Nonlinear Filtering with a Polynomial Series of Gaussian Random Variables

Simone Servadio *The University of Texas at Austin*, Austin, Texas, simo_serva@utexas.edu
Renato Zanetti *The University of Texas at Austin*, Austin, Texas, renato@utexas.edu
Brandon A. Jones *The University of Texas at Austin*, Austin, Texas, brandon.jones@utexas.edu

Abstract

Filters relying on the Gaussian approximation typically incorporate the measurement linearly, i.e., the value of the measurement is pre-multiplied by a matrix-valued gain in the state update. Nonlinear filters that relax the Gaussian assumption, on the other hand, typically approximate the distribution of the state with a finite sum of point masses or Gaussian distributions. In this work, the distribution of the state is approximated by a polynomial transformation of a Gaussian distribution, allowing for all moments, central and raw, to be rapidly computed in closed form. Knowledge of the higher-order moments is then employed to perform a polynomial measurement update, i.e., the value of the measurement enters the update function as a polynomial of arbitrary order. A filter employing a Gaussian approximation with linear update is, therefore, a special case of the proposed algorithm when both the order of the series and the order of the update are set to one: it reduces to the extended Kalman filter. At the cost of more computations, the new methodology guarantees performance better than the linear/Gaussian approach for nonlinear systems. This work employs monomial basis functions and Taylor series, developed in the Differential Algebra framework, but it is readily extendable to an orthogonal polynomial basis.

Index Terms

Differential Algebra, Nonlinear filtering, Polynomial update

I. Introduction

The conditional mean is the optimal estimate to the sequential stochastic Minimum Mean Square Error (MMSE) estimation problem. It is derived recursively thought Bayes' rule. When the system is linear and Gaussian, the Kalman filter [1] [2] exactly represents the conditional distribution. However, when the dynamics of the system are nonlinear, the conditional probability density function (PDF) is generally a non-Gaussian distribution, and the optimal MMSE estimate is not typically obtainable in closed form.

Consequentially, many algorithms have been developed that approximate the optimal MMSE solution in the presence of nonlinearities. A simple approach is to perform a linearization of the measurement and dynamics functions around the current estimate, and apply the Kalman filter equations as if the system were linear; the so-called Extended Kalman Filter (EKF) [3]. However, simple linearization fails to achieve convergence for highly nonlinear systems [4].

The Unscented Kalman Filter (UKF) [5], [6] usually achieves better results and is more robust to nonlinearities than the EKF. Using deterministically placed sigma points, the distribution of the state is approximated by applying the actual nonlinear dynamics and nonlinear measurement function to estimate the mean and covariance matrix via a weighted average.

Several methods to better incorporate the nonlinear dynamics during the time propagation phase of filtering algorithms exist. Park and Sheeres [7] employ State Transition Tensors (STT) to propagate mean and high order central moments. Majji et al. [8] expand Park and Sheeres work to include the high-order moments in the update. Valli et al. [9] reproduced these results in the differential algebra (DA) framework.

Filters based on differential algebra approximate functions with their Taylor expansion series up to a fixed truncation order. The relation between the accuracy of the DA estimators and the truncation order has been studied by Rasotto et al. [10], [11]. The studies concluded that increasing the order of the expansion is beneficial up to a certain threshold, after which the increase of computational effort no longer warrants the small increment in accuracy.

All above-mentioned filters, including the EKF and UKF, are linear estimators, i.e., the estimate is a linear function of the measurements. The conditional mean, which is the optimal MMSE solution, is generally some unknown nonlinear function of the measurement whose exact form usually cannot be calculated. A linear estimator, even when accounting for the nonlinearities of the measurement function, is typically outperformed by nonlinear estimators such as the Gaussian Sum Filter (GSF) [12] [13] or particle filters [14].

In the GSF, the optimal nonlinear update is approximated by representing the state PDF with multiple normal distributions; the Gaussian Mixture Models (GMM). As long as each component of the GMM has a covariance “small” enough, linearization of the measurement and dynamic functions can accurately represent the uncertainty in the component’s support, and linear update equations (EKF or UKF) can be used for each component. The total update is a weighted combination of the linear updates, where each weight is obtained from the likelihood of the corresponding Gaussian component.

De Santis et al. [15] developed a quadratic update by augmenting the state of the system with its square. The estimator in [15] also augments the measurement with its square, by doing so they are able to rewrite the quadratic update with a linear update structure. A polynomial approximation of the optimal

nonlinear update also exists [16].

Servadio and Zanetti [17] implemented a quadratic update (extendable to a polynomial update of any order) based on Taylor series expansions. By carrying central moments up to a desired order, the polynomial high-order coefficients are evaluated to minimize the mean square estimation error. The computational demand of calculating higher order central moments grows quickly with the order of the series, the size of the state vector, and the order of the polynomial update.

In this work all non-Gaussian distributions are approximated as a polynomial transformation of Gaussian random variables. In doing so, all high order central moments are easily and efficiently calculated in close form with Isserlis formula [18]. As a consequence, polynomial updates can be performed more efficiently than in prior works.

Preliminary results for this work were presented in [19]. This paper expands on that work by including a new polynomial reduction technique, the results section is expanded to reflect this new case and includes a comparison of accuracy between the two reduction methodologies. The two formulations of the polynomial reduction offer different accuracy and computational complexity. The least squares approach first proposed in Ref. [19] better approximates the shape of the distribution, and hence it is more accurate than the new method that relies on the Gaussian approximation. The newly introduced Gaussian approximation used for polynomial reduction is computationally simpler and faster than solving a least squares problem. The draw back is that it looses the shape of the distribution as it approximates the PDF as Gaussian and preserves only the mean and covariance. This approximation results in a more conservative filter, i.e., loss of performance. Hence, this work expands [19] by proposing a similar but simpler nonlinear filter, the trade between the two is overall estimation accuracy vs. computational cost, the decreased accuracy of the new simpler algorithm is made evident in a numerical example. Finally, this paper includes an additional scalar estimation example not shown in [19] to highlight the benefits of polynomial estimation when compared to a linear approach.

The remainder of this paper is structured in the following way: Section II presents the mathematics and fundamentals for the newly proposed polynomial estimator; Section III describes the filtering algorithm and its implementation in the DA framework; then, Section IV applies the new filtering techniques both to a demonstrative scalar problem and to the more challenging simulation of a Lorenz96 application [20], [21]; lastly, Section V draws conclusions.

II. Polynomial Estimator

In our previous work, [17], a nonlinear update using polynomial residuals was presented. While the polynomial approach can be extended to arbitrary order, the quadratic update is shown here. Let $\mathbf{x}$ be

the state of the system we wish to estimate, modeled as a random vector, and $\mathbf{y}$ another, related, random vector that is sampleable. We will use the known outcome of $\mathbf{y}$ to estimate the unknown outcome of $\mathbf{x}$. Let $\mathbf{g}(\mathbf{y})$ be a family of quadratic estimators of $\mathbf{x}$ from $\mathbf{y}$, defined by constants $\mathbf{a}$, which is a vector, and $\mathbf{K}_1, \mathbf{K}_2$ which are matrices of appropriate dimensions.

$$\mathbf{g}(\mathbf{y}) = \mathbf{a} + \mathbf{K}_1\mathbf{y} + \mathbf{K}_2\mathbf{y}^{[2]} \tag{1}$$

where $\mathbf{y}^{[2]} = \mathbf{y} \otimes \mathbf{y}$ and the symbol $\otimes$ indicates the Kronecker product between two vectors, but with the redundant components of $\mathbf{y} \otimes \mathbf{y}$ eliminated, e.g., only one y_1y_2 or y_2y_1 is present in $\mathbf{y}^{[2]}$. It is often convenient to work with deviation vectors of the variables from their mean instead of the vectors themselves. Therefore, let us define the quantity $d\mathbf{y} = \mathbf{y} - \mathbb{E}\{\mathbf{y}\}$, which indicates the measurement residual, and similarly, the state deviation as $d\mathbf{x} = \mathbf{x} - \mathbb{E}\{\mathbf{x}\}$. Ref. [17] defines the quadratic estimator in terms of $d\mathbf{y}^{[2]} = d\mathbf{y} \otimes d\mathbf{y}$, but, in this work, a different quantity is used:

$$d\mathbf{y}^{\{2\}} = \mathbf{y} \otimes \mathbf{y} - \mathbb{E}\{\mathbf{y} \otimes \mathbf{y}\} = \mathbf{y}^{[2]} - \mathbb{E}\left\{\mathbf{y}^{[2]}\right\}$$

The two formulations are equivalent, but the latter is convenient because it has zero mean. Thus, we work with the deviation of the square instead of the square of the deviation. Without any loss of generality, we can redefine $\mathbf{a}$, $\mathbf{K}_1$, and $\mathbf{K}_2$ by adding and subtracting constants in order to obtain a different, but equivalent, family of quadratic estimators

$$\mathbf{g}(\mathbf{y}) = \mathbf{a} + \mathbb{E}\{\mathbf{x}\} + \mathbf{K}_1(\mathbf{y} - \mathbb{E}\{\mathbf{y}\}) + \mathbf{K}_2\left(\mathbf{y}^{[2]} - \mathbb{E}\left\{\mathbf{y}^{[2]}\right\}\right) \tag{2}$$

It is known that the optimal values of the estimator's coefficients (denoted with an asterisk) satisfy the orthogonality principle

$$\begin{aligned}\mathbb{E}\Bigg\{\left(d\mathbf{x} - \mathbf{a}^* - \mathbf{K}_1^* d\mathbf{y} - \mathbf{K}_2^* d\mathbf{y}^{\{2\}}\right)\cdot \\ \cdot\left(\mathbf{a} + \mathbb{E}\{\mathbf{x}\} + \mathbf{K}_1 d\mathbf{y} + \mathbf{K}_2 d\mathbf{y}^{\{2\}}\right)^{\mathrm{T}}\Bigg\} = \mathbf{O}\end{aligned} \tag{3}$$

which is valid $\forall \mathbf{a}, \mathbf{K}_1, \mathbf{K}_2$. Therefore the optimal coefficients can be found solving the linear system

$$\mathbf{a}^* = \mathbf{0} \tag{4}$$

$$\mathbf{K}_1^*\mathbf{P}_{d\mathbf{y}d\mathbf{y}} + \mathbf{K}_2^*\mathbf{P}_{d\mathbf{y}^{\{2\}}d\mathbf{y}} = \mathbf{P}_{d\mathbf{x}d\mathbf{y}} \tag{5}$$

$$\mathbf{K}_1^*\mathbf{P}_{d\mathbf{y}d\mathbf{y}^{\{2\}}} + \mathbf{K}_2^*\mathbf{P}_{d\mathbf{y}^{\{2\}}d\mathbf{y}^{\{2\}}} = \mathbf{P}_{d\mathbf{x}d\mathbf{y}^{\{2\}}} \tag{6}$$

where the following notation is used:

$$\mathbf{P}_{\mathbf{xy}} = \mathbb{E}\left\{(\mathbf{x} - \mathbb{E}\{\mathbf{x}\})\ (\mathbf{y} - \mathbb{E}\{\mathbf{y}\})^{\mathrm{T}}\right\} \tag{7}$$

Since $\mathbf{y}^{[2]}$ and $d\mathbf{y}^{\{2\}}$ only differ by a constant, they share the same covariance matrix.

The optimal solution is given by

$$\mathbf{a}^* = \mathbf{0} \tag{8}$$

$$\begin{bmatrix}\mathbf{K}_1^* & \mathbf{K}_2^*\end{bmatrix} = \begin{bmatrix}\mathbf{P}_{d\mathbf{x}d\mathbf{y}}^{\mathrm{T}} \\ \mathbf{P}_{d\mathbf{x}d\mathbf{y}^{\{2\}}}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \begin{bmatrix}\mathbf{P}_{d\mathbf{y}d\mathbf{y}} & \mathbf{P}_{d\mathbf{y}d\mathbf{y}^{\{2\}}} \\ \mathbf{P}_{d\mathbf{y}^{\{2\}}d\mathbf{y}} & \mathbf{P}_{d\mathbf{y}^{\{2\}}d\mathbf{y}^{\{2\}}}\end{bmatrix}^{-1} \tag{9}$$

The optimal estimator is

$$\hat{\mathbf{x}} = \mathbb{E}\{\mathbf{x}\} + \mathbf{K}_1^* d\mathbf{y} + \mathbf{K}_2^*\ d\mathbf{y}^{\{2\}} \tag{10}$$

and the posterior estimation error is

$$\epsilon = \mathbf{x} - \hat{\mathbf{x}} = \mathbf{x} - \mathbf{g}(\mathbf{y}) = \tag{11}$$

$$= \mathbf{x} - \left(\mathbb{E}\{\mathbf{x}\} + \mathbf{K}_1^* d\mathbf{y} + \mathbf{K}_2^* d\mathbf{y}^{\{2\}}\right) \tag{12}$$

$$= d\mathbf{x} - \mathbf{K}_1^* d\mathbf{y} - \mathbf{K}_2^* d\mathbf{y}^{\{2\}} \tag{13}$$

A. Generalization to Higher Orders

It is possible to generalize the results of the previous section to higher orders of the polynomial update. Let us expand Equation (1) to an arbitrary order

$$\mathbf{g}(\mathbf{y}) = \mathbf{a} + \mathbf{K}_1\mathbf{y} + \mathbf{K}_2\mathbf{y}^{[2]} + \mathbf{K}_3\mathbf{y}^{[3]} + \mathbf{K}_4\mathbf{y}^{[4]} + \dots \tag{14}$$

where each $\mathbf{K}_i$ is a matrix of appropriate dimensions and each $\mathbf{y}^{[i]}$ is calculated using the Kronecker product with the repeated elements deleted. The measurements and its powers can be stacked, defining the augmented measurement vector

$$\mathcal{Y} = \begin{bmatrix}\mathbf{y}^{\mathrm{T}} & \mathbf{y}^{[2]T} & \mathbf{y}^{[3]T} & \dots\end{bmatrix}^{\mathrm{T}} \tag{15}$$

Once again, the estimator family is redefined in order to work with deviations. Since deviations have zero mean by construction, the identities $\mathbf{P}_{\mathbf{y}^{[i]}\mathbf{y}^{[j]}} = \mathbf{P}_{d\mathbf{y}^{\{i\}}d\mathbf{y}^{\{j\}}}$ and $\mathbf{P}_{\mathbf{x}\mathbf{y}^{[j]}} = \mathbf{P}_{d\mathbf{x}d\mathbf{y}^{\{j\}}}$ are valid $\forall i, j \in \mathbb{N}_0$.

Therefore, it is possible to evaluate the augmented measurement covariance matrix as

$$\mathbf{P}_{\mathcal{Y}\mathcal{Y}} = \begin{bmatrix} \mathbf{P}_{\mathbf{y}\mathbf{y}} & \mathbf{P}_{\mathbf{y}\mathbf{y}^{[2]}} & \mathbf{P}_{\mathbf{y}\mathbf{y}^{[3]}} & \dots \\ \mathbf{P}_{\mathbf{y}^{[2]}\mathbf{y}} & \mathbf{P}_{\mathbf{y}^{[2]}\mathbf{y}^{[2]}} & \mathbf{P}_{\mathbf{y}^{[2]}\mathbf{y}^{[2]}} & \dots \\ \mathbf{P}_{\mathbf{y}^{[3]}\mathbf{y}} & \mathbf{P}_{\mathbf{y}^{[3]}\mathbf{y}^{[2]}} & \mathbf{P}_{\mathbf{y}^{[3]}\mathbf{y}^{[3]}} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{16}$$

and the augmented state-measurement cross-covariance matrix as

$$\mathbf{P}_{\mathbf{x}\mathcal{Y}} = \begin{bmatrix} \mathbf{P}_{\mathbf{x}\mathbf{y}} & \mathbf{P}_{\mathbf{x}\mathbf{y}^{[2]}} & \mathbf{P}_{\mathbf{x}\mathbf{y}^{[3]}} & \dots \end{bmatrix} \tag{17}$$

The deviations can be stacked as well, creating an augmented deviation vector.

$$d\mathcal{Y} = \begin{bmatrix} d\mathbf{y}^{\mathrm{T}} & d\mathbf{y}^{\{2\}\mathrm{T}} & d\mathbf{y}^{\{3\}\mathrm{T}} & \dots \end{bmatrix}^{\mathrm{T}} \tag{18}$$

The optimal polynomial update estimator becomes

$$\hat{\mathbf{x}} = \mathbb{E}\left\{\mathbf{x}\right\} + \mathbf{P}_{\mathbf{x}\mathcal{Y}}\mathbf{P}_{\mathcal{Y}\mathcal{Y}}^{-1} d\mathcal{Y} \tag{19}$$

where the product $\mathbf{P}_{\mathbf{x}\mathcal{Y}}\mathbf{P}_{\mathcal{Y}\mathcal{Y}}^{-1}$ is similar to the Kalman gain but realized with knowledge of the central moments of the distribution up to a selected order, improving accuracy.

III. The High-Order Polynomial Update Filter - HOPUF-ℓ-c

The new filtering technique has been developed in the Differential Algebra (DA) framework, using an algebra based on Taylor series expansions. The Taylor series coefficients are obtained up to an arbitrary and specified order using a library of know Taylor series expansions of elementary functions using the Differential Algebra Core Engine (DACE2.0) software [22], [23]. DACE2.0 does not compute derivatives numerically, e. g. finite differences, but expresses any function as an array of coefficients and exponents from a hard-coded library. Please refer to [9], [24]–[26] for a more detailed review of DA, and to [10], [27] for details on the DACE2.0 software implementation.

The High-Order Polynomial Update Filter (HOPUF-ℓ-c) algorithm is composed of three main parts: the prediction, the polynomial update, and the polynomial reduction. The two integers ℓ and c in HOPUF-ℓ-c refer to the order of the polynomial update and of the Taylor series expansion, respectively. Hence, HOPUF-1-1 is the extended Kalman filter, HOPUF-1-2 is the Gaussian Second Order Filter, and HOPUF-2-3 is a quadratic estimator with nonlinear functions approximated by their third order Taylor series expansion.

A. Prediction

Consider the following equations of motion and measurement equations describing a generic dynamic system:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + \mathbf{v}_k \tag{20}$$

$$\mathbf{y}_{k+1} = \mathbf{h}(\mathbf{x}_{k+1}) + \mathbf{w}_{k+1} \tag{21}$$

where $\mathbf{f}(\cdot)$ is the dynamics function, $\mathbf{x}_k$ is the n-dimensional state of the system at time-step k, $\mathbf{y}_{k+1}$ is the m-dimensional measurement vector at time-step $k+1$, and $\mathbf{h}(\cdot)$ is the measurement function. The process noise and the measurement noise are assumed to be zero mean and uncorrelated, i.e., $\mathbb{E}\left\{\mathbf{v}_i \mathbf{w}_j^T\right\} = 0 \ \forall i, j$, with the autocovariance functions $\mathbb{E}\left\{\mathbf{v}_i \mathbf{v}_j^T\right\} = \mathbf{Q}_i \delta_{ij}$, and $\mathbb{E}\left\{\mathbf{w}_i \mathbf{w}_j^T\right\} = \mathbf{R}_i \delta_{ij}$ for all discrete time indexes i and j. It is assumed that the initial condition $\mathbf{x}_0$ and the noises are Gaussian, however, the nonlinear functions will make the distribution of $\mathbf{x}_k$ non-Gaussian for all $k > 0$.

DA expresses all functions as their Taylor series expansion up to an arbitrary order. The state is initialized in the DA framework with a first order polynomial.

$$\mathbf{x}_0 = \mathbf{x}_0(\delta\mathbf{x}_0) = \hat{\mathbf{x}}_\mathbf{0} + \mathbf{S}_0 \delta\mathbf{x}_0 \tag{22}$$

where $\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_\mathbf{0}, \mathbf{P}_0)$, $\mathbf{S}_0 \mathbf{S}_0^T = \mathbf{P}_0$, and the DA variable $\delta\mathbf{x}_0 = \mathbf{x}_0 - \hat{\mathbf{x}}_\mathbf{0}$ is therefore interpreted as Gaussian with zero mean and identity covariance matrix.

The propagation function is applied directly to the state polynomials $\mathbf{x}_k$, thus, the ith component of the predicted state vector is

$$\begin{aligned} x_{i,k+1}^- &= x_{i,k+1}^-(\delta\mathbf{x}_k) = f_i(\mathbf{x}_k(\delta\mathbf{x}_k)) = \\ &= f_i(\hat{\mathbf{x}}_k) + \sum_{r=1}^{c} \frac{1}{r!} \sum \frac{\partial^r f_i(\mathbf{x}_k)}{\partial x_1^{\gamma_1} \dots \partial x_n^{\gamma_r}} \delta x_{1,k}^{\gamma_1} \dots \delta x_{n,k}^{\gamma_r} \end{aligned} \tag{23}$$

where the integer c indicates the user-defined order of the Taylor expansion (the same numerical value specified in HOPUF-ℓ-c); the second summation is over all permutations of $\gamma_i \in \{1, \dots, n\}$ with $i \in \{1, \dots, r\}$. Therefore $\mathbf{x}_{k+1}^-(\delta\mathbf{x}_k)$ is a vector of polynomial functions that map the deviations $(\delta\mathbf{x}_k)$ into the distribution of the state at time $k+1$. The final step of the propagation is to add the process noise contribution. The process noise is introduced as a DA variable $\delta\mathbf{v}_k$, which once again is interpreted as a standard normal random vector

$$\mathbf{x}_{k+1}^-(\delta\mathbf{x}_k, \delta\mathbf{v}_k) \coloneqq \mathbf{x}_{k+1}^-(\delta\mathbf{x}_k) + \mathbf{T}_k \delta\mathbf{v}_k \tag{24}$$

where $\mathbf{v}_k = \mathbf{T}_k \delta\mathbf{v}_k$ and $\mathbf{T}_k \mathbf{T}_k^{\mathrm{T}} = \mathbf{Q}_k$.

The predicted measurements are evaluated similarly and the jth component is given by:

$$\begin{aligned} y_{j,k+1} &= y_{j,k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k) = h_j(\mathbf{f}(\mathbf{x}_{k+1}^-)) = \\ &= h_j(\mathbf{f}(\hat{\mathbf{x}}_{k+1})) + \sum_{r=1}^{c} \frac{1}{r!} \sum \frac{\partial^r h_j(\mathbf{x}_{k+1}^-)}{\partial x_1^{\gamma_1} \dots \partial x_n^{\gamma_r} \partial v_1^{\gamma_1} \dots \partial v_n^{\gamma_r}} \cdot \\ &\cdot \delta x_{1,k}^{\gamma_1} \dots \delta x_{n,k}^{\gamma_r} \delta v_{1,k}^{\gamma_1} \dots \delta v_{n,k}^{\gamma_r} \end{aligned} \tag{25}$$

where, again, the second summation is over all permutations of $\gamma_i \in \{1, \dots, n\}$ with $i \in \{1, \dots, r\}$. Note that the Taylor series expansion is now in terms of both $\delta\mathbf{x}_k$ and $\delta\mathbf{v}_k$. The measurement noise is also introduced as a DA variable $\delta\mathbf{w}_{k+1}$, which is interpreted as a standard normal random vector

$$\mathbf{y}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) \coloneqq \mathbf{y}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k) + \mathbf{U}_k \delta\mathbf{w}_{k+1} \tag{26}$$

where $\mathbf{w}_k = \mathbf{U}_k \delta\mathbf{w}_k$ and $\mathbf{U}_k \mathbf{U}_k^T = \mathbf{R}_k$.

The total number of DA variables is $2n + m$: n are used to map the state behavior ($\delta\mathbf{x}_k$), n map the process noise ($\delta\mathbf{v}_k$), and the remaining m map the measurement noise ($\delta\mathbf{w}_{k+1}$). All quantities of interest are represented by polynomial functions of standard normal random vectors. Therefore all expectations can be easily computed using Isserlis' formula to calculate central moments of standard normals [18].

B. Quadratic Update

Once the predicted state and measurement polynomials have been calculated, the next step in performing the quadratic update is evaluating the augmented Kalman gain with Equation (9). We start by calculating

$$\mathbf{y}_{k+1}^{[2]} = \mathbf{y}_{k+1} \otimes \mathbf{y}_{k+1} \tag{27}$$

where, once again, all the redundant components are eliminated.

The mean of each component of the predicted state is

$$\hat{x}_{i,k+1}^- = \mathbb{E}\left\{ x_{i,k+1}^- \right\} \tag{28}$$

$$= f_i(\hat{\mathbf{x}}_k) + \sum_{r=1}^{c} \frac{1}{r!} \sum \frac{\partial^r f_i(\mathbf{x}_k)}{\partial x_1^{\gamma_1} \dots \partial x_n^{\gamma_r}} \mathbb{E}\left\{ \delta x_{1,k}^{\gamma_1} \dots \delta x_{n,k}^{\gamma_r} \right\} \tag{29}$$

where the expectations are applied directly on the monomials of the series. Since the deviations have a Gaussian distribution with zero mean and identity covariance, the expected value operator just substitutes the relative Isserlis' moment in for to each monomial, according to Table I. As an illustrative example:

TABLE I
ISSERLIS' MOMENTS OF GAUSSIAN $\mathcal{N}(0,1)$

exponent	0	1	2	3	4	5	6	7	8	...
coefficient	1	0	1	0	3	0	15	0	105	...

$\mathbb{E}\left\{a_i \delta x_1^2 \delta x_2^6 \delta x_4^4\right\} = 45 a_i$. Similar to Equation (28), the predicted means of the measurements are evaluated as:

$$\hat{\mathbf{y}}_{k+1} = \mathbb{E}\left\{\mathbf{y}_{k+1}\right\} \tag{30}$$

$$\hat{\mathbf{y}}_{k+1}^{[2]} = \mathbb{E}\left\{\mathbf{y}_{k+1}^{[2]}\right\} \tag{31}$$

The augmented measurement covariance matrix is calculated blockwise:

$$\mathbf{P}_{\mathcal{Y}\mathcal{Y}} = \begin{bmatrix} \mathbf{P}_{\mathbf{yy}} & \mathbf{P}_{\mathbf{yy}^{[2]}} \\ \mathbf{P}_{\mathbf{y}^{[2]}\mathbf{y}} & \mathbf{P}_{\mathbf{y}^{[2]}\mathbf{y}^{[2]}} \end{bmatrix} \tag{32}$$

where, having removed the redundant components from the square of the measurements, the matrix is guaranteed to avoid any singularities. The matrix is symmetric and the three different blocks are calculated as:

$$\mathbf{P}_{\mathbf{yy}} = \mathbb{E}\left\{(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1})(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1})^{\mathrm{T}}\right\} \tag{33}$$

$$\mathbf{P}_{\mathbf{yy}^{[2]}} = \mathbb{E}\left\{(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1})(\mathbf{y}_{k+1}^{[2]} - \hat{\mathbf{y}}_{k+1}^{[2]})^{\mathrm{T}}\right\} \tag{34}$$

$$\mathbf{P}_{\mathbf{y}^{[2]}\mathbf{y}^{[2]}} = \mathbb{E}\left\{(\mathbf{y}_{k+1}^{[2]} - \hat{\mathbf{y}}_{k+1}^{[2]})(\mathbf{y}_{k+1}^{[2]} - \hat{\mathbf{y}}_{k+1}^{[2]})^{\mathrm{T}}\right\} \tag{35}$$

The augmented cross-covariance matrix between the state and measurement is evaluated blockwise as well:

$$\mathbf{P}_{\mathbf{x}\mathcal{Y}} = \begin{bmatrix} \mathbf{P}_{\mathbf{xy}} & \mathbf{P}_{\mathbf{xy}^{[2]}} \end{bmatrix} \tag{36}$$

where each block is calculated in a similar way:

$$\mathbf{P}_{\mathbf{xy}} = \mathbb{E}\left\{(\mathbf{x}_{k+1}^{-} - \hat{\mathbf{x}}_{k+1}^{-})(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1})^{\mathrm{T}}\right\} \tag{37}$$

$$\mathbf{P}_{\mathbf{xy}^{[2]}} = \mathbb{E}\left\{(\mathbf{x}_{k+1}^{-} - \hat{\mathbf{x}}_{k+1}^{-})(\mathbf{y}_{k+1}^{[2]} - \hat{\mathbf{y}}_{k+1}^{[2]})^{\mathrm{T}}\right\} \tag{38}$$

From these quantities it is now possible to use Equation (9) to calculate the Kalman gain associated with the augmented system:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_1 & \mathbf{K}_2 \end{bmatrix} = \mathbf{P}_{\mathbf{x}\mathcal{Y}} \mathbf{P}_{\mathcal{Y}\mathcal{Y}}^{-1} \tag{39}$$

Denote $\tilde{\mathbf{y}}_k$ as the outcome of random vector $\mathbf{y}_k$, i.e., the numerical value of the measurement from the sensors, and its square as

$$\tilde{\mathbf{y}}_k^{[2]} = \tilde{\mathbf{y}}_k \otimes \tilde{\mathbf{y}}_k \tag{40}$$

The updated distribution of the state and the posterior estimate are given by

$$\begin{aligned} &\mathbf{x}_{k+1}^{+}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) \\ &\quad = \mathbf{x}_{k+1}^{-}(\delta\mathbf{x}_k, \delta\mathbf{v}_k) + \mathbf{K} \begin{bmatrix} \tilde{\mathbf{y}} - \mathbf{y}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) \\ \tilde{\mathbf{y}}^{[2]} - \mathbf{y}_{k+1}^{[2]}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) \end{bmatrix} \end{aligned} \tag{41}$$

$$\hat{\mathbf{x}}_{k+1}^{+} = \mathbb{E}\left\{\mathbf{x}_{k+1}^{+}\right\} = \hat{\mathbf{x}}_{k+1}^{-} + \mathbf{K} \begin{bmatrix} \tilde{\mathbf{y}}_{k+1} - \hat{\mathbf{y}}_{k+1} \\ \tilde{\mathbf{y}}_{k+1}^{[2]} - \hat{\mathbf{y}}_{k+1}^{[2]} \end{bmatrix} \tag{42}$$

Note that the updated polynomials are a function of the state deviations, the process noise and the measurement noise. Moreover, since the order of $\mathbf{x}_{k+1}^{-}(\delta\mathbf{x}_k, \delta\mathbf{v}_k)$ is c, then, because of the quadratic update, $\mathbf{x}_{k+1}^{+}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1})$ is a polynomial of order $2c$. The higher the polynomial order, the higher the number of moments to be calculated with Table I. The covariance matrix is obtained as

$$\mathbf{P}_{\mathbf{xx},k+1} = \mathbb{E}\left\{(\mathbf{x}_{k+1}^{+} - \hat{\mathbf{x}}_{k+1}^{+})(\mathbf{x}_{k+1}^{+} - \hat{\mathbf{x}}_{k+1}^{+})^{\mathrm{T}}\right\} \tag{43}$$

Thus, starting form a polynomial of order c at the beginning of each filter's iteration, equation (43) employs a polynomial of order $4c$ in order to calculate the state covariance matrix. Equations (42) and (43) are outputs calculated for downstream users; the recursive algorithm does not employ the actual value of the state estimate and its uncertainties to start the next iteration; the coefficients of the Taylor series representation of the updated polynomial $\mathbf{x}_{k+1}^{+}$ are stored and contain all the information needed for the recursion.

C. Polynomial Reduction

The updated state $\mathbf{x}_{k+1}^{+}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1})$ is a $2c$th order polynomial in $2n+m$ variables and it describes the (typically non-Gaussian) posterior probability density function of the state. Without taking any further action and starting the next filter iteration from $\mathbf{x}_{k+1}^{+}$ to calculate $\mathbf{x}_{k+2}^{+}$, the order of the series will double to $4c$ and the number of independent variables will increase by two: the process noise associated with the subsequent propagation and the measurement noise associated with the next measurement.

To keep the recursive algorithm tractable, it is therefore desirable to: *i.* reduce the polynomial order back to c, and *ii.* reduce the number of input variables back to n. In this section we introduce two new methodologies to achieve these two objectives.

A possible solution is to approximate the distribution as Gaussian, which simplifies the problem and is consistent with the first filtering step taken after initialization. The updated polynomial is rewritten using $\hat{\mathbf{x}}_{k+1}^{+}$ and $\mathbf{P}_{\mathbf{xx},k+1}$ in a manner similar to initialization, Eq. (22):

$$\mathbf{x}_{k+1} = \hat{\mathbf{x}}_{k+1}^{+} + \mathbf{S}_{\mathbf{xx},k+1}\delta\mathbf{x}_{k+1} \tag{44}$$

$$\mathbf{S}_{\mathbf{xx},k+1}\mathbf{S}_{\mathbf{xx},k+1}^{\mathrm{T}} = \mathbf{P}_{\mathbf{xx},k+1} \tag{45}$$

where $\delta\mathbf{x}_{k+1}$ is a new (DA) deviation interpreted as Gaussian with zero mean and identity covariance matrix. The polynomial expansion of the state is redefined after each measurement update using the state estimate and the corresponding covariance matrix. The result is a fast and computationally efficient filter. The drawback is that all the information about the non-Gaussianity of the distribution is lost. In the remainder of the paper performing the Gaussian approximation after the update is referred to as HOPUFG: this computationally faster variant can be used when the nonlinearities of the dynamics or measurements are not too marked, and a Gaussian filter is a sensible choice.

The second approach proposed preserves the non-Gaussian nature of the posterior distribution. The idea is to exactly match the zeroth and first-order components of the series representation of $\mathbf{x}_{k+1}^{+}$ and to approximate the higher-order elements with least squares.

The posterior is a nonlinear (polynomial) function of three Gaussian distributions, and the goal is to accurately approximate it as a polynomial function of a single Gaussian. The linear part of the Taylor series (zeroth and first order) is represented exactly as a single Gaussian, while higher than first-order contributions are approximated via least-squares.

We start by dividing the polynomials into their linear part (constant plus first order) and the remaining higher-order terms. Thus, at time t_{k+1}

$$\mathbf{x}_{k+1}^{+} = \mathbf{x}_{L,k+1} + \mathbf{x}_{H,k+1} \tag{46}$$

where $\mathbf{x}_{L,k+1}$ indicates the linear part of the Taylor series, while $\mathbf{x}_{H,k+1}$ represents the non-linear terms. For linear and Gaussian systems, $\mathbf{x}_{H,k+1}$ is identically zero and $\mathbf{x}_{L,k+1}$ is the posterior solution found with the Kalman filter equations. For nonlinear systems, $\mathbf{x}_{H,k+1}$ is non-zero and $\mathbf{x}_{L,k+1}$ is not the Kalman filter posterior, however $\mathbf{x}_{L,k+1}$ is still Gaussian distributed because it is the linear combination of three Gaussian random vectors. The mean and covariance matrix of the linear polynomials are readily found

in the DA framework

$$\hat{\mathbf{x}}_{L,k+1} = \mathbb{E}\left\{\mathbf{x}_{L,k+1}\right\} \tag{47}$$

$$\mathbf{P}_{L,k+1} = \mathbb{E}\left\{(\mathbf{x}_{L,k+1} - \hat{\mathbf{x}}_{L,k+1})(\mathbf{x}_{L,k+1} - \hat{\mathbf{x}}_{L,k+1})^{\mathrm{T}}\right\} \tag{48}$$

$$\mathbf{S}_{k+1}\mathbf{S}_{k+1}^{\mathrm{T}} = \mathbf{P}_{L,k+1} \tag{49}$$

Defining $\delta\mathbf{x}_{k+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we can represent $\mathbf{x}_{L,k+1}$ exactly:

$$\mathbf{x}_{L,k+1} = \hat{\mathbf{x}}_{L,k+1} + \mathbf{S}_{k+1}\delta\mathbf{x}_{k+1} \tag{50}$$

The goal is to create a new series approximation of $\mathbf{x}_{k+1}^{+}$, which is only a function of $\delta\mathbf{x}_{k+1}$ rather than the three variables $(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1})$. Effectively, a complex transformation of random vectors is approximated with a surrogate of sufficient accuracy. This approach is motivated by similar approaches in Uncertainty Quantification (UQ) where the surrogate provides analytic knowledge on the propagated uncertainty or knowledge on the quantities of interest via sampling (e.g., see [28], [29]).

The approach leveraged here is similar to generating a polynomial chaos expansion via a non-intrusive approach with least squares (e.g., [30]). The basic idea is to minimize the square of the difference between the full transformation and the surrogate. This is achieved by propagating random samples from the input space, i.e., $\delta\mathbf{x}_k$, $\delta\mathbf{v}_k$, and $\delta\mathbf{w}_{k+1}$, through the function to be approximated and then fitting the surrogate using least squares. Accuracy of the surrogate is determined by the basis and the number of discrete samples.

We draw N independent and identically distributed samples from a $2n+m$ Gaussian distribution with zero mean and identity covariance matrix in order to have N independent samples of $\delta\mathbf{x}_k, \delta\mathbf{v}_k$, and $\delta\mathbf{w}_{k+1}$. Samples are denoted with a superscript $(j),\ j \in [1,\, N]$. The polynomials are then evaluated at the samples, giving N independent realizations of the posterior. We calculate N samples of $\mathbf{x}_{k+1}^{+}$, $\mathbf{x}_{L,k+1}$ and $\delta\mathbf{x}_{k+1}$:

$$\mathbf{x}_{k+1}^{+(j)} = \mathbf{x}_{k+1}^{+}(\delta\mathbf{x}_k^{(j)}, \delta\mathbf{v}_k^{(j)}, \delta\mathbf{w}_{k+1}^{(j)}) \tag{51}$$

$$\mathbf{x}_{L,k+1}^{(j)} = \mathbf{x}_{L,k+1}(\delta\mathbf{x}_k^{(j)}, \delta\mathbf{v}_k^{(j)}, \delta\mathbf{w}_{k+1}^{(j)}) \tag{52}$$

$$\delta\mathbf{x}_{k+1}^{(j)} = \mathbf{S}_{k+1}^{-1}\big(\mathbf{x}_{L,k+1}^{(j)} - \hat{\mathbf{x}}_{L,k+1}\big) \tag{53}$$

where Equation (53) scales and centers the realizations from the linear polynomials, such that their

distribution is a standard Gaussian. The ith component of the new series is

$$\begin{aligned} x^+_{i,k+1} &= x^+_{i,k+1}(\delta \mathbf{x}_{k+1}) \\ &= a_{i,0} + \sum_{r=1}^{c} \frac{1}{r!} \sum_{\ell} a_{i,r,\ell}\ \delta x^{\gamma_1}_{1,k+1} \dots \delta x^{\gamma_r}_{n,k+1} \end{aligned} \tag{54}$$

where the second summation is over all permutations of $\gamma_i \in \{1, \dots, n\}$ with $i \in \{1, \dots, r\}$, describing each possible monomial through combinatorial mathematics.

The above equation is linear in the coefficients, therefore it can be rewritten as

$$x^+_{i,k+1} = \mathbf{\Delta}_{k+1} \mathbf{a}_i \tag{55}$$

where vector $\mathbf{a}_i$ contains the coefficients of the series we are solving for. We can now use the samples of $\mathbf{x}^+_{i,k+1}$ and $\delta \mathbf{x}_{k+1}$ to compute a least square estimate of $\mathbf{a}_i$.

We start by stacking all N realizations ot the ith component $\mathbf{x}^{+(j)}_{i,k+1}$ of the posterior state into a single vector,

$$\tilde{\mathbf{x}}^+_{i,k+1} = \begin{bmatrix} x^{+(1)}_{i,k+1} & x^{+(2)}_{i,k+1} & x^{+(3)}_{i,k+1} & \dots \end{bmatrix}^{\mathrm{T}} \tag{56}$$

while the unknown coefficients of the ith state polynomial are stored in the γ-long vector, $\mathbf{a}_i$. The integer γ expresses the number of coefficients needed to represents a state polynomial component of order c. For example, setting $c = 2$ leads to

$$\gamma = 1 + n + \frac{(n+1)n}{2}$$

where n is the state dimension. Lastly, matrix $\mathbf{\Delta}_{k+1}$, of dimensions $N \times \gamma$, is constructed using deviations $\delta \mathbf{x}^{(j)}_{k+1}$, where each jth row follows Equation (54):

$$\mathbf{\Delta}_{k+1} = \begin{bmatrix} 1 & \delta x^{(1)}_1 & \dots & \delta x^{(1)}_n & (\delta x^{(1)}_1)^2 & \delta x^{(1)}_1 \delta x^{(1)}_2 & \dots \\ 1 & \delta x^{(2)}_1 & \dots & \delta x^{(2)}_n & (\delta x^{(2)}_1)^2 & \delta x^{(2)}_1 \delta x^{(2)}_2 & \dots \\ 1 & \delta x^{(3)}_1 & \dots & \delta x^{(3)}_n & (\delta x^{(3)}_1)^2 & \delta x^{(3)}_1 \delta x^{(3)}_2 & \dots \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{57}$$

where the subscript $k+1$ on the deviations has been omitted.

The coefficients can now be calculated via least square solution to Equation (55)

$$\mathbf{a}_i = \left(\mathbf{\Delta}^{\mathrm{T}}_{k+1} \mathbf{\Delta}_{k+1} \right)^{-1} \mathbf{\Delta}_{k+1} \tilde{\mathbf{x}}^+_{i,k+1} \tag{58}$$

The deviations $\delta \mathbf{x}^{(j)}_{k+1}$ are the same regardless of which component of the state is calculated: for each ith component, we fit different coefficients to their relative realizations. Consequentially, instead of applying

Equation (58) multiple times, it is convenient to rewrite the least square problem in matrix form

$$\mathbf{\Xi}_{k+1} = \begin{bmatrix} \tilde{\mathbf{x}}^{+}_{1,k+1} & \tilde{\mathbf{x}}^{+}_{2,k+1} & \tilde{\mathbf{x}}^{+}_{3,k+1} & \dots \end{bmatrix} \tag{59}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \dots \end{bmatrix} \tag{60}$$

such that the whole set of coefficients is calculated at once:

$$\mathbf{A} = \left(\mathbf{\Delta}^{\mathrm{T}}_{k+1}\mathbf{\Delta}_{k+1}\right)^{-1}\mathbf{\Delta}_{k+1}\mathbf{\Xi}_{k+1} \tag{61}$$

After evaluating all realizations of the random variables, HOPUF directly applies this last equation for an efficient evaluation of the reduced series.

The algorithm is now ready to start the next iteration with a new polynomial propagation.

The number of samples N used in the reduction step of HOPUF-ℓ-c is chosen to obtain an accurate approximation of the surrogate; an exact number can be determined via cross-validation (e.g., see [31]). For an appropriate value of c, a common approximation for the value of N is $2P$-$4P$ where

$$P = \binom{n+c}{c} = \frac{(n+c)!}{n!\,c!} \tag{62}$$

and n is the dimension of the state vector $\mathbf{x}$.

D. Algorithm Summary

The proposed filtering techniques are summarized in Algorithms 1 (HOPUFG-2-c) and 2 (HOPUF-2-c). The algorithms can be expanded to any arbitrary update order ℓ using the results of Section II-A. The two algorithms differ only by the polynomial reduction step. HOPUFG approximates the reduced posterior as a Gaussian random variable. HOPUF applies the least square polynomial reduction step which approximates the non-Gaussian the shape of the posterior. Consequently, HOPUF starts each iteration with a more accurate prior PDF, leading to a performance increase in the estimation accuracy, as shown the numerical applications. The benefits in accuracy form the least squares algorithm are achieved at the expenses of an higher computational cost.

By selecting a linear update, a linear approximation of the functions, and a Gaussian reduction, HOPUFG-1-1 reduces to the classic EKF. Increasing the order of the update, the order of the Taylor series approximation, and/or performing the least-squares reduction step will result in a more accurate filter at the cost of more computations.

Algorithm 1 HOPUFG-2-c

Declare Taylor truncation order c;
Initialize iteration counter $k = 0$;
Initialize state polynomial as Gaussian: $\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_\mathbf{0}, \mathbf{P}_0)$
$\mathbf{S}_0\mathbf{S}_0^T = \mathbf{P}_0$;
$\mathbf{x}_0 = \hat{\mathbf{x}}_\mathbf{0} + \mathbf{S}_0\delta\mathbf{x}_0$; // Polynomial Initialization
$\mathbf{T}_k\mathbf{T}_k^{\mathrm{T}} = \mathbf{Q}_k$;
$\mathbf{U}_k\mathbf{U}_k^{\mathrm{T}} = \mathbf{R}_k$;
while new measurements $\tilde{\mathbf{y}}_{k+1}$ available **do**

$//Prediction//$
$\mathbf{x}_{k+1}^-(\delta\mathbf{x}_k) = \mathbf{f}(\mathbf{x}_k(\delta\mathbf{x}_k))$; // Polynomial Propagation
$\mathbf{x}_{k+1}^-(\delta\mathbf{x}_k, \delta\mathbf{v}_k) = \mathbf{x}_{k+1}^-(\delta\mathbf{x}_k) + \mathbf{T}_k\delta\mathbf{v}_k$; // Add Prop. Noise Variables
$\mathbf{y}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k) = \mathbf{h}(\mathbf{f}(\mathbf{x}_{k+1}^-))$; // Meas. Polynomial
$\mathbf{y}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) := \mathbf{y}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k) + \mathbf{U}_k\delta\mathbf{w}_{k+1}$; //Add Meas. Noise Variables

$//Quadratic\ Update//$
$\mathbf{y}_{k+1}^{[2]} = \mathbf{y}_{k+1} \otimes \mathbf{y}_{k+1}$; // Square Meas. Polynomial;
$\hat{\mathbf{x}}_{k+1}^- = \mathbb{E}\left\{\mathbf{x}_{k+1}^-\right\}$;
$\hat{\mathbf{y}}_{k+1} = \mathbb{E}\left\{\mathbf{y}_{k+1}\right\}$;
$\hat{\mathbf{y}}_{k+1}^{[2]} = \mathbb{E}\left\{\mathbf{y}_{k+1}^{[2]}\right\}$;
$\mathbf{P}_{\mathbf{yy}} = \mathbb{E}\left\{(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1})(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1})^{\mathrm{T}}\right\}$;
$\mathbf{P}_{\mathbf{yy}^{[2]}} = \mathbb{E}\left\{(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1})(\mathbf{y}_{k+1}^{[2]} - \hat{\mathbf{y}}_{k+1}^{[2]})^{\mathrm{T}}\right\}$;
$\mathbf{P}_{\mathbf{y}^{[2]}\mathbf{y}^{[2]}} = \mathbb{E}\left\{(\mathbf{y}_{k+1}^{[2]} - \hat{\mathbf{y}}_{k+1}^{[2]})(\mathbf{y}_{k+1}^{[2]} - \hat{\mathbf{y}}_{k+1}^{[2]})^{\mathrm{T}}\right\}$;
$\mathbf{P}_{\mathcal{Y}\mathcal{Y}} = \begin{bmatrix} \mathbf{P}_{\mathbf{yy}} & \mathbf{P}_{\mathbf{yy}^{[2]}} \\ \mathbf{P}_{\mathbf{y}^{[2]}\mathbf{y}} & \mathbf{P}_{\mathbf{y}^{[2]}\mathbf{y}^{[2]}} \end{bmatrix}$;
$\mathbf{P}_{\mathbf{xy}} = \mathbb{E}\left\{(\mathbf{x}_{k+1}^- - \hat{\mathbf{x}}_{k+1}^-)(\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1})^{\mathrm{T}}\right\}$;
$\mathbf{P}_{\mathbf{xy}^{[2]}} = \mathbb{E}\left\{(\mathbf{x}_{k+1}^- - \hat{\mathbf{x}}_{k+1}^-)(\mathbf{y}_{k+1}^{[2]} - \hat{\mathbf{y}}_{k+1}^{[2]})^{\mathrm{T}}\right\}$;
$\mathbf{P}_{\mathbf{x}\mathcal{Y}} = \begin{bmatrix} \mathbf{P}_{\mathbf{xy}} & \mathbf{P}_{\mathbf{xy}^{[2]}} \end{bmatrix}$;
$\mathbf{K} = \begin{bmatrix} \mathbf{K}_1 & \mathbf{K}_2 \end{bmatrix} = \mathbf{P}_{\mathbf{x}\mathcal{Y}}\mathbf{P}_{\mathcal{Y}\mathcal{Y}}^{-1}$; // Augmented Gain
$\tilde{\mathbf{y}}_{k+1}^{[2]} = \tilde{\mathbf{y}}_{k+1} \otimes \tilde{\mathbf{y}}_{k+1}$; // Square Meas. Outcome
$\mathbf{x}_{k+1}^+(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) = \mathbf{x}_{k+1}^-(\delta\mathbf{x}_k, \delta\mathbf{v}_k) + \mathbf{K}\begin{bmatrix} \tilde{\mathbf{y}}_{k+1} - \mathbf{y}_{k+1}^-(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) \\ \tilde{\mathbf{y}}_{k+1}^{[2]} - \mathbf{y}_{k+1}^{[2]-}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) \end{bmatrix}$; // Polyn. Update
$\hat{\mathbf{x}}_{k+1}^+ = \mathbb{E}\left\{\mathbf{x}_{k+1}^+\right\} = \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}\begin{bmatrix} \tilde{\mathbf{y}}_{k+1} - \hat{\mathbf{y}}_{k+1} \\ \tilde{\mathbf{y}}_{k+1}^{[2]} - \hat{\mathbf{y}}_{k+1}^{[2]} \end{bmatrix}$;
$\mathbf{P}_{\mathbf{xx},k+1} = \mathbb{E}\left\{(\mathbf{x}_{k+1}^+ - \hat{\mathbf{x}}_{k+1}^+)(\mathbf{x}_{k+1}^+ - \hat{\mathbf{x}}_{k+1}^+)^{\mathrm{T}}\right\}$;

$//Gaussian\ Reduction//$
$\mathbf{S}_{\mathbf{xx},k+1} = CholeskyDecomposition(\mathbf{P}_{\mathbf{xx},k+1})$;
$\mathbf{x}_{k+1} = \hat{\mathbf{x}}_{k+1}^+ + \mathbf{S}_{\mathbf{xx},k+1}\delta\mathbf{x}_{k+1}$; // Polynomial DA Scaling
$k = k + 1$;
end while

Algorithm 2 HOPUF-2-c

Declare Taylor truncation order c;
Initialize iteration counter $k = 0$;
Initialize state polynomial as Gaussian: $\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_\mathbf{0}, \mathbf{P}_0)$
$\mathbf{S}_0\mathbf{S}_0^T = \mathbf{P}_0$;
$\mathbf{x}_0 = \hat{\mathbf{x}}_\mathbf{0} + \mathbf{S}_0\delta\mathbf{x}_0$; // Polynomial Initialization
$\mathbf{T}_k\mathbf{T}_k^{\mathrm{T}} = \mathbf{Q}_k$;
$\mathbf{U}_k\mathbf{U}_k^{\mathrm{T}} = \mathbf{R}_k$;
while new measurements $\tilde{\mathbf{y}}_{k+1}$ available **do**

$//Prediction//$
...Same as HOPUFG-2-c ...

$//Quadratic\ Update//$
...Same as HOPUFG-2-c ...

$//Least\ Square\ Reduction//$
Extract linear part from the updated polynomial: $\hat{\mathbf{x}}_{L,k+1}$
$\hat{\mathbf{x}}_{L,k+1} = \mathbb{E}\left\{\mathbf{x}_{L,k+1}\right\}$;
$\mathbf{P}_{L,k+1} = \mathbb{E}\left\{(\mathbf{x}_{L,k+1} - \hat{\mathbf{x}}_{L,k+1})(\mathbf{x}_{L,k+1} - \hat{\mathbf{x}}_{L,k+1})^{\mathrm{T}}\right\}$;
$\mathbf{S}_{k+1} = CholeskyDecomposition(\mathbf{P}_{L,k+1})$;
Draw N samples $\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}$ from $\mathcal{N}(\mathbf{0}, \mathbf{I})$;
for $j = 1 : N$ **do**
$\delta\mathbf{x}_{k+1}^{(j)} = \mathbf{S}_{k+1}^{-1}\left(\mathbf{x}_{L,k+1}^{(j)} - \hat{\mathbf{x}}_{L,k+1}\right)$;
$\mathbf{x}_{k+1}^{+(j)} = \mathbf{x}_{k+1}^{+}(\delta\mathbf{x}_k^{(j)}, \delta\mathbf{v}_k^{(j)}, \delta\mathbf{w}_{k+1}^{(j)})$;
$\mathbf{x}_{L,k+1}^{(j)} = \mathbf{x}_{L,k+1}(\delta\mathbf{x}_k^{(j)}, \delta\mathbf{v}_k^{(j)}, \delta\mathbf{w}_{k+1}^{(j)})$;
end for
for $i = 1 : n$ **do**
$\tilde{\mathbf{x}}_{i,k+1}^{+} = \begin{bmatrix} x_{i,k+1}^{+(1)} & x_{i,k+1}^{+(2)} & x_{i,k+1}^{+(3)} & \dots \end{bmatrix}^{\mathrm{T}}$;
end for
$\boldsymbol{\Xi}_{k+1} = \begin{bmatrix} \tilde{\mathbf{x}}_{1,k+1}^{+} & \tilde{\mathbf{x}}_{2,k+1}^{+} & \tilde{\mathbf{x}}_{3,k+1}^{+} & \dots \end{bmatrix}$;
$\boldsymbol{\Delta}_{k+1} = MatrixOfDeviations(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1})$; // LS matrix form the N samples
$\mathbf{A} = \left(\boldsymbol{\Delta}_{k+1}^{\mathrm{T}}\boldsymbol{\Delta}_{k+1}\right)^{-1}\boldsymbol{\Delta}_{k+1}\boldsymbol{\Xi}_{k+1}$;
Extract LS coefficients from $\mathbf{A}$ to create new prior state polynomials;
$k = k + 1$;
end while

IV. Numerical Examples

A. Scalar Problem

A simple scalar problem is presented here to underline the improvements in accuracy gained by a nonlinear estimator over linear ones. The following application shows how higher orders of polynomial estimators are better approximations of the MMSE.

Define a prior state $x \sim \mathcal{N}(0, 0.1)$ and a measurement

$$y = arctan(x) + \eta \tag{63}$$

where $\eta \sim \mathcal{N}(0, 10^{-2})$ is measurement noise.

Figure 1 shows the true joint distribution of x and y represented using 10^5 samples (blue dots in the figure). The optimal (nonlinear) MMSE is the conditional mean, which visually is the curved line dividing in half the distribution of y (horizontal spread of blue points) for each value of x. The figure also shows different estimators: the EKF, the UKF and HOPUF-ℓ-c. The notation HOPUF-ℓ-c indicates with c the selected truncation order of the Taylor series expansion, and with ℓ the order of the update.

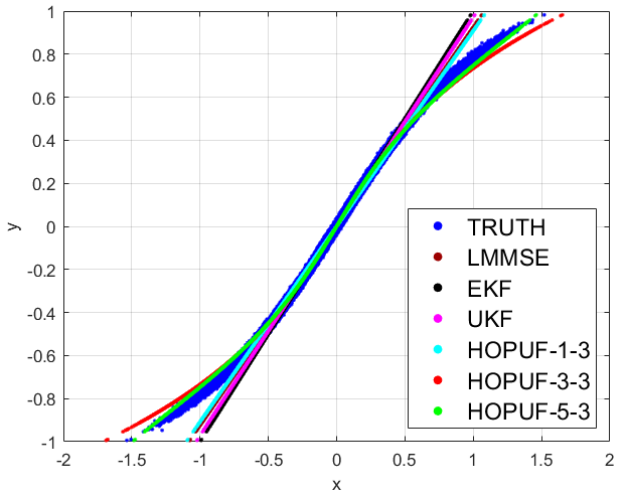


Fig. 1. Representation of the state-measurement joint distribution (blue points) and of the relative estimates from different estimators.

Both the EKF and UKF are linear estimators, therefore their representations in Figure 1 are straight lines, black and magenta, respectively. The slope of the lines is the Kalman gain, whose optimal value is $\mathbf{P_{xy}}\mathbf{P_{yy}^{-1}}$; different approximations of $\mathbf{P_{xy}}$ and $\mathbf{P_{yy}}$ result in different values of the Kalman gain and different slopes in the figure. The EKF is a local approximation, hence the line is aligned with the slope of the optimal MMSE at the prior mean, and has the largest deviation from the optimal MMSE at the edges of the distribution. The brown line in the figure shows the optimal Linear MMSE (LMMSE) which is given by

$$\hat{\mathbf{x}}^+ = \mathbb{E}\{\mathbf{x}\} + \mathbf{P_{xy}}\mathbf{P_{yy}^{-1}}(\mathbf{y} - \mathbb{E}\{\mathbf{y}\}) \tag{64}$$

where the values of $\mathbf{P_{xy}}$, $\mathbf{P_{yy}}$, and $\mathbb{E}\{\mathbf{y}\}$ are calculated from the 10^5 samples of the joint distribution of $\mathbf{x}$ and $\mathbf{y}$. The Sigma Points employed in the UKF allow for some global information, as such, the

UKF provides an approximation closer to the LMMSE than the EKF's.

The DA-based polynomial estimator is reported with three different orders: a linear estimator, a cubic, and a fifth order one, all employing a third order Taylor series approximation of the $arctan$ function. The estimates from HOPUF-1-3 form a straight line, depicted in cyan, with different slope with respect to the EKF and the UKF; HOPUF-1-3 is almost completely superimposed and indistinguishable from the optimal LMMSE. Therefore, the higher order polynomial representation of the measurement equation leads to a better approximation of the true LMMSE (when compareted to the EKF and UKF) and it improves accuracy, as shown by the RMSE analysis in Figure 2.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N_{samples}} (x_i - \hat{x}_i^+)^2}{N_{samples}}} \tag{65}$$

The DA estimator uses Taylor polynomials approximation to represent the system function and the moments of the PDF are calculated accordingly. In this example, a third order Taylor series approximation is used, therefore the Kalman gain evaluated by the LMMSE differs from the EKF, that uses linearization (Jacobian), and from the UKF, that applies the unscented transformation.

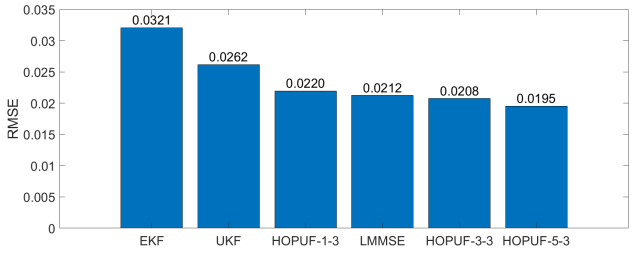


Fig. 2. RMSE

Figure 2 shows the RMSE of each estimator evaluated using the entire set of 10^5 samples. The first 3 bars in the figure show that the more accurate representation of the measurement equation relates to higher accuracy and to lower error level of the estimator, while still having a linear dependency on the measurement outcome, HOPUF-1-3 shows lower estimation error than the EKF and the UKF, indicating it is a closer approximation to the true LMMSE estimator.

In Figure 1, the Cubic MMSE (HOPUF-3-3) and the 5th order MMSE (HOPUF-5-3) are also reported, with red and green points, respectively. They both work with a 3rd order polynomial truncation of the Taylor series representation of the $arctan$ function. These estimators are a polynomial function of the measurement and, therefore, they can better follow the true (nonlinear) behavior of the MMSE. The estimator functions are curved and follow the tails of the distribution. The improvement in accuracy can

be appreciated by looking at the RMSE values in Figure 2: the higher the order of the update, the lower the RMSE, which leads to a more accurate estimate.

This simple example shows how the LMMSE can be interpreted as the first order approximation of the true nonlinear MMSE, while the HOPUF-3-3 and HOPUF-5-3 are respectively the 3rd and 5th order approximations. Theoretically, under some regularity assumptions on the optimal MMSE function, by increasing the estimators order to infinity, we will asymptotically reach the true MMSE.

As a consequence, high order update filters, such as HOPUF-ℓ-c, achieve better results in terms of accuracy and robustness when compared to linear estimators. The improvement in robustness is shown in the next example.

B. Lorenz96 System

The performance of the proposed filter is tested in a Lorenz96 example [20] [21]. The estimated state is:

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) & x_2(t) & x_3(t) & x_4(t) \end{bmatrix}^{\mathrm{T}}$$

and the state dynamics are

$$\frac{dx_i(t)}{dt} = x_{i-1}(t)(x_{i+1}(t) - x_{i-2}(t)) - x_i(t) + F + \nu_i(t) \tag{66}$$

where $i = 1, \ldots, 4$ since $\mathbf{x}(t)$ is four-dimensional. The following conventions are used: $x_{-1}(t) = x_{n-1}(t)$, $x_0(t) = x_n(t)$, and $x_1(t) = x_{n+1}(t)$. The term F is a constant external force and it is chosen equal to eight in order to introduce a chaotic behavior in the system. The initial condition is assumed to be Gaussian, with mean $\hat{\mathbf{x}} = \begin{bmatrix} F & F & F + 0.01 & F \end{bmatrix}^T$ and diagonal covariance matrix, with the same standard deviation for each component of the state: $\sigma_{\mathbf{x}} = 10^{-3}$. The process noise is assumed to be Gaussian and uncorrelated among states, with known standard deviation $\sigma_{\boldsymbol{\nu}} = 10^{-3}$.

Measurements are acquired at discrete time-steps t_k according to the following model:

$$\mathbf{y}_k = \mathbf{H}_{i,j}\mathbf{x}(t_k) + \boldsymbol{\mu}_k, \quad \mathbf{H} = \begin{cases} 1 & j = 2i - 1 \\ 0 & \text{otherwise} \end{cases} \tag{67}$$

with $i = \{1, 2\}$ and $j = \{1, 2, 3, 4\}$. In other words, the measurements are linear and the components of the state vector with odd indeces are measured. The measurement noises, uncorrelated from the process noise and with each other, are assumed to be Gaussian as well with standard deviation $\sigma_{\boldsymbol{\mu}} = 10^{-2}$. The dynamics are propagated at 2 Hz with a Runge-Kutta 7-8 integrator.

Figure 3 shows the Monte Carlo analysis performed with HOPUF-2-2 on the presented problem. Once again, the notation HOPUF-ℓ-c indicates the selected truncation order of the Taylor series expansion c and the order of the update ℓ. The value $N = 1000$ samples is chosen for the least squares problem and the number of Monte Carlo runs is also 1000. Figure 3 shows the 1000 realizations of the estimation error (gray lines), evaluated as distance between the estimate and the true state, for each one of the four state components. For the i-th component of the state

$$\epsilon_{j,i} = x_{j,i} - \hat{x}_{j,i} \tag{68}$$

The algorithms' predicted estimation error standard deviation of each state is reported in green lines (3σ values), calculated as the square roots of the diagonal terms of $\mathbf{P_{xx}}$ in Equation (43). The actual performance of the system is assessed by the sample standard deviation of the Monte Carlo estimation errors, represented in the figure as blue lines (3σ values). The consistency of HOPUF-2-2 is established by the close values of the estimated and effective standard deviations, as shown by the overlapping of the green and blue lines. Finally, the black lines are the sample means of the estimation errors at each time-step and they demonstrate the unbiased nature of the filter, as expected from the theory of MMSE estimation.

The benefits of the quadratic update can be appreciated when comparing HOPUF-2-2 with linear estimators, such as the EKF, the UKF [6], [32] and the DAHO-k [9]. DAHO-k is a linear estimator that approximates the nonlinear functions with their kth order Taylor series representation. The EKF performs a simple linearization of the equation of motion and, in the presence of high nonlinearities such as in the Lorenz96 problem, fails to estimate the state of the system (therefore it is not reported in the figures). The UKF and the DAHO-k, on the other hand, are linear estimators that better account for the system's nonlinearities and achieve better performance than the EKF [26]. Figure 4 compares HOPUF-2-2, with DAHO-2 and the UKF. DAHO-k is a well suited test bench for higher-order linear estimators [17], [23].

Figure 4 has six lines. The three continuous lines represent the filter's own estimate of accuracy in terms of the predicted estimation error standard deviation, calculated from the covariance matrix as the square root of its trace: $\bar{\sigma} = \sqrt{\text{tr}(\mathbf{P_{xx}})}$. The three dashed lines represent the effective error standard deviation derived from the Monte Carlo analysis (1000 runs). The filter is consistent when it predicts its own uncertainty, i.e., when the continuous and dashed lines coincide. The figure shows how the quadratic update (blue lines) reduces the system uncertainty, improving accuracy especially in the first steps of the simulation, during the transient behavior. HOPUF-2-2 reaches steady state quickly and the filter avoids overconfidence. The linear update of DAHO-2 (red lines), on the other hand, has a much slower and less

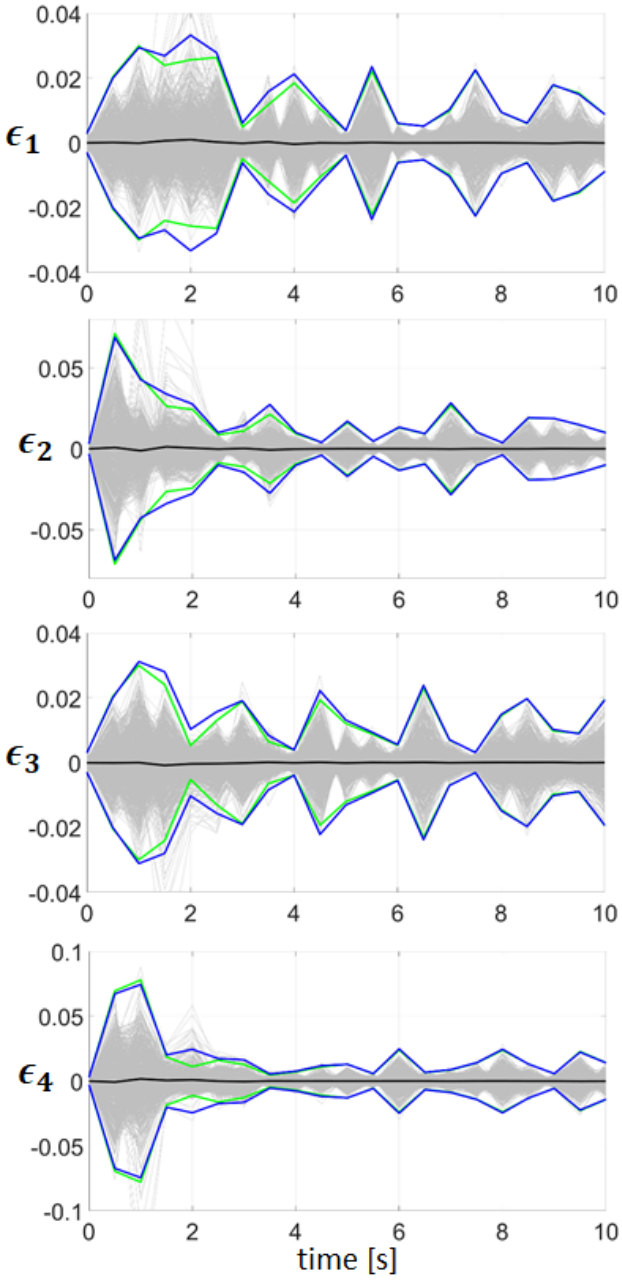


Fig. 3. 1000 runs Monte Carlo performance test for the HOPUF-2-2, $N = 1000$.

accurate transient response but it is consistent and eventually converges. Once steady-state is reached, the nonlinearities cease to dominate and the quadratic update no longer significantly outperforms the linear update. The UKF (green lines) is inconsistent as the filter's prediction of its own estimation error does not match the actual behavior, which, as shown by the dashed green line, does not settle to the same accuracy level as HOPUF-2-2 and DAHO-2.

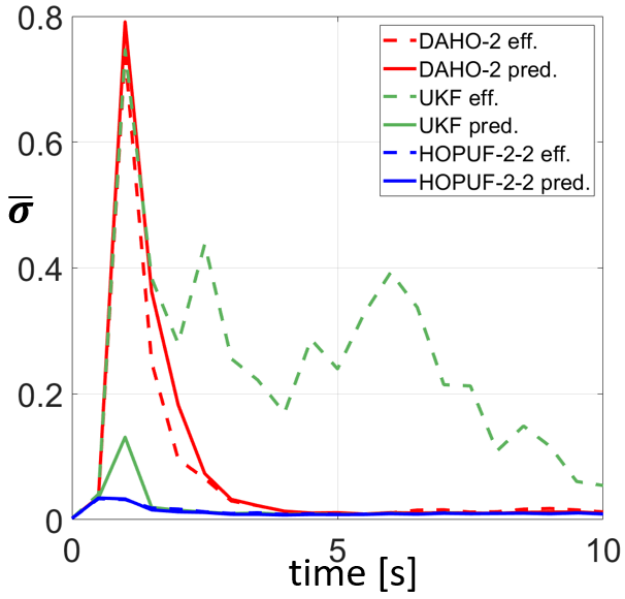


Fig. 4. Covariance comparison: quadratic vs. linear estimator. Solid lines are the filters own predictions, dashed lines represent the sample standard deviation from Monte Carlo.

For the Lorenz96 example considered, the new quadratic estimator exhibits more robustness than the linear estimators tested. Increasing the measurement noise levels or decreasing the measurement frequency causes divergence of the DAHO-k as tested, while the consistency of HOPUF-2-2 is not affected by those changes (the estimation accuracy on the other hand degrades, because of fewer or less precise measurements).

Therefore, a new simulation has been performed after increasing the measurement noise standard deviation to a higher level, $\sigma_{\mu} = 0.5$, while keeping all the other parameters fixed. The performance of HOPUFG-2-2 and HOPUF-2-2 are shown in Figure 5.

Figure 5 reports, in a logarithmic scale, the standard deviation analysis of a 40 seconds simulation. HOPUF-2-2, in its two versions, is the only filter shown, as the linear estimators (EKF, UKF and DAHO-k) diverge and fail to estimate the state of the system, and are therefore omitted. The figure shows the accuracy improvements given by the non Gaussian representation of the updated state polynomials achieved by the least squares reduction. The blue lines represent the behavior of the proposed filter when the distribution is assumed as Gaussian at the beginning of each time step (HOPUFG-2-2). The red lines show the accuracy level achieved by HOPUF-2-2: they remain below the blue lines during the whole simulation, indicating smaller estimation error. Moreover, the effective (dashed) line from the Monte Carlo runs and the predicted (continuous) line form the filter's equations overlap closely for HOPUF-2-2, showing consistency, while HOPUFG-2-2 is too conservative on its uncertainty estimation.

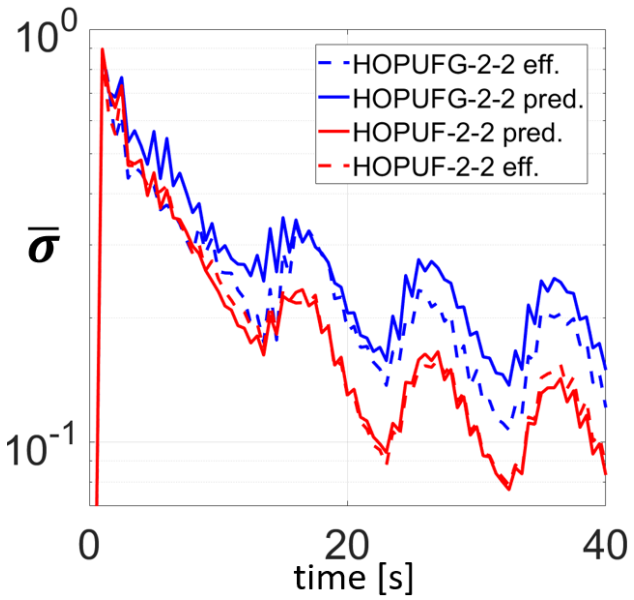


Fig. 5. Covariance comparison: HOPUFG vs. HOPUF

We attribute this discrepancy to the fact that, for a fixed mean and covariance, a Gaussian distribution is less informative than any other distribution. Hence, successive updates performed after replacing the calculated posterior with a Gaussian with the same mean and covariance do not infer as much information about the system. This approximation is the most conservative approximation that can be made given the mean and covariance, since the Gaussian distribution has the highest entropy for a fixed covariance [33]. The higher accuracy level achieved by the non Gaussian representation of the prior for the following step is paid by the higher computational burden requested by the least square polynomial fit. HOPUFG-2-2, through the Gaussian approximation of the state prior, is a computational faster filter that, at each time step, scales the polynomial coefficients using the most recent update covariance.

By increasing the measurement noise, filters that rely only on the linear dependency from the measurements are not able to estimate the state of the system. The quadratic update improves accuracy by evaluating not only the square of the measurements, but also its higher order moments, which provide more precise (Kalman) gains. The measurement information is better fused with the state prediction to produce superior performances.

V. Conclusions

An estimator with quadratic update has been presented. Unlike prior techniques, the proposed approach does not require storing higher order central moments of the state's distribution. The new technique accounts for the nonlinearities of the system both in the prediction and in the update step by approximating

the distributions resulting from nonlinear transformations as polynomial functions of Gaussian random vectors. The new approach is easily expandable to any order of the polynomial update, as shown by the proposed scalar example. For systems other than linear and Gaussian (where a linear update is globally optimal) the higher the order of the polynomial update, the more precise the resulting state estimate.

The proposed algorithm can be interpreted as an expansion of Gaussian filters. Gaussian filters approximate uncertainties as Gaussian. By representing the state uncertainty with an arbitrary polynomial function of a Gaussian random vector, it is possible to better approximate the shape of distributions undergoing nonlinear dynamics and measurements.

ACKNOWLEDGMENT

This work was sponsored in part by the Air Force Office of Scientific Research under grant number FA9550-18-1-0351

REFERENCES

[1] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems, Journal of Basic Engineering", March, doi:10.1115/1.3662552, No. Series D, pp. 35–45, Vol. 82, 1960

[2] Kalman, R. E. and Bucy, R. S., "New Results in Linear Filtering and Prediction", Journal of Basic Engineering, March, doi:10.1115/1.3658902, No. Series D, pp. 95–108, Vol. 83, 1961

[3] Gelb, A., "Applied Optimal Estimation", Massachusetts Institute of Technology, ISBN:9780262200271, Cambridge, MA, The MIT press, 1974

[4] Junkins, J. and Singla, P., "How Nonlinear is it? A tutorial on Nonlinearity of Orbit and Attitude Dynamics", Journal of the Astronautical Sciences, No. 1-2, pp. 7–60, Vol. 52, 2004

[5] Julier, S. J. and Uhlmann, J. K. and Durrant-Whyte, H. F., "A new method for the nonlinear transformation of means and covariances in filters and estimators", IEEE Transactions on Automatic Control, March, doi: 10.1109/9.847726, No. 3, pp. 477–482, Vol. 45, 2000

[6] Julier, S. J. and Uhlmann, J. K., "Unscented filtering and nonlinear estimation", Proceedings of the IEEE, March, doi: 10.1109/JPROC.2003.823141, No. 3, pp. 401–422, Vol. 92, 2004

[7] Park, R. S. and Scheeres, D. J., "Nonlinear mapping of Gaussian statistics: theory and applications to spacecraft trajectory design", Journal of Guidance, Control, and Dynamics, doi: 10.2514/1.20177, No. 6, pp. 1367–1375, American Institute of Aeronautics and Astronautics, Vol. 29, 2006

[8] Majji, M and Junkins, J. and Turner, J., "A High Order Method for Estimation of Dynamic Systems", The Journal of the Astronautical Sciences, doi: 10.1007/BF03256560, No. 3, pp. 401–440, Vol. 56, 2008

[9] Valli, M., Armellin, R., Di Lizia, P. and Lavagna, M., "Nonlinear mapping of uncertainties in celestial mechanics", Journal of Guidance, Control, and Dynamics, doi: 10.2514/1.58068, No. 1, pp. 48–63, Vol. 36,2012

[10] Rasotto, M., Morselli, A., Wittig, A., Massari, M., Di Lizia, P., Armellin, R. and Valles, C. and Ortega, G., "Differential algebra space toolbox for nonlinear uncertainty propagation in space dynamics", 2016

[11] Massari, M., Di Lizia, P. and Rasotto, M., "Nonlinear uncertainty propagation in astrodynamics using differential algebra and graphics processing units", Journal of Aerospace Information Systems, Vol. 14, No. 9, pp. 493–503, 2017, American Institute of Aeronautics and Astronautics

[12] Sorenson, H. W. and Alspach, D. L., “Recursive Bayesian Estimation Using Gaussian Sums”, Automatica, July, doi:10.1016/0005-1098(71)90097-5, No. 4, pp. 465–479, Vol. 7, 1971

[13] Alspach, D. and Sorenson, H., “Nonlinear Bayesian estimation using Gaussian sum approximations”, IEEE Transactions on Automatic Control, August, doi: 10.1109/SAP.1970.270017, No. 4, pp. 439–448, Vol. 17, 1972

[14] Gordon, N. J., Salmond, D. J., and Smith, A. F. M., “Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation, Radar and Signal Processing, IEE Proceedings F, Vol. 140, No. 2, April 1993, pp. 107113. doi:10.1049/ip-f-2.1993.0015

[15] De Santis, A., Germani A. and Raimondi, M., “Optimal quadratic filtering of linear discrete-time non-Gaussian systems”, doi:10.1109/9.400478, IEEE Transactions on Automatic Control, No. 7, pp. 1274-1278, Vol. 40, 1995, http://dx.doi.org/10.1109/9.400478.

[16] Carravetta, F., Germani, A. and Raimondi, M., “Polynