the Air Force Institute of Technology, and the University of Dayton teaching signal processing, target tracking, and information fusion. He has also held various assignements as a reserve military officer supporting space situational awareness, photonics design, and dynamic data driven application systems.

Dr. Blasch was a founding member of the International Society of Information Fusion (ISIF) (www.isif.org) in 1998, held various leadership roles for the Fusion conferences, a Board of Governors (BoG) member (1999–2010), and the 2007 ISIF President. For the Institute of Electrical and Electronics Engineers (IEEE), he has served as a member of the Aerospace and Electronics Systems Society (AESS) BoG (2011–2016), the AESS International Chapters Chair (2012–2016), representative to AIAA (2013–2016), and an AESS Distinguished lecturer (2013–2016). He has focused on information fusion, target tracking, pattern recognition, and robitcs research compiling 600+ scientific papers and book chapters. He holds 8 patents, presented over 30 tutorials, and is an associate editor of three academic journals. His books include High-Level Information Fusion Management and Systems Design (Artech House, 2012) and Context Enhanced Information Fusion (Springer, 2016).

Dr. Blasch received his B.S. in Mechanical Engineering from the Massachusetts Institute of Technology in 1992 and Master's Degrees in Mechanical ('94), Health Science ('95), and Industrial Engineering (Human Factors) ('95) from Georgia Tech and attended the University of Wisconsin for a MD/PhD in Neurosciences/Mech. Eng until being called to military service in 1996 to the United States Air Force. He completed an MBA ('98), MSEE ('98), MS Econ('99), and a PhD ('99) in Electrical Engineering from Wright State University and is a graduate of Air War College ('08). He is the recipeint of the IEEE Bio-Engineering Award (Russ-2008), IEEE AESS Magazine Best Paper Award (Mimno-2012), and Military Sensing Symposium Ledaership in Data Fusion Award (Mignogna-2014). He is a Fellow of SPIE, Associate Fellow of AIAA, and a senior member of IEEE.

JAIF

Journal of Advances in Information Fusion

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

Volumes 1–10 Index

A

Adurthi, N., *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.
Alford, M., *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.
Alsun, M., Lecornu, L., Solaiman, B., Possibilistic Medical Knowledge Representation Model, *JAIF*, **7**, 2 (December 2012), 101–113.
An, W., Singh, S., Pattipati, K. R., Kleinman, D. L., and Gokhale, S. S., Dynamic Scheduling of Multiple Hidden Markov Model-Based Sensors, *JAIF*, **3**, 1 (July 2008), 33–49.
Anderson, S. L., *see* Stone, L. D., *JAIF*, **10**, 1 (June 2015), 3–12.
Andler, S. F., *see* Karlsson, A., *JAIF*, **6**, 2 (December 2011), 150–166.
Aravinthan, A., *see* Habtemariam, B. K., *JAIF*, **7**, 2 (December 2012), 114–130.
Areta, J., Bar-Shalom, Y., and Pattipati, K. R., T2T and M2T Association with Combined Hypotheses, *JAIF*, **4**, 1 (July 2009), 40–51.
Areta, J., Bar-Shalom, Y., and Rothrock, R., Misassociation Probability in M2TA and T2TA, *JAIF*, **2**, 2 (Dec. 2007), 113–127.
Areta, J., Bar-Shalom, Y., Levedahl, M., and Pattipati, K. R., Hierarchical Track Association and Fusion for a Networked Surveillance System, *JAIF*, **1**, 2 (Dec. 2006), 144–157.
Arnborg, S., *see* Brynielsson, J., *JAIF*, **1**, 2 (Dec. 2006), 108–121.
Arnborg, S., Robust Bayesianism: Relation to Evidence Theory *JAIF*, **1**, 1 (July 2006), 75–90.
Aughenbaugh, J. M. and La Cour, B. R., Measurement-Guided Likelihood Sampling for Grid-Based Bayesian Tracking *JAIF*, **5**, 2 (Dec. 2010), 108–127.
Avasarala, V., Mullen, T., and Hall, D., A Market-based Approach to Sensor Management, *JAIF*, **4**, 1 (July 2009), 52–71.

B

Bab-Hadiashar, A., *see* Hoseinnezhad, R., *JAIF*, **1**, 1 (July 2006), 52–62.
Balasingam, B., *see* Choi, S., *JAIF*, **8**, 2 (December 2013), 143–155.
Bar-Shalom, Y., *see* Areta, J., *JAIF*, **1**, 2 (Dec. 2006), 144–157.
Bar-Shalom, Y., *see* Areta, J., *JAIF*, **2**, 2 (Dec. 2007), 113–127.
Bar-Shalom, Y., *see* Areta, J., *JAIF*, **4**, 1 (July 2009), 40–51.
Bar-Shalom, Y., *see* Belfadel, D., *JAIF*, **9**, 2 (December 2014), 59–74.
Bar-Shalom, Y., *see* Belfadel, D., *JAIF*, **10**, 2 (December 2015), 101–112.
Bar-Shalom, Y. and Chen, H., Covariance Reconstruction for Track Fusion with Legacy Track Sources *JAIF*, **3**, 2 (Dec. 2008), 107–117.
Bar-Shalom, Y. and Chen, H., Multisensor Track-to-Track Association for Tracks with Dependent Errors *JAIF*, **1**, 1 (July 2006), 3–14.
Bar-Shalom, Y. and Chen, H., Track-to-Track Association Using Attributes *JAIF*, **2**, 1 (July 2007), 49–59.
Bar-Shalom, Y., *see* Crouse, D. F., *JAIF*, **8**, 1 (July 2013), 73–89.
Bar-Shalom, Y., *see* Dou, W., *JAIF*, **10**, 2 (December 2015), 163–182.
Bar-Shalom, Y., *see* Osborne, R. W., III, *JAIF*, **7**, 1 (June 2012), 3–15.
Bar-Shalom, Y., *see* Osborne, R. W., III, *JAIF*, **9**, 2 (December 2014), 75–89.
Bar-Shalom, Y., *see* Osborne, III, R. W., *JAIF*, **10**, 2 (December 2015), 199–210.
Bar-Shalom, Y., *see* Ravindra, V. C., *JAIF*, **5**, 2 (Dec. 2010), 88–107.
Bar-Shalom, Y., *see* Rodningsby, A., *JAIF*, **4**, 2 (Dec. 2009), 117–145.
Bar-Shalom, Y., *see* Romeo, K., *JAIF*, **10**, 2 (December 2015), 113–124.
Bar-Shalom, Y., *see* Tharmarasa, R., *JAIF*, **7**, 1 (June 2012), 46–60.
Bar-Shalom, Y., *see* Tian, X., *JAIF*, **4**, 2 (Dec. 2009), 146–164.
Bar-Shalom, Y., *see* Tian, X., *JAIF*, **5**, 1 (July 2010), 3–17.
Bar-Shalom, Y., *see* Tian, X., *JAIF*, **5**, 2 (Dec. 2010), 128–138.
Bar-Shalom, Y., *see* Yuan, T., *JAIF*, **6**, 2 (December 2011), 131–149.
Bar-Shalom, Y., *see* Zhang, S., *JAIF*, **6**, 1 (June 2011), 3–23.
Bar-Shalom, Y., *see* Zhang, S., *JAIF*, **9**, 1 (July 2014), 38–46.
Baum, M., *see* Faion, F., *JAIF*, **10**, 1 (June 2015), 13–30.
Belaroussi, R., Prevost, L., and Milgram, M., Algorithms Fusion for Face Localization, *JAIF*, **1**, 1 (July 2006), 35–51.
Belfadel, D., Osborne, R. W., III, Bar-Shalom, Y., Bias Estimation and Observability for Optical Sensor Measurements with Targets of Opportunity, *JAIF*, **9**, 2 (December 2014), 59–74.
Belfadel, D., Osborne, R. W., III, Bar-Shalom, Y., Bias Estimation for Moving Optical Sensor Measurements with Targets of Opportunity, *JAIF*, **10**, 2 (December 2015), 101–112.
Benaskeur, A. R., Rhéaume, F., and Paradis, S., Target Engageability Improvement through Adaptive Tracking, *JAIF*, **2**, 2 (Dec. 2007), 99–112.
Biermann, J., Hörling, P., Snidaro, L., Experiences and Challenges in Automated Support for Intelligence in Asymmetric Operations, *JAIF*, **8**, 2 (December 2013), 101–118.
Blair, W. D. and Miceli, P. A., Performance Prediction of Multisensor Tracking Systems for Single Maneuvering Targets *JAIF*, **7**, 1 (June 2012), 28–45.
Blasch, E., *see* Chen, G., *JAIF*, **2**, 1 (July 2007), 35–48.
Blasch, E., Kadar, I., Salerno, J., Kokar, M. M., Das, S., Powell, G. M., Corkill, D. D., and Ruspini, E. H., Issues and Challenges in Situation Assessment (Level 2 Fusion), *JAIF*, **1**, 2 (Dec. 2006), 122–143.
Blasch, E., *see* Kahler, B., *JAIF*, **6**, 2 (December 2011), 101–118.
Blasch, E., *see* Yang, C., *JAIF*, **3**, 1 (July 2008), 14–32.
Blasch, E., *see* Zheng, Y., *JAIF*, **9**, 2 (December 2014), 124–135.
Bloem, E. A., *see* Blom, H. A. P., *JAIF*, **1**, 1 (July 2006), 15–34.
Blom, H. A. P. and Bloem, E. A., Joint Particle Filtering of Multiple Maneuvering Targets From Unassociated Measurements *JAIF*, **1**, 1 (July 2006), 15–34.
Bossé, É., *see* Valin, P., *JAIF*, **5**, 1 (July 2010), 32–40.
Brynielsson, J. and Arnborg, S., An Information Fusion Game Component *JAIF*, **1**, 2 (Dec. 2006), 108–121.
Bubalo, A., *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.
Bursik, M. I., *see* Rogova, G. L., *JAIF*, **3**, 2 (Dec. 2008), 118–128.

C

Carthel, C., *see* Coraluppi, S., *JAIF*, **5**, 1 (July 2010), 18–31.
Carthel, C., *see* Coraluppi, S., *JAIF*, **6**, 1 (June 2011), 57–67.
Carthel, C., *see* Coraluppi, S., *JAIF*, **7**, 2 (December 2012), 153–164.
Chakravorty, R. and Challa, S., Augmented State Integrated Probabilistic Data Association Smoothing for Automatic Track Initiation in Clutter *JAIF*, **1**, 1 (July 2006), 63–74.
Carvalho, R. N. and Chang, K., A Fusion Analysis and Evaluation Tool for Multi-Sensor Classification Systems *JAIF*, **7**, 2 (December 2012), 141–152.
Challa, S., *see* Chakravorty, R., *JAIF*, **1**, 1 (July 2006), 63–74.
Chang, K., *see* Carvalho, R. N., *JAIF*, **7**, 2 (December 2012), 141–152.
Chang, K. C. and Hill, J. P., Level I and Level II Target Valuations for Sensor Management *JAIF*, **1**, 2 (Dec. 2006), 95–107.
Chen, G., Shen, D., Kwan, C., Cruz, J. B., Jr., Kruger, M., and Blasch, E., Game Theoretic Approach to Threat Prediction and Situation Awareness, *JAIF*, **2**, 1 (July 2007), 35–48.
Chen, H., *see* Bar-Shalom, Y., *JAIF*, **1**, 1 (July 2006), 3–14.
Chen, H., *see* Bar-Shalom, Y., *JAIF*, **2**, 1 (July 2007), 49–59.
Chen, H., *see* Bar-Shalom, Y., *JAIF*, **3**, 2 (Dec. 2008), 107–117.
Choi, S., Balasingam, B., Willett, P., Testing Under Communication Constraints, *JAIF*, **8**, 2 (December 2013), 143–155.
Choi, S., Willett, P., Zhou, S., The PMHT for Passive Radar in a DAB/DVB Network, *JAIF*, **9**, 1 (July 2014), 27–37.
Coraluppi, S. and Carthel, C., Multi-Stage Multiple-Hypothesis Tracking *JAIF*, **6**, 1 (June 2011), 57–67.
Coraluppi, S. and Carthel, C., Modified Scoring in Multiple-Hypothesis Tracking *JAIF*, **7**, 2 (December 2012), 153–164.
Coraluppi, S., *see* Erdinc, O., *JAIF*, **2**, 1 (July 2007), 22–34.
Coraluppi, S., Guerriero, M., Willett, P., and Carthel, C., Fuse-before-Track in Large Sensor Networks, *JAIF*, **5**, 1 (July 2010), 18–31.
Corkill, D. D., *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.
Cornacchia, M., *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.
Crassidis, J. L., *see* George, J., *JAIF*, **6**, 1 (June 2011), 39–56.
Crouse, D. F., Guerriero, M., and Willett, P., A Critical Look at the PMHT, *JAIF*, **4**, 2 (Dec. 2009), 93–116.
Crouse, D. F., Osborne, R. W., III, Pattipati, K., Willett, P., Bar-Shalom, Y., Efficient 2D Sensor Location Estimation using Targets of Opportunity, *JAIF*, **8**, 1 (July 2013), 73–89.
Crouse, D. F., Simulating Aerial Targets in 3D Accounting for the Earth's Curvature *JAIF*, **10**, 1 (June 2015), 31–57.
Crouse, D. F., Zheng, L., Willett, P., Corrections to "A Critical Look at the PMHT", *JAIF*, **7**, 1 (June 2012), 97–98.
Cruz, J. B., Jr., *see* Chen, G., *JAIF*, **2**, 1 (July 2007), 35–48.

D

Damarla, T., *see* Ravindra, V. C., *JAIF*, **5**, 2 (Dec. 2010), 88–107.
Das, S., *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.
Davey, S. J., Detecting a Small Boat using Histogram PMHT *JAIF*, **6**, 2 (December 2011), 167–186.
Degen, C., Govaers, F., Koch, W., Tracking Targets with Multiple Measurements per Scan Using the Generalized PHD Filter, *JAIF*, **10**, 2 (December 2015), 125–142.
Dezert, J., *see* Martin, A., *JAIF*, **3**, 2 (Dec. 2008), 67–89.
Djiknavorian, P., *see* Valin, P., *JAIF*, **5**, 1 (July 2010), 32–40.
Dou, W., Bar-Shalom, Y., Willett, P., Bistatic Measurement Fusion from Multistatic Configurations for Air Collision Warning, *JAIF*, **10**, 2 (December 2015), 163–182.

E

Eilbert, J., *see* Schrag, R. C., *JAIF*, **2**, 2 (Dec. 2007), 77–98.
Erdinc, O., Willett, P., and Coraluppi, S., Multistatic Sensor Placement: A Tracking Approach, *JAIF*, **2**, 1 (July 2007), 22–34.

F

Faion, F., Zea, A., Baum, M., Hanebeck, U. D., Symmetries in Bayesian Extended Object Tracking, *JAIF*, **10**, 1 (June 2015), 13–30.
Falkman, G., *see* Johansson, F., *JAIF*, **6**, 2 (December 2011), 187–199.
Ferry, J. P., Exact Association Probability for Data with Bias and Features *JAIF*, **5**, 1 (July 2010), 41–67.
Foo, P. H. and Ng, G. W., High-level Information Fusion: An Overview *JAIF*, **8**, 1 (July 2013), 33–72.
Foo, P. H., Ng, G. W., Ng, K. H., and Yang, R., Application of Intent Inference for Air Defense and Conformance Monitoring, *JAIF*, **4**, 1 (July 2009), 3–26.
Fosbury, A. M., *see* George, J., *JAIF*, **6**, 1 (June 2011), 39–56.

G

George, J., Crassidis, J. L., Singh, T., Fosbury, A. M., Anomaly Detection using Context-Aided Target Tracking, *JAIF*, **6**, 1 (June 2011), 39–56.
George, J. and Kaplan, L. M., Shooter Localization using a Wireless Sensor Network of Soldier-Worn Gunfire Detection Systems *JAIF*, **8**, 1 (July 2013), 15–32.
George, J., *see* Osborne, R. W., III, *JAIF*, **9**, 2 (December 2014), 75–89.
Georgescu, R., Willett, P., Marano, S., Matta, V., Predetection Fusion in Large Sensor Networks with Unknown Target Locations, *JAIF*, **7**, 1 (June 2012), 61–77.
Gilitschenski, I., *see* Kurz, G., *JAIF*, **9**, 2 (December 2014), 90–105.
Glattetre, J., *see* Rodningsby, A., *JAIF*, **4**, 2 (Dec. 2009), 117–145.
Goger, P., *see* Schrag, R. C., *JAIF*, **2**, 2 (Dec. 2007), 77–98.
Goker, A., *see* Kaliciak, L., *JAIF*, **10**, 2 (December 2015), 183–198.
Gokhale, S. S., *see* An, W., *JAIF*, **3**, 1 (July 2008), 33–49.
Govaers, F., *see* Degen, C., *JAIF*, **10**, 2 (December 2015), 125–142.
Guerriero, M., *see* Coraluppi, S., *JAIF*, **5**, 1 (July 2010), 18–31.
Gregory Watson, *see* Zhang, S., *JAIF*, **6**, 1 (June 2011), 3–23.
Guerriero, M., *see* Crouse, D. F., *JAIF*, **4**, 2 (Dec. 2009), 93–116.
Guitouni, A., *see* Jabeur, K., *JAIF*, **4**, 2 (Dec. 2009), 75–92.

H

Habtemariam, B. K., Aravinthan, A., Tharmarasa, R., Punithakumar, K., Lang, T., Kirubarajan, T., Distributed Tracking with a PHD Filter using Efficient Measurement Encoding, *JAIF*, **7**, 2 (December 2012), 114–130.
Hall, D., *see* Avasarala, V., *JAIF*, **4**, 1 (July 2009), 52–71.
Hallingstad, O., *see* Rodningsby, A., *JAIF*, **4**, 2 (Dec. 2009), 117–145.
Hamp, Q. and Reindl, L., Association Performance Enhancement Through Classification *JAIF*, **7**, 2 (December 2012), 131–140.
Han, X., Yu, F., Levchuck, G., Pattipati, K., Tu, F., A Probabilistic Computational Model for Identifying Organizational Structures from Uncertain Activity Data, *JAIF*, **7**, 1 (June 2012), 78–96.
Hanebeck, U. D., *see* Faion, F., *JAIF*, **10**, 1 (June 2015), 13–30.
Hanebeck, U. D., *see* Kurz, G., *JAIF*, **9**, 1 (July 2014), 13–26.
Hanebeck, U. D., *see* Kurz, G., *JAIF*, **9**, 2 (December 2014), 90–105.
Hanebeck, U. D., *see* Steinbring, J., *JAIF*, **9**, 2 (December 2014), 106–123.
Hanson-Hedgecock, S., *see* Rogova, G. L., *JAIF*, **3**, 2 (Dec. 2008), 118–128.
Hill, J. P., *see* Chang, K. C., *JAIF*, **1**, 2 (Dec. 2006), 95–107.
Holender, M., *see* Papageorgiou, D. J., *JAIF*, **6**, 2 (December 2011), 77–100.
Holender, M., *see* Sudit, M., *JAIF*, **2**, 1 (July 2007), 3–21.
Hörling, P., *see* Biermann, J., *JAIF*, **8**, 2 (December 2013), 101–118.
Hoseinnezhad, R. and Bab-Hadiashar, A., Fusion of Redundant Information in Brake-By-Wire Systems Using a Fuzzy Voter *JAIF*, **1**, 1 (July 2006), 52–62.
Howell, C. and Moyer, S., Establishment of Human Performance Baseline for Image Fusion Algorithms in the LWIR and SWIR Spectra *JAIF*, **8**, 2 (December 2013), 133–142.

J

Jabeur, K. and Guitouni, A., A Generalized Framework for Multi-Criteria Classifiers with Automated Learning: Application on FLIR Ship Imagery *JAIF*, **4**, 2 (Dec. 2009), 75–92.
Jauffret, C., Pillon, D., Pignol, A-C., Leg-by-leg Bearings-Only Target Motion Analysis Without Observer Maneuver, *JAIF*, **6**, 1 (June 2011), 24–38.
Jentschel, H-J., *see* Junghans, M., *JAIF*, **3**, 1 (July 2008), 50–62.
Jilkov, V. P. and Li, X. R., On Fusion of Multiple Objectives for UAV Search & Track Path Optimization *JAIF*, **4**, 1 (July 2009), 27–39.
Jilkov, V. P. and Wu, J., Efficient GPU-Accelerated Implementation of Particle and Particle Flow Filters for Target Tracking *JAIF*, **10**, 1 (June 2015), 73–88.
Johansson, F. and Falkman, G., Real-time Allocation of Firing Units To Hostile Targets *JAIF*, **6**, 2 (December 2011), 187–199.
Johansson, R., *see* Karlsson, A., *JAIF*, **6**, 2 (December 2011), 150–166.
Julier, S., *see* Kurz, G., *JAIF*, **9**, 2 (December 2014), 90–105.
Jones, E., *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.
Junghans, M. and Jentschel, H-J., Correction of Selection Bias in Traffic Data by Bayesian Network Data Fusion *JAIF*, **3**, 1 (July 2008), 50–62.

K

Kadar, I., *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.
Kahler, B. and Blasch, E., Decision-Level Fusion Performance Improvement From Enhanced HRR Radar Clutter Suppression *JAIF*, **6**, 2 (December 2011), 101–118.
Kaliciak, L., Myrhaug, H., Goker, A., Song, D., Early Fusion and Query Modification in Their Dual Late Fusion Forms, *JAIF*, **10**, 2 (December 2015), 183–198.
Kaplan, L. M., *see* George, J., *JAIF*, **8**, 1 (July 2013), 15–32.
Kaplan, L., *see* Osborne, R. W., III, *JAIF*, **9**, 2 (December 2014), 75–89.
Karlsson, A., Johansson, R., Andler, S. F., Characterization and Empirical Evaluation of Bayesian and Credal Combination Operators, *JAIF*, **6**, 2 (December 2011), 150–166.
Kirubarajan, T., *see* Habtemariam, B. K., *JAIF*, **7**, 2 (December 2012), 114–130.
Kirubarajan, T., *see* Tharmarasa, R., *JAIF*, **7**, 1 (June 2012), 46–60.
Kleinman, D. L., *see* An, W., *JAIF*, **3**, 1 (July 2008), 33–49.
Koch, W., *see* Degen, C., *JAIF*, **10**, 2 (December 2015), 125–142.
Kokar, M. M., *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.
Kroschel, K., *see* Schlosser, M. S., *JAIF*, **2**, 2 (Dec. 2007), 65–76.
Kruger, M., *see* Chen, G., *JAIF*, **2**, 1 (July 2007), 35–48.
Kurz, G., Gilitschenski, I., Julier, S., Hanebeck, U. D., Recursive Bingham Filter for Directional Estimation Involving 180 Degree Symmetry, *JAIF*, **9**, 2 (December 2014), 90–105.
Kurz, G. and Hanebeck, U. D., Dynamic Surface Reconstruction by Recursive Fusion of Depth and Position Measurements *JAIF*, **9**, 1 (July 2014), 13–26.
Kwan, C., *see* Chen, G., *JAIF*, **2**, 1 (July 2007), 35–48.

L

La Cour, B. R., *see* Aughenbaugh, J. M., *JAIF*, **5**, 2 (Dec. 2010), 108–127.
Lang, T., *see* Habtemariam, B. K., *JAIF*, **7**, 2 (December 2012), 114–130.
Lecornu, L., *see* Alsun, M., *JAIF*, **7**, 2 (December 2012), 101–113.
Levchuck, G., *see* Han, X., *JAIF*, **7**, 1 (June 2012), 78–96.
Levedahl, M., *see* Areta, J., *JAIF*, **1**, 2 (Dec. 2006), 144–157.
Li, X. R., *see* Jilkov, V. P., *JAIF*, **4**, 1 (July 2009), 27–39.

M

Marano, S., *see* Georgescu, R., *JAIF*, **7**, 1 (June 2012), 61–77.
Martin, A., Osswald, C., Dezert, J., Smarandache, F., General Combination Rules for Qualitative and Quantitative Beliefs, *JAIF*, **3**, 2 (Dec. 2008), 67–89.
Matta, V., *see* Georgescu, R., *JAIF*, **7**, 1 (June 2012), 61–77.
Miceli, P. A., *see* Blair, W. D., *JAIF*, **7**, 1 (June 2012), 28–45.
Milgram, M., *see* Belaroussi, R., *JAIF*, **1**, 1 (July 2006), 35–51.
Moyer, S., *see* Howell, C., *JAIF*, **8**, 2 (December 2013), 133–142.
Mullen, T., *see* Avasarala, V., *JAIF*, **4**, 1 (July 2009), 52–71.
Myrhaug, H., *see* Kaliciak, L., *JAIF*, **10**, 2 (December 2015), 183–198.

N

Nagi, R., *see* Sambhoos, K., *JAIF*, **3**, 2 (Dec. 2008), 90–106.
Nadarajah, N., *see* Tharmarasa, R., *JAIF*, **7**, 1 (June 2012), 46–60.
Ng, G. W., *see* Foo, P. H., *JAIF*, **4**, 1 (July 2009), 3–26.
Ng, K. H., *see* Foo, P. H., *JAIF*, **4**, 1 (July 2009), 3–26.
Ng, G. W., *see* Foo, P. H., *JAIF*, **8**, 1 (July 2013), 33–72.

O

Osborne, R. W., III, Bar-Shalom, Y., George, J., Kaplan, L., Statistical Efficiency of Target Localization from Angle and Shockwave Measurements, *JAIF*, **9**, 2 (December 2014), 75–89.
Osborne, R. W., III, Bar-Shalom, Y., Willett, P., Fusion of Asynchronous Passive Measurements, *JAIF*, **10**, 2 (December 2015), 199–210.
Osborne, R. W., III, Bar-Shalom, Y., Willett, P., Track-to-Track Association with Augmented State, *JAIF*, **7**, 1 (June 2012), 3–15.
Osborne, R. W., III, *see* Belfadel, D., *JAIF*, **9**, 2 (December 2014), 59–74.
Osborne, R. W., III, *see* Belfadel, D., *JAIF*, **10**, 2 (December 2015), 101–112.
Osborne, R. W., III, *see* Crouse, D. F., *JAIF*, **8**, 1 (July 2013), 73–89.
Osswald, C., *see* Martin, A., *JAIF*, **3**, 2 (Dec. 2008), 67–89.

P

Panakkal, V. P. and Velmurugan, R., An improved measurement model for target tracking under measurement origin uncertainty *JAIF*, **10**, 2 (December 2015), 143–162.

Papageorgiou, D. J. and Holender, M., Track-to-Track Association and Ambiguity Management in the Presence of Sensor Bias *JAIF*, **6**, 2 (December 2011), 77–100.

Paradis, S., *see* Benaskeur, A. R., *JAIF*, **2**, 2 (Dec. 2007), 99–112.

Pattipati, K. R., *see* An, W., *JAIF*, **3**, 1 (July 2008), 33–49.

Pattipati, K. R., *see* Areta, J., *JAIF*, **1**, 2 (Dec. 2006), 144–157.

Pattipati, K. R., *see* Areta, J., *JAIF*, **4**, 1 (July 2009), 40–51.

Pattipati, K., *see* Crouse, D. F., *JAIF*, **8**, 1 (July 2013), 73–89.

Pattipati, K., *see* Han, X., *JAIF*, **7**, 1 (June 2012), 78–96.

Pattipati, K. R., *see* Tian, X., *JAIF*, **5**, 1 (July 2010), 3–17.

Peel, L., Estimating Network Parameters for Selecting Community Detection Algorithms *JAIF*, **6**, 2 (December 2011), 119–130.

Pignol, A-C., *see* Jauffret, C., *JAIF*, **6**, 1 (June 2011), 24–38.

Pillon, D., *see* Jauffret, C., *JAIF*, **6**, 1 (June 2011), 24–38.

Powell, G. M., *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.

Prevost, L., *see* Belaroussi, R., *JAIF*, **1**, 1 (July 2006), 35–51.

Punithakumar, K., *see* Habtemariam, B. K., *JAIF*, **7**, 2 (December 2012), 114–130.

R

Ravindra, V. C., Bar-Shalom, Y., and Damarla, T., Feature-Aided Tracking of Ground Vehicles using Passive Acoustic Sensor Arrays, *JAIF*, **5**, 2 (Dec. 2010), 88–107.

Rhéaume, F., *see* Benaskeur, A. R., *JAIF*, **2**, 2 (Dec. 2007), 99–112.

Reindl, L., *see* Hamp, Q., *JAIF*, **7**, 2 (December 2012), 131–140.

Rickard, T., *see* Sambhoos, K., *JAIF*, **3**, 2 (Dec. 2008), 90–106.

Rickard, T., *see* Sudit, M., *JAIF*, **2**, 1 (July 2007), 3–21.

Rodningsby, A., Bar-Shalom, Y., Hallingstad, O., and Glattetre, J., Multitarget Multisensor Tracking in the Presence of Wakes, *JAIF*, **4**, 2 (Dec. 2009), 117–145.

Rogova, G. L., Bursik, M. I., and Hanson-Hedgecock, S., Intelligent System for Interpreting the Pattern of Volcanic Eruptions, *JAIF*, **3**, 2 (Dec. 2008), 118–128.

Romeo, K., Bar-Shalom, Y., Willett, P., Fusion of Multipath Data with ML-PMHT for Very Low SNR Track Detection in an OTHR, *JAIF*, **10**, 2 (December 2015), 113–124.

Rothrock, R., *see* Areta, J., *JAIF*, **2**, 2 (Dec. 2007), 113–127.

Ruspini, E. H., *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.

S

Salerno, E., Adurthi, N., Singh, T., Singla, P., Bubalo, A., Cornacchia, M., Alford, M., Jones, E., Road Network Identification by means of the Hough Transform with Uncertainty Analysis, *JAIF*, **10**, 1 (June 2015), 58–72.

Salerno, J., *see* Blasch, E., *JAIF*, **1**, 2 (Dec. 2006), 122–143.

Sambhoos, K., Nagi, R., Sudit, M., and Rickard, T., Hierarchical Higher Level Data Fusion using Fuzzy Hamming and Hypercube Clustering, *JAIF*, **3**, 2 (Dec. 2008), 90–106.

Schlosser, M. S. and Kroschel, K., Performance Analysis of Decentralized Kalman Filters under Communication Constraints *JAIF*, **2**, 2 (Dec. 2007), 65–76.

Schneider, M. K., *see* Washburn, R. B., *JAIF*, **3**, 1 (July 2008), 3–13.

Schrag, R. C., Takikawa, M., Goger, P., and Eilbert, J., Performance Evaluation for Automated Threat Detection, *JAIF*, **2**, 2 (Dec. 2007), 77–98.

Scott, P. D., *see* Terejanu, G., *JAIF*, **5**, 2 (Dec. 2010), 73–87.

Shen, D., *see* Chen, G., *JAIF*, **2**, 1 (July 2007), 35–48.

Singh, S., *see* An, W., *JAIF*, **3**, 1 (July 2008), 33–49.

Singh, T., *see* George, J., *JAIF*, **6**, 1 (June 2011), 39–56.

Singh, T., *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.

Singh, T., *see* Terejanu, G., *JAIF*, **5**, 2 (Dec. 2010), 73–87.

Singla, P., *see* Salerno, E., *JAIF*, **10**, 1 (June 2015), 58–72.

Singla, P., *see* Terejanu, G., *JAIF*, **5**, 2 (Dec. 2010), 73–87.

Smarandache, F., *see* Martin, A., *JAIF*, **3**, 2 (Dec. 2008), 67–89.

Snidaro, L., *see* Biermann, J., *JAIF*, **8**, 2 (December 2013), 101–118.

Solaiman, B., *see* Alsun, M., *JAIF*, **7**, 2 (December 2012), 101–113.

Song, D., *see* Kaliciak, L., *JAIF*, **10**, 2 (December 2015), 183–198.

Song, X., Willett, P., Zhou, S., Bearings-Only Localization with NLOS Reflected AoAs, *JAIF*, **8**, 1 (July 2013), 3–14.

Song, X., Willett, P., Zhou, S., Posterior Cramér-Rao Bounds for Doppler Biased Distributed Tracking, *JAIF*, **7**, 1 (June 2012), 16–27.

Steinbring, J. and Hanebeck, U. D., LRKF Revisited: The Smart Sampling Kalman Filter (S^2KF) *JAIF*, **9**, 2 (December 2014), 106–123.

Stone, L. D. and Anderson, S. L., Standard Bayesian Approach to Quantized Measurements and Imprecise Likelihoods *JAIF*, **10**, 1 (June 2015), 3–12.

Stotz, A., *see* Sudit, M., *JAIF*, **2**, 1 (July 2007), 3–21.

Streit, R., A Technique for Deriving Multitarget Intensity Filters Using Ordinary Derivatives *JAIF*, **9**, 1 (July 2014), 3–12.

Streit, R., The Probability Generating Functional for Finite Point Processes, and Its Application to the Comparison of PHD and Intensity Filters *JAIF*, **8**, 2 (December 2013), 119–132.

Sudit, M., Holender, M., Stotz, A., Rickard, T., and Yager, R., INFERD and Entropy for Situational Awareness, *JAIF*, **2**, 1 (July 2007), 3–21.

Sudit, M., *see* Sambhoos, K., *JAIF*, **3**, 2 (Dec. 2008), 90–106.

T

Takikawa, M., *see* Schrag, R. C., *JAIF*, **2**, 2 (Dec. 2007), 77–98.

Terejanu, G., Singla, P., Singh, T., and Scott, P. D., A Decision-Centric Framework for Density Forecasting, *JAIF*, **5**, 2 (Dec. 2010), 73–87.

Tharmarasa, R., *see* Habtemariam, B. K., *JAIF*, **7**, 2 (December 2012), 114–130.

Tharmarasa, R., Kirubarajan, T., Nadarajah, N., Bar-Shalom, Y., Thayaparan, T., Profile-Free Launch Point Estimation for Ballistic Targets using Passive Sensors, *JAIF*, **7**, 1 (June 2012), 46–60.

Thayaparan, T., *see* Tharmarasa, R., *JAIF*, **7**, 1 (June 2012), 46–60.

Tian, X., Bar-Shalom, Y., and Pattipati, K. R., Multi-step Look-Ahead Policy for Autonomous Cooperative Surveillance by UAVs in Hostile Environments, *JAIF*, **5**, 1 (July 2010), 3–17.

Tian, X. and Bar-Shalom, Y., Algorithms for Asynchronous Track-to-Track Fusion *JAIF*, **5**, 2 (Dec. 2010), 128–138.

Tian, X., *see* Yuan, T., *JAIF*, **6**, 2 (December 2011), 131–149.

Tian, X. and Bar-Shalom, Y., Track-to-Track Fusion Configurations and Association in a Sliding Window *JAIF*, **4**, 2 (Dec. 2009), 146–164.

Tu, F., *see* Han, X., *JAIF*, **7**, 1 (June 2012), 78–96.

V

Valin, P., Djiknavorian, P., and Bossé, É., A Pragmatic Approach for the use of Dempster-Shafer Theory in Fusing Realistic Sensor Data, *JAIF*, **5**, 1 (July 2010), 32–40.

Velmurugan, R., *see* Panakkal, V. P., *JAIF*, **10**, 2 (December 2015), 143–162.

W

Washburn, R. B. and Schneider, M. K., Optimal Policies for a Class of Restless Multiarmed Bandit Scheduling Problems with Applications to Sensor Management *JAIF*, **3**, 1 (July 2008), 3–13.

Willett, P., *see* Choi, S., *JAIF*, **8**, 2 (December 2013), 143–155.

Willett, P., *see* Choi, S., *JAIF*, **9**, 1 (July 2014), 27–37.

Willett, P., *see* Coraluppi, S., *JAIF*, **5**, 1 (July 2010), 18–31.

Willett, P., *see* Crouse, D. F., *JAIF*, **4**, 2 (Dec. 2009), 93–116.

Willett, P., *see* Crouse, D. F., *JAIF*, **7**, 1 (June 2012), 97–98.

Willett, P., *see* Crouse, D. F., *JAIF*, **8**, 1 (July 2013), 73–89.

Willett, P., *see* Dou, W., *JAIF*, **10**, 2 (December 2015), 163–182.

Willett, P., *see* Erdinc, O., *JAIF*, **2**, 1 (July 2007), 22–34.

Willett, P., *see* Georgescu, R., *JAIF*, **7**, 1 (June 2012), 61–77.

Willett, P., *see* Osborne, R. W., III, *JAIF*, **7**, 1 (June 2012), 3–15.

Willett, P., *see* Osborne, R. W., III, *JAIF*, **10**, 2 (December 2015), 199–210.

Willett, P., *see* Romeo, K., *JAIF*, **10**, 2 (December 2015), 113–124.

Willett, P., *see* Song, X., *JAIF*, **7**, 1 (June 2012), 16–27.

Willett, P., *see* Song, X., *JAIF*, **8**, 1 (July 2013), 3–14.

Wu, J., *see* Jilkov, V. P., *JAIF*, **10**, 1 (June 2015), 73–88.

Y

Yager, R., *see* Sudit, M., *JAIF*, **2**, 1 (July 2007), 3–21.

Yang, C. and Blasch, E., Fusion of Tracks with Road Constraints *JAIF*, **3**, 1 (July 2008), 14–32.

Yang, R., *see* Foo, P. H., *JAIF*, **4**, 1 (July 2009), 3–26.

Yu, F., *see* Han, X., *JAIF*, **7**, 1 (June 2012), 78–96.

Yuan, T., Bar-Shalom, Y., Tian, X., Heterogeneous Track-to-Track Fusion, *JAIF*, **6**, 2 (December 2011), 131–149.

Z

Zea, A., *see* Faion, F., *JAIF*, **10**, 1 (June 2015), 13–30.

Zhang, S. and Bar-Shalom, Y., Practical Data Association for Passive Sensors in 3D *JAIF*, **9**, 1 (July 2014), 38–46.

Zhang, S., Bar-Shalom, Y., Watson, G., Tracking with Multisensor Out-of-Sequence Measurements with Residual Biases, *JAIF*, **6**, 1 (June 2011), 3–23.
Zheng, L., *see* Crouse, D. F., *JAIF*, **7**, 1 (June 2012), 97–98.
Zheng, Y. and Blasch, E., An Exploration of the Impacts of Three Factors in Multimodal Biometric Score Fusion: Score Modality, Recognition Method, and Fusion Process *JAIF*, **9**, 2 (December 2014), 124–135.
Zhou, S., *see* Choi, S., *JAIF*, **9**, 1 (July 2014), 27–37.
Zhou, S., *see* Song, X., *JAIF*, **7**, 1 (June 2012), 16–27.
Zhou, S., *see* Song, X., *JAIF*, **8**, 1 (July 2013), 3–14.

SUBJECT INDEX

0–1 nonlinear optimization
Papageorgiou, D. J., +, *JAIF*, **6**, 2 (December 2011), 77–100.

A

acoustic localization
Osborne, R. W., III, +, *JAIF*, **9**, 2 (December 2014), 75–89.
adaptive encoding
Habtemariam, B. K., +, *JAIF*, **7**, 2 (December 2012), 114–130.
Adaptive Gaussian Mixture Models
Terejanu, G., +, *JAIF*, **5**, 2 (Dec. 2010), 73–87.
adaptive model selection
Tharmarasa, R., +, *JAIF*, **7**, 1 (June 2012), 46–60.
air defense
Johansson, F., +, *JAIF*, **6**, 2 (December 2011), 187–199.
alerting
Schrag, R. C., +, *JAIF*, **2**, 2 (Dec. 2007), 77–98.
algorithm selection
Peel, L., *JAIF*, **6**, 2 (December 2011), 119–130.
ambiguity management
Papageorgiou, D. J., +, *JAIF*, **6**, 2 (December 2011), 77–100.
angular quantities
Kurz, G., +, *JAIF*, **9**, 2 (December 2014), 90–105.
anomaly detection
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
approximate crosscorrelation
Bar-Shalom, Y., +, *JAIF*, **3**, 2 (Dec. 2008), 107–117.
assignment
Areta, J., +, *JAIF*, **2**, 2 (Dec. 2007), 113–127.
Association ambiguity
Choi, S., +, *JAIF*, **9**, 1 (July 2014), 27–37.
association benchmark
Areta, J., +, *JAIF*, **1**, 2 (Dec. 2006), 144–157.
association
Areta, J., +, *JAIF*, **4**, 1 (July 2009), 40–51.
Bar-Shalom, Y., +, *JAIF*, **2**, 1 (July 2007), 49–59.
Ferry, J. P., *JAIF*, **5**, 1 (July 2010), 41–67.
Hamp, Q., +, *JAIF*, **7**, 2 (December 2012), 131–140.
Tian, X., +, *JAIF*, **4**, 2 (Dec. 2009), 146–164.
asynchronous
Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.
Asynchronous Track-to-Track fusion
Tian, X., +, *JAIF*, **5**, 2 (Dec. 2010), 128–138.
augmented state
Chakravorty, R., +, *JAIF*, **1**, 1 (July 2006), 63–74.
Osborne, R. W., III, +, *JAIF*, **7**, 1 (June 2012), 3–15.
automatic target recognition (ATR)
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.

B

ballistic targets
Tharmarasa, R., +, *JAIF*, **7**, 1 (June 2012), 46–60.
Bayesian
Ferry, J. P., *JAIF*, **5**, 1 (July 2010), 41–67.
Bayesian and Reasoning Methods
Arnborg, S., *JAIF*, **1**, 1 (July 2006), 75–90.
Brynielsson, J., +, *JAIF*, **1**, 2 (Dec. 2006), 108–121.
Ferry, J. P., *JAIF*, **5**, 1 (July 2010), 41–67.
Stone, L. D., +, *JAIF*, **10**, 1 (June 2015), 3–12.
Terejanu, G., +, *JAIF*, **5**, 2 (Dec. 2010), 73–87.
Valin, P., +, *JAIF*, **5**, 1 (July 2010), 32–40.
Washburn, R. B., +, *JAIF*, **3**, 1 (July 2008), 3–13.
Bayesian Data Fusion
Junghans, M., +, *JAIF*, **3**, 1 (July 2008), 50–62.
Bayesian filtering
Blom, H. A. P., +, *JAIF*, **1**, 1 (July 2006), 15–34.
Bayesian Game
Brynielsson, J., +, *JAIF*, **1**, 2 (Dec. 2006), 108–121.
Bayesian Networks (BNs)
Junghans, M., +, *JAIF*, **3**, 1 (July 2008), 50–62.
Bayesian network
Chang, K. C., +, *JAIF*, **1**, 2 (Dec. 2006), 95–107.
Bayesian theory
Karlsson, A., +, *JAIF*, **6**, 2 (December 2011), 150–166.
Bayesian tracking
Aughenbaugh, J. M., +, *JAIF*, **5**, 2 (Dec. 2010), 108–127.
bearings-only target motion analysis
Jauffret, C., +, *JAIF*, **6**, 1 (June 2011), 24–38.
benchmark performance
Howell, C., +, *JAIF*, **8**, 2 (December 2013), 133–142.
belief function theory
Martin, A., +, *JAIF*, **3**, 2 (Dec. 2008), 67–89.
bias
Ferry, J. P., *JAIF*, **5**, 1 (July 2010), 41–67.
Bias estimation
Belfadel, D., +, *JAIF*, **9**, 2 (December 2014), 59–74.
Belfadel, D., +, *JAIF*, **10**, 2 (December 2015), 101–112.
bias observability
Belfadel, D., +, *JAIF*, **9**, 2 (December 2014), 59–74.
biased measurement
Zhang, S., +, *JAIF*, **6**, 1 (June 2011), 3–23.
bias-variance trade-off
George, J., +, *JAIF*, **8**, 1 (July 2013), 15–32.
BOMTMA
Jauffret, C., +, *JAIF*, **6**, 1 (June 2011), 24–38.
BOTMA
Jauffret, C., +, *JAIF*, **6**, 1 (June 2011), 24–38.
brake-by-wire
Hoseinnezhad, R., +, *JAIF*, **1**, 1 (July 2006), 52–62.

C

capacities
Arnborg, S., *JAIF*, **1**, 1 (July 2006), 75–90.
cardinality tracking
Coraluppi, S., +, *JAIF*, **7**, 2 (December 2012), 153–164.
centralized fusion
Rodningsby, A., +, *JAIF*, **4**, 2 (Dec. 2009), 117–145.
circular data
Kurz, G., +, *JAIF*, **9**, 2 (December 2014), 90–105.
circular statistics
Crouse, D. F., +, *JAIF*, **8**, 1 (July 2013), 73–89.
classification
Hamp, Q., +, *JAIF*, **7**, 2 (December 2012), 131–140.
Classification, Learning, Data Mining
Jabeur, K., +, *JAIF*, **4**, 2 (Dec. 2009), 75–92.
Martin, A., +, *JAIF*, **3**, 2 (Dec. 2008), 67–89.
Sambhoos, K., +, *JAIF*, **3**, 2 (Dec. 2008), 90–106.
clustering
Sambhoos, K., +, *JAIF*, **3**, 2 (Dec. 2008), 90–106.
clutter suppression
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.
collation
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
Combat Power Management
Benaskeur, A. R., +, *JAIF*, **2**, 2 (Dec. 2007), 99–112.
combination
Areta, J., +, *JAIF*, **4**, 1 (July 2009), 40–51.
Belaroussi, R., +, *JAIF*, **1**, 1 (July 2006), 35–51.
combination operators
Karlsson, A., +, *JAIF*, **6**, 2 (December 2011), 150–166.
combinatorial auctions
Avasarala, V., +, *JAIF*, **4**, 1 (July 2009), 52–71.
Command and Control
Brynielsson, J., +, *JAIF*, **1**, 2 (Dec. 2006), 108–121.
communication rate reduction
Schlosser, M. S., +, *JAIF*, **2**, 2 (Dec. 2007), 65–76.
Community detection
Peel, L., *JAIF*, **6**, 2 (December 2011), 119–130.
composite measurement
Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.

composite measurements
Belfadel, D., +, *JAIF*, **9**, 2 (December 2014), 59–74.
Belfadel, D., +, *JAIF*, **10**, 2 (December 2015), 101–112.
Con-Tracker
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
conflict management
Martin, A., +, *JAIF*, **3**, 2 (Dec. 2008), 67–89.
conservative fusion approach
Schlosser, M. S., +, *JAIF*, **2**, 2 (Dec. 2007), 65–76.
Content-based Image Retrieval
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
context-aware modeling
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
Continuous time systems
Crouse, D. F., +, *JAIF*, **10**, 1 (June 2015), 31–57.
convergent analysis
Carvalho, R. N., +, *JAIF*, **7**, 2 (December 2012), 141–152.
coordinated turn
Yuan, T., +, *JAIF*, **6**, 2 (December 2011), 131–149.
Cramér-Rao lower bound
Jauffret, C., +, *JAIF*, **6**, 1 (June 2011), 24–38.
credal sets
Karlsson, A., +, *JAIF*, **6**, 2 (December 2011), 150–166.
CRLB
Belfadel, D., +, *JAIF*, **9**, 2 (December 2014), 59–74.
Belfadel, D., +, *JAIF*, **10**, 2 (December 2015), 101–112.
Osborne, R. W., III, +, *JAIF*, **9**, 2 (December 2014), 75–89.
Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.

D

data association
Areta, J., +, *JAIF*, **1**, 2 (Dec. 2006), 144–157.
Osborne, R. W., III, +, *JAIF*, **7**, 1 (June 2012), 3–15.
Rodningsby, A., +, *JAIF*, **4**, 2 (Dec. 2009), 117–145.
Zhang, S., +, *JAIF*, **9**, 1 (July 2014), 38–46.
data fusion
Chen, G., +, *JAIF*, **2**, 1 (July 2007), 35–48.
Foo, P. H., +, *JAIF*, **8**, 1 (July 2013), 33–72.
Papageorgiou, D. J., +, *JAIF*, **6**, 2 (December 2011), 77–100.
Data Qualification
Junghans, M., +, *JAIF*, **3**, 1 (July 2008), 50–62.
Decentralized Kalman Filter (DKF)
Schlosser, M. S., +, *JAIF*, **2**, 2 (Dec. 2007), 65–76.
decision making
Terejanu, G., +, *JAIF*, **5**, 2 (Dec. 2010), 73–87.
Decision-Level Fusion (DLF)
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.
Dempster-Shafer
Stone, L. D., +, *JAIF*, **10**, 1 (June 2015), 3–12.
Valin, P., +, *JAIF*, **5**, 1 (July 2010), 32–40.
Density Forecasting
Terejanu, G., +, *JAIF*, **5**, 2 (Dec. 2010), 73–87.
depth camera
Kurz, G., +, *JAIF*, **9**, 1 (July 2014), 13–26.
descision fusion
Rogova, G. L., +, *JAIF*, **3**, 2 (Dec. 2008), 118–128.
descriptor system
Blom, H. A. P., +, *JAIF*, **1**, 1 (July 2006), 15–34.
detection
Davey, S. J., *JAIF*, **6**, 2 (December 2011), 167–186.
Jilkov, V. P., +, *JAIF*, **4**, 1 (July 2009), 27–39.
Differentiation
Streit, R., +, *JAIF*, **9**, 1 (July 2014), 3–12.
Digital audio/video broadcasting
Choi, S., +, *JAIF*, **9**, 1 (July 2014), 27–37.
Dirac Mixtures
Steinbring, J., +, *JAIF*, **9**, 2 (December 2014), 106–123.
direction of arrival (DoA) estimation
Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.
directional statistics
Crouse, D. F., +, *JAIF*, **8**, 1 (July 2013), 73–89.
Kurz, G., +, *JAIF*, **9**, 2 (December 2014), 90–105.
discounting operators
Karlsson, A., +, *JAIF*, **6**, 2 (December 2011), 150–166.
Displaced Phase Center Antenna (DPCA)
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.
distributed detection
Coraluppi, S., +, *JAIF*, **5**, 1 (July 2010), 18–31.
distributed inference
Choi, S., +, *JAIF*, **8**, 2 (December 2013), 143–155.
distributed tracking
Habtemariam, B. K., +, *JAIF*, **7**, 2 (December 2012), 114–130.
DS-structures
Arnborg, S., *JAIF*, **1**, 1 (July 2006), 75–90.
DSmT
Martin, A., +, *JAIF*, **3**, 2 (Dec. 2008), 67–89.
dynamic programming
Washburn, R. B., +, *JAIF*, **3**, 1 (July 2008), 3–13.

E

early fusion
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
EM Algorithm
Crouse, D. F., +, *JAIF*, **4**, 2 (Dec. 2009), 93–116.
entropy
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
estimator efficiency
Osborne, R. W., III, +, *JAIF*, **9**, 2 (December 2014), 75–89.
Expected Loss
Terejanu, G., +, *JAIF*, **5**, 2 (Dec. 2010), 73–87.
experiment design
Schrag, R. C., +, *JAIF*, **2**, 2 (Dec. 2007), 77–98.
experimental results
George, J., +, *JAIF*, **8**, 1 (July 2013), 15–32.
Extended Object Tracking
Steinbring, J., +, *JAIF*, **9**, 2 (December 2014), 106–123.

F

false measurements
Blom, H. A. P., +, *JAIF*, **1**, 1 (July 2006), 15–34.
fault diagnosis
Choi, S., +, *JAIF*, **8**, 2 (December 2013), 143–155.
features
Ferry, J. P., *JAIF*, **5**, 1 (July 2010), 41–67.
feature extraction
Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.
feature-aided tracking
Coraluppi, S., +, *JAIF*, **6**, 1 (June 2011), 57–67.
FIM
Osborne, R. W., III, +, *JAIF*, **9**, 2 (December 2014), 75–89.
Finite point processes, Multitarget tracking
Streit, R., +, *JAIF*, **9**, 1 (July 2014), 3–12.
Fire Control
Benaskeur, A. R., +, *JAIF*, **2**, 2 (Dec. 2007), 99–112.
flight profile
Foo, P. H., +, *JAIF*, **4**, 1 (July 2009), 3–26.
fuse-before-track
Coraluppi, S., +, *JAIF*, **5**, 1 (July 2010), 18–31.
fusion
Crouse, D. F., +, *JAIF*, **4**, 2 (Dec. 2009), 93–116.
Foo, P. H., +, *JAIF*, **4**, 1 (July 2009), 3–26.
Jilkov, V. P., +, *JAIF*, **4**, 1 (July 2009), 27–39.
Osborne, R. W., III, +, *JAIF*, **9**, 2 (December 2014), 75–89.
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
Tian, X., +, *JAIF*, **4**, 2 (Dec. 2009), 146–164.
Fusion Applications
Areta, J., +, *JAIF*, **1**, 2 (Dec. 2006), 144–157.
Benaskeur, A. R., +, *JAIF*, **2**, 2 (Dec. 2007), 99–112.
Chen, G., +, *JAIF*, **2**, 1 (July 2007), 35–48.
Foo, P. H., +, *JAIF*, **4**, 1 (July 2009), 3–26.
Hoseinnezhad, R., +, *JAIF*, **1**, 1 (July 2006), 52–62.
Junghans, M., +, *JAIF*, **3**, 1 (July 2008), 50–62.
Rogova, G. L., +, *JAIF*, **3**, 2 (Dec. 2008), 118–128.
Schrag, R. C., +, *JAIF*, **2**, 2 (Dec. 2007), 77–98.
Yang, C., +, *JAIF*, **3**, 1 (July 2008), 14–32.
Fusion Architectures and Management Issues
Areta, J., +, *JAIF*, **4**, 1 (July 2009), 40–51.
Avasarala, V., +, *JAIF*, **4**, 1 (July 2009), 52–71.
Blasch, E., +, *JAIF*, **1**, 2 (Dec. 2006), 122–143.
Chang, K. C., +, *JAIF*, **1**, 2 (Dec. 2006), 95–107.
Coraluppi, S., +, *JAIF*, **5**, 1 (July 2010), 18–31.

Schlosser, M. S., +, *JAIF*, **2**, 2 (Dec. 2007), 65–76.
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
Tian, X., +, *JAIF*, **5**, 1 (July 2010), 3–17.
fusion of clustering results
Rogova, G. L., +, *JAIF*, **3**, 2 (Dec. 2008), 118–128.
fuzzy
Stone, L. D., +, *JAIF*, **10**, 1 (June 2015), 3–12.
fuzzy hamming distance
Sambhoos, K., +, *JAIF*, **3**, 2 (Dec. 2008), 90–106.
fuzzy inference
Belaroussi, R., +, *JAIF*, **1**, 1 (July 2006), 35–51.
fuzzy logic
Foo, P. H., +, *JAIF*, **4**, 1 (July 2009), 3–26.
fuzzy rule-base
Hoseinnezhad, R., +, *JAIF*, **1**, 1 (July 2006), 52–62.

G

game theory
Brynielsson, J., +, *JAIF*, **1**, 2 (Dec. 2006), 108–121.
Chen, G., +, *JAIF*, **2**, 1 (July 2007), 35–48.
Gauss-Newton method
George, J., +, *JAIF*, **8**, 1 (July 2013), 15–32.
genetic algorithms
Avasarala, V., +, *JAIF*, **4**, 1 (July 2009), 52–71.
Jabeur, K., +, *JAIF*, **4**, 2 (Dec. 2009), 75–92.
Geodesy
Crouse, D. F., +, *JAIF*, **10**, 1 (June 2015), 31–57.
geological data
Rogova, G. L., +, *JAIF*, **3**, 2 (Dec. 2008), 118–128.
global
Areta, J., +, *JAIF*, **2**, 2 (Dec. 2007), 113–127.
Graph Matching
Han, X., +, *JAIF*, **7**, 1 (June 2012), 78–96.
grid-based filtering
Aughenbaugh, J. M., +, *JAIF*, **5**, 2 (Dec. 2010), 108–127.
GMTI
Salerno, E., +, *JAIF*, **10**, 1 (June 2015), 58–72.
gunfire detection system
George, J., +, *JAIF*, **8**, 1 (July 2013), 15–32.

H

heterogeneous track-to-track fusion
Yuan, T., +, *JAIF*, **6**, 2 (December 2011), 131–149.
hidden Markov model
An, W., +, *JAIF*, **3**, 1 (July 2008), 33–49.
hidden Markov model (HMM)
Zheng, Y., +, *JAIF*, **9**, 2 (December 2014), 124–135.
hierarchical entity aggregation
Chen, G., +, *JAIF*, **2**, 1 (July 2007), 35–48.
High Level Data Fusion
Sambhoos, K., +, *JAIF*, **3**, 2 (Dec. 2008), 90–106.
High Range Resolution (HRR)
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.
histogram PMHT
Davey, S. J., *JAIF*, **6**, 2 (December 2011), 167–186.
Hough transform
Belaroussi, R., +, *JAIF*, **1**, 1 (July 2006), 35–51.
Salerno, E., +, *JAIF*, **10**, 1 (June 2015), 58–72.
human cognition
Foo, P. H., +, *JAIF*, **4**, 1 (July 2009), 3–26.
Human-Fusion Interaction
Blasch, E., +, *JAIF*, **1**, 2 (Dec. 2006), 122–143.
human identification
Zheng, Y., +, *JAIF*, **9**, 2 (December 2014), 124–135.
Human-Machine Interface
Blasch, E., +, *JAIF*, **1**, 2 (Dec. 2006), 122–143.
hybrid relevance feedback
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
hypercube distance
Sambhoos, K., +, *JAIF*, **3**, 2 (Dec. 2008), 90–106.
hypotheses
Areta, J., +, *JAIF*, **4**, 1 (July 2009), 40–51.
hypothesis generator
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
hypothesis management
Coraluppi, S., +, *JAIF*, **6**, 1 (June 2011), 57–67.

I

Image Fusion
Belaroussi, R., +, *JAIF*, **1**, 1 (July 2006), 35–51.
image fusion performance assessment
Howell, C., +, *JAIF*, **8**, 2 (December 2013), 133–142.
image fusion task performance
Howell, C., +, *JAIF*, **8**, 2 (December 2013), 133–142.
imetrics
Schrag, R. C., +, *JAIF*, **2**, 2 (Dec. 2007), 77–98.
imprecise
Stone, L. D., +, *JAIF*, **10**, 1 (June 2015), 3–12.
imprecise probability
Arnborg, S., *JAIF*, **1**, 1 (July 2006), 75–90.
Karlsson, A., +, *JAIF*, **6**, 2 (December 2011), 150–166.
index rules
Washburn, R. B., +, *JAIF*, **3**, 1 (July 2008), 3–13.
indicator
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
INFERD
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
Influence Diagram
Brynielsson, J., +, *JAIF*, **1**, 2 (Dec. 2006), 108–121.
Information and data fusion
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
information extraction
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
information fusion
Bar-Shalom, Y., +, *JAIF*, **1**, 1 (July 2006), 3–14.
Blasch, E., +, *JAIF*, **1**, 2 (Dec. 2006), 122–143.
Foo, P. H., +, *JAIF*, **8**, 1 (July 2013), 33–72.
Jabeur, K., +, *JAIF*, **4**, 2 (Dec. 2009), 75–92.
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
information gain heuristic
An, W., +, *JAIF*, **3**, 1 (July 2008), 33–49.
information heuristics
Choi, S., +, *JAIF*, **8**, 2 (December 2013), 143–155.
Information Retrieval
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
intelligence
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
intent inference
Foo, P. H., +, *JAIF*, **4**, 1 (July 2009), 3–26.
intensity filter
Streit, R., *JAIF*, **8**, 2 (December 2013), 119–132.
interaction networks
Peel, L., *JAIF*, **6**, 2 (December 2011), 119–130.
IPDA
Chakravorty, R., +, *JAIF*, **1**, 1 (July 2006), 63–74.
Iterative solution
Yang, C., +, *JAIF*, **3**, 1 (July 2008), 14–32.

K

Kalman filter
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
Knowledge Representation
Blasch, E., +, *JAIF*, **1**, 2 (Dec. 2006), 122–143.

L

L2/L3 fusion
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
Lagrangian multiplier
Yang, C., +, *JAIF*, **3**, 1 (July 2008), 14–32.
late fusion
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
launch point estimation
Tharmarasa, R., +, *JAIF*, **7**, 1 (June 2012), 46–60.
LCD
Steinbring, J., +, *JAIF*, **9**, 2 (December 2014), 106–123.
least squares
Song, X., +, *JAIF*, **8**, 1 (July 2013), 3–14.
least squares estimation
Tharmarasa, R., +, *JAIF*, **7**, 1 (June 2012), 46–60.
legacy sensors
Bar-Shalom, Y., +, *JAIF*, **3**, 2 (Dec. 2008), 107–117.

level 2 fusion
Chang, K. C., +, *JAIF*, **1**, 2 (Dec. 2006), 95–107.
level 4 data fusion
Avasarala, V., +, *JAIF*, **4**, 1 (July 2009), 52–71.
likelihood
Arnborg, S., *JAIF*, **1**, 1 (July 2006), 75–90.
Stone, L. D., +, *JAIF*, **10**, 1 (June 2015), 3–12.
likelihood sampling
Aughenbaugh, J. M., +, *JAIF*, **5**, 2 (Dec. 2010), 108–127.
linear minimum mean square error
Yuan, T., +, *JAIF*, **6**, 2 (December 2011), 131–149.
link-analysis
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
localization
Song, X., +, *JAIF*, **8**, 1 (July 2013), 3–14.
LOS
Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.
LRKF
Steinbring, J., +, *JAIF*, **9**, 2 (December 2014), 106–123.

M

m-Best Soft Assignment Algorithm
Han, X., +, *JAIF*, **7**, 1 (June 2012), 78–96.
Mahler/Fixs?n rule
Arnborg, S., *JAIF*, **1**, 1 (July 2006), 75–90.
Maneuvering target
Jauffret, C., +, *JAIF*, **6**, 1 (June 2011), 24–38.
maximum likelihood
Belfadel, D., +, *JAIF*, **10**, 2 (December 2015), 101–112.
Belfadel, D., +, *JAIF*, **9**, 2 (December 2014), 59–74.
Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.
Song, X., +, *JAIF*, **8**, 1 (July 2013), 3–14.
maximum likelihood estimate
Jauffret, C., +, *JAIF*, **6**, 1 (June 2011), 24–38.
maximum likelihood estimation
Crouse, D. F., +, *JAIF*, **8**, 1 (July 2013), 73–89.
George, J., +, *JAIF*, **8**, 1 (July 2013), 15–32.
maximum likelihood fusion
Yuan, T., +, *JAIF*, **6**, 2 (December 2011), 131–149.
measurement fusion
Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.
measurement origin uncertainty
Panakkal, V. P., +, *JAIF*, **10**, 2 (December 2015), 143–162.
Medical Decision Support Systems
Alsun, M., +, *JAIF*, **7**, 2 (December 2012), 101–113.
Medical Reasoning
Alsun, M., +, *JAIF*, **7**, 2 (December 2012), 101–113.
MHT
Areta, J., +, *JAIF*, **4**, 1 (July 2009), 40–51.
military application
Jabeur, K., +, *JAIF*, **4**, 2 (Dec. 2009), 75–92.
MIMO radar
Song, X., +, *JAIF*, **7**, 1 (June 2012), 16–27.
minimum variance distortionless response (MVDR)
Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.
misassociation
Areta, J., +, *JAIF*, **2**, 2 (Dec. 2007), 113–127.
ML-PMHT
Romeo, K., +, *JAIF*, **10**, 2 (December 2015), 113–124.
model-based classifier
Belaroussi, R., +, *JAIF*, **1**, 1 (July 2006), 35–51.
modelling
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
moving target identification
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.
multi-channel signal subspace (MSS)
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.
multi-criteria classification
Jabeur, K., +, *JAIF*, **4**, 2 (Dec. 2009), 75–92.
multi-hypothesis tracking
Coraluppi, S., +, *JAIF*, **5**, 1 (July 2010), 18–31.
Coraluppi, S., +, *JAIF*, **6**, 1 (June 2011), 57–67.
multi-sensor multi-target tracking
Coraluppi, S., +, *JAIF*, **6**, 1 (June 2011), 57–67.
multi-sensor target tracking
Schlosser, M. S., +, *JAIF*, **2**, 2 (Dec. 2007), 65–76.
multidimensional assignment
Areta, J., +, *JAIF*, **1**, 2 (Dec. 2006), 144–157.
Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.
Multimedia Retrieval
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
Multimodal biometric score fusion
Zheng, Y., +, *JAIF*, **9**, 2 (December 2014), 124–135.
multiple dimension assignment (MDA)
Zhang, S., +, *JAIF*, **9**, 1 (July 2014), 38–46.
multiobjective optimization
Jilkov, V. P., +, *JAIF*, **4**, 1 (July 2009), 27–39.
multipath
Romeo, K., +, *JAIF*, **10**, 2 (December 2015), 113–124.
multiple-hypothesis tracking
Coraluppi, S., +, *JAIF*, **7**, 2 (December 2012), 153–164.
Multiple-Model Adaptive Estimator
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
multisensor tracking
Bar-Shalom, Y., +, *JAIF*, **1**, 1 (July 2006), 3–14.
Blair, W. D., +, *JAIF*, **7**, 1 (June 2012), 28–45.
Yuan, T., +, *JAIF*, **6**, 2 (December 2011), 131–149.
multispectral face recognition
Zheng, Y., +, *JAIF*, **9**, 2 (December 2014), 124–135.
multistatic
Erdinc, O., +, *JAIF*, **2**, 1 (July 2007), 22–34.
multitarget multisensor
Rodningsby, A., +, *JAIF*, **4**, 2 (Dec. 2009), 117–145.
Multitarget Multisensor Tracking
Georgescu, R., +, *JAIF*, **7**, 1 (June 2012), 61–77.
multitarget tracking filters
Streit, R., *JAIF*, **8**, 2 (December 2013), 119–132.
mutli-sensor tracking
Coraluppi, S., +, *JAIF*, **5**, 1 (July 2010), 18–31.

N

Naval Anti-Air Warfare
Benaskeur, A. R., +, *JAIF*, **2**, 2 (Dec. 2007), 99–112.
nearest neighbor
Areta, J., +, *JAIF*, **2**, 2 (Dec. 2007), 113–127.
neural networks
Belaroussi, R., +, *JAIF*, **1**, 1 (July 2006), 35–51.
Nonlinear filtering
Jilkov, V. P., +, *JAIF*, **10**, 1 (June 2015), 73–00.
Nonlinear Kalman Filtering
Steinbring, J., +, *JAIF*, **9**, 2 (December 2014), 106–123.
non-uniform quantization
Habtemariam, B. K., +, *JAIF*, **7**, 2 (December 2012), 114–130.

O

observability
Belfadel, D., +, *JAIF*, **10**, 2 (December 2015), 101–112.
OFDM modulation
Choi, S., +, *JAIF*, **9**, 1 (July 2014), 27–37.
Organizational Structures Identification
Han, X., +, *JAIF*, **7**, 1 (June 2012), 78–96.
OTHR
Romeo, K., +, *JAIF*, **10**, 2 (December 2015), 113–124.
out-of-sequence measurement
Zhang, S., +, *JAIF*, **6**, 1 (June 2011), 3–23.

P

parallel and distributed computing, GPU
Jilkov, V. P., +, *JAIF*, **10**, 1 (June 2015), 73–00.
particle filter
Jilkov, V. P., +, *JAIF*, **10**, 1 (June 2015), 73–00.
particle filtering
Blom, H. A. P., +, *JAIF*, **1**, 1 (July 2006), 15–34.
particle flow filter
Jilkov, V. P., +, *JAIF*, **10**, 1 (June 2015), 73–00.
passive acoustic sensor network
Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.
Passive radar
Choi, S., +, *JAIF*, **9**, 1 (July 2014), 27–37.

passive sensor
Zhang, S., +, *JAIF*, **9**, 1 (July 2014), 38–46.
PCRLB, LFM
Song, X., +, *JAIF*, **7**, 1 (June 2012), 16–27.
PDA
Panakkal, V. P., +, *JAIF*, **10**, 2 (December 2015), 143–162.
performance analysis
Schlosser, M. S., +, *JAIF*, **2**, 2 (Dec. 2007), 65–76.
performance evaluation
Schrag, R. C., +, *JAIF*, **2**, 2 (Dec. 2007), 77–98.
performance model
Carvalho, R. N., +, *JAIF*, **7**, 2 (December 2012), 141–152.
Performance Prediction
Blair, W. D., +, *JAIF*, **7**, 1 (June 2012), 28–45.
PMHT
Crouse, D. F., +, *JAIF*, **4**, 2 (Dec. 2009), 93–116.
point cloud
Kurz, G., +, *JAIF*, **9**, 1 (July 2014), 13–26.
point processes
Streit, R., *JAIF*, **8**, 2 (December 2013), 119–132.
Possibilistic Reasoning
Alsun, M., +, *JAIF*, **7**, 2 (December 2012), 101–113.
posterior Cramer-Rao lower bound
Habtemariam, B. K., +, *JAIF*, **7**, 2 (December 2012), 114–130.
predetection Fusion
Georgescu, R., +, *JAIF*, **7**, 1 (June 2012), 61–77.
Probabilistic Multi-Hypothesis Tracker (PMHT)
Choi, S., +, *JAIF*, **9**, 1 (July 2014), 27–37.
probability
Areta, J., +, *JAIF*, **2**, 2 (Dec. 2007), 113–127.
Ferry, J. P., *JAIF*, **5**, 1 (July 2010), 41–67.
Probability generating function
Streit, R., +, *JAIF*, **9**, 1 (July 2014), 3–12.
Probability generating functional
Streit, R., +, *JAIF*, **9**, 1 (July 2014), 3–12.
probability hypothesis density
Streit, R., *JAIF*, **8**, 2 (December 2013), 119–132.
Probability hypothesis density filter
Habtemariam, B. K., +, *JAIF*, **7**, 2 (December 2012), 114–130.
profile-free method
Tharmarasa, R., +, *JAIF*, **7**, 1 (June 2012), 46–60.
proportional conflict redistribution rules
Martin, A., +, *JAIF*, **3**, 2 (Dec. 2008), 67–89.

Q

Quadratic Assignment Problem
Han, X., +, *JAIF*, **7**, 1 (June 2012), 78–96.
qualitative beliefs
Martin, A., +, *JAIF*, **3**, 2 (Dec. 2008), 67–89.
quantized
Stone, L. D., +, *JAIF*, **10**, 1 (June 2015), 3–12.

R

radar
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.
random sets
Stone, L. D., +, *JAIF*, **10**, 1 (June 2015), 3–12.
Real-time Fusion
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
realistic sensor data
Valin, P., +, *JAIF*, **5**, 1 (July 2010), 32–40.
reasoning
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
recursive filtering
Kurz, G., +, *JAIF*, **9**, 2 (December 2014), 90–105.
reflection
Song, X., +, *JAIF*, **8**, 1 (July 2013), 3–14.
resource allocation
Avasarala, V., +, *JAIF*, **4**, 1 (July 2009), 52–71.
resource management
Carvalho, R. N., +, *JAIF*, **7**, 2 (December 2012), 141–152.
Johansson, F., +, *JAIF*, **6**, 2 (December 2011), 187–199.
restless bandits
Washburn, R. B., +, *JAIF*, **3**, 1 (July 2008), 3–13.
risk
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
road map
Yang, C., +, *JAIF*, **3**, 1 (July 2008), 14–32.
road network extraction
Salerno, E., +, *JAIF*, **10**, 1 (June 2015), 58–72.
rollout algorithm
An, W., +, *JAIF*, **3**, 1 (July 2008), 33–49.

S

S^2KF
Steinbring, J., +, *JAIF*, **9**, 2 (December 2014), 106–123.
***S*-D algorithm**
Zhang, S., +, *JAIF*, **9**, 1 (July 2014), 38–46.
Sampling
Steinbring, J., +, *JAIF*, **9**, 2 (December 2014), 106–123.
Schmidt-Kalman filter
Zhang, S., +, *JAIF*, **6**, 1 (June 2011), 3–23.
score fusion evaluation metric
Zheng, Y., +, *JAIF*, **9**, 2 (December 2014), 124–135.
Secular function
Streit, R., +, *JAIF*, **9**, 1 (July 2014), 3–12.
Selection Bias
Junghans, M., +, *JAIF*, **3**, 1 (July 2008), 50–62.
self-fusion
Howell, C., +, *JAIF*, **8**, 2 (December 2013), 133–142.
sensor assignment
An, W., +, *JAIF*, **3**, 1 (July 2008), 33–49.
sensor fusion
Blair, W. D., +, *JAIF*, **7**, 1 (June 2012), 28–45.
Carvalho, R. N., +, *JAIF*, **7**, 2 (December 2012), 141–152.
Sensor Layout
Erdinc, O., +, *JAIF*, **2**, 1 (July 2007), 22–34.
sensor localization
Crouse, D. F., +, *JAIF*, **8**, 1 (July 2013), 73–89.
sensor management
Avasarala, V., +, *JAIF*, **4**, 1 (July 2009), 52–71.
Washburn, R. B., +, *JAIF*, **3**, 1 (July 2008), 3–13.
Sensor network
Song, X., +, *JAIF*, **8**, 1 (July 2013), 3–14.
Sensor Networks
Georgescu, R., +, *JAIF*, **7**, 1 (June 2012), 61–77.
sensor resource management
Chang, K. C., +, *JAIF*, **1**, 2 (Dec. 2006), 95–107.
sensor scheduling
An, W., +, *JAIF*, **3**, 1 (July 2008), 33–49.
Shooter localization
George, J., +, *JAIF*, **8**, 1 (July 2013), 15–32.
Similarity Estimation
Alsun, M., +, *JAIF*, **7**, 2 (December 2012), 101–113.
similarity index
Jabeur, K., +, *JAIF*, **4**, 2 (Dec. 2009), 75–92.
Simulation
Crouse, D. F., +, *JAIF*, **10**, 1 (June 2015), 31–57.
situation assessment
Chen, G., +, *JAIF*, **2**, 1 (July 2007), 35–48.
Situation Awareness
Blasch, E., +, *JAIF*, **1**, 2 (Dec. 2006), 122–143.
Brynielsson, J., +, *JAIF*, **1**, 2 (Dec. 2006), 108–121.
situational assessment
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
situational awareness
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
smoothing
Chakravorty, R., +, *JAIF*, **1**, 1 (July 2006), 63–74.
social networks
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
sonar
Erdinc, O., +, *JAIF*, **2**, 1 (July 2007), 22–34.
space-time adaptive processing (STAP)
Kahler, B., +, *JAIF*, **6**, 2 (December 2011), 101–118.
space tracking
Belfadel, D., +, *JAIF*, **10**, 2 (December 2015), 101–112.
spline
Kurz, G., +, *JAIF*, **9**, 1 (July 2014), 13–26.

Stansfield estimator
Song, X., +, *JAIF*, **8**, 1 (July 2013), 3–14.
state constraints
Yang, C., +, *JAIF*, **3**, 1 (July 2008), 14–32.
statistical efficiency
Belfadel, D., +, *JAIF*, **9**, 2 (December 2014), 59–74.
Belfadel, D., +, *JAIF*, **10**, 2 (December 2015), 101–112.
Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.
stereo camera
Kurz, G., +, *JAIF*, **9**, 1 (July 2014), 13–26.
structure discovery
Biermann, J., +, *JAIF*, **8**, 2 (December 2013), 101–118.
structured hypotheses
Schrag, R. C., +, *JAIF*, **2**, 2 (Dec. 2007), 77–98.
subgraph matching
Sambhoos, K., +, *JAIF*, **3**, 2 (Dec. 2008), 90–106.
submodular minimization
Papageorgiou, D. J., +, *JAIF*, **6**, 2 (December 2011), 77–100.
sudden maneuvers
Blom, H. A. P., +, *JAIF*, **1**, 1 (July 2006), 15–34.
surface estimation
Kurz, G., +, *JAIF*, **9**, 1 (July 2014), 13–26.
surveillance
Tian, X., +, *JAIF*, **5**, 1 (July 2010), 3–17.

T

target attributes
Bar-Shalom, Y., +, *JAIF*, **2**, 1 (July 2007), 49–59.
Target Engageability
Benaskeur, A. R., +, *JAIF*, **2**, 2 (Dec. 2007), 99–112.
Target Existence Uncertainty
Chakravorty, R., +, *JAIF*, **1**, 1 (July 2006), 63–74.
target tracking
Aughenbaugh, J. M., +, *JAIF*, **5**, 2 (Dec. 2010), 108–127.
Bar-Shalom, Y., +, *JAIF*, **2**, 1 (July 2007), 49–59.
Benaskeur, A. R., +, *JAIF*, **2**, 2 (Dec. 2007), 99–112.
Chakravorty, R., +, *JAIF*, **1**, 1 (July 2006), 63–74.
Crouse, D. F., +, *JAIF*, **10**, 1 (June 2015), 31–57.
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
Jilkov, V. P., +, *JAIF*, **10**, 1 (June 2015), 73–00.
Zhang, S., +, *JAIF*, **6**, 1 (June 2011), 3–23.
target-death problem
Coraluppi, S., +, *JAIF*, **7**, 2 (December 2012), 153–164.
TBM, Monte Carlo simulation
Hamp, Q., +, *JAIF*, **7**, 2 (December 2012), 131–140.
templates
Sudit, M., +, *JAIF*, **2**, 1 (July 2007), 3–21.
test sequencing
Choi, S., +, *JAIF*, **8**, 2 (December 2013), 143–155.
textual representation
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
threat evaluation
Johansson, F., +, *JAIF*, **6**, 2 (December 2011), 187–199.
threat prediction
Chen, G., +, *JAIF*, **2**, 1 (July 2007), 35–48.
total least squares
Salerno, E., +, *JAIF*, **10**, 1 (June 2015), 58–72.
track association
Bar-Shalom, Y., +, *JAIF*, **3**, 2 (Dec. 2008), 107–117.
Osborne, R. W., III, +, *JAIF*, **7**, 1 (June 2012), 3–15.
track detection
Romeo, K., +, *JAIF*, **10**, 2 (December 2015), 113–124.
track fusion
Areta, J., +, *JAIF*, **4**, 1 (July 2009), 40–51.
Bar-Shalom, Y., +, *JAIF*, **3**, 2 (Dec. 2008), 107–117.
Yang, C., +, *JAIF*, **3**, 1 (July 2008), 14–32.
track initialization
Osborne, III, R. W., +, *JAIF*, **10**, 2 (December 2015), 199–210.
track-before-detect
Davey, S. J., *JAIF*, **6**, 2 (December 2011), 167–186.
Track-to-Track
Tian, X., +, *JAIF*, **4**, 2 (Dec. 2009), 146–164.
Track-to-track association
Papageorgiou, D. J., +, *JAIF*, **6**, 2 (December 2011), 77–100.
tracking
An, W., +, *JAIF*, **3**, 1 (July 2008), 33–49.
Areta, J., +, *JAIF*, **2**, 2 (Dec. 2007), 113–127.
Aughenbaugh, J. M., +, *JAIF*, **5**, 2 (Dec. 2010), 108–127.
Bar-Shalom, Y., +, *JAIF*, **1**, 1 (July 2006), 3–14.
Bar-Shalom, Y., +, *JAIF*, **2**, 1 (July 2007), 49–59.
Bar-Shalom, Y., +, *JAIF*, **3**, 2 (Dec. 2008), 107–117.
Blom, H. A. P., +, *JAIF*, **1**, 1 (July 2006), 15–34.
Chakravorty, R., +, *JAIF*, **1**, 1 (July 2006), 63–74.
Crouse, D. F., +, *JAIF*, **4**, 2 (Dec. 2009), 93–116.
Erdinc, O., +, *JAIF*, **2**, 1 (July 2007), 22–34.
Jilkov, V. P., +, *JAIF*, **4**, 1 (July 2009), 27–39.
Kurz, G., +, *JAIF*, **9**, 1 (July 2014), 13–26.
Ravindra, V. C., +, *JAIF*, **5**, 2 (Dec. 2010), 88–107.
Rodningsby, A., +, *JAIF*, **4**, 2 (Dec. 2009), 117–145.
Song, X., +, *JAIF*, **7**, 1 (June 2012), 16–27.
Tian, X., +, *JAIF*, **4**, 2 (Dec. 2009), 146–164.
Tian, X., +, *JAIF*, **5**, 2 (Dec. 2010), 128–138.
tracking and classification
Carvalho, R. N., +, *JAIF*, **7**, 2 (December 2012), 141–152.
Traffic Surveillance
Junghans, M., +, *JAIF*, **3**, 1 (July 2008), 50–62.
trafficability
George, J., +, *JAIF*, **6**, 1 (June 2011), 39–56.
Transferable Belief Model
Rogova, G. L., +, *JAIF*, **3**, 2 (Dec. 2008), 118–128.

U

UAV search
Jilkov, V. P., +, *JAIF*, **4**, 1 (July 2009), 27–39.
UAV
Tian, X., +, *JAIF*, **5**, 1 (July 2010), 3–17.
uncertainty
Hamp, Q., +, *JAIF*, **7**, 2 (December 2012), 131–140.
Uncertainty Quantification
Terejanu, G., +, *JAIF*, **5**, 2 (Dec. 2010), 73–87.
unscented transform
Osborne, R. W., III, +, *JAIF*, **7**, 1 (June 2012), 3–15.

V

visual representation
Kaliciak, L., +, *JAIF*, **10**, 2 (December 2015), 183–198.
VLO
Romeo, K., +, *JAIF*, **10**, 2 (December 2015), 113–124.
volcanic eruptions
Rogova, G. L., +, *JAIF*, **3**, 2 (Dec. 2008), 118–128.
voting algorithms
Hoseinnezhad, R., +, *JAIF*, **1**, 1 (July 2006), 52–62.

W

wake
Rodningsby, A., +, *JAIF*, **4**, 2 (Dec. 2009), 117–145.
weapon allocation
Johansson, F., +, *JAIF*, **6**, 2 (December 2011), 187–199.

INTERNATIONAL SOCIETY OF INFORMATION FUSION

ISIF Website: http://www.isif.org

2015 BOARD OF DIRECTORS*

2013–2015	2014–2016	2015–2017
Jean Dezert	Sten F. Andler	Joachim Biermann
Gee-Wah Ng	Murat Efe	Frederik Gustafsson
Anne-Laure Jousselme	Lyudmila Mihaylova	Jesús García

*Board of Directors are elected by the members of ISIF for a three year term.

PAST PRESIDENTS

Darin Dunham, 2015	Elisa Shahbazian, 2009	Xiao-Rong Li, 2003
Darin Dunham, 2014	Darko Musicki, 2008	Yaakov Bar-Shalom, 2002
Wolfgang Koch, 2013	Erik Blasch, 2007	Pramod Varshney, 2001
Roy Streit, 2012	Pierre Valin, 2006	Yaakov Bar-Shalom, 2000
Joachim Biermann, 2011	W. Dale Blair, 2005	Jim Llinas, 1999
Stefano Coraluppi, 2010	Chee Chong, 2004	Jim Llinas, 1998

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 11 Number 1 June 2016 ISSN 1557-6418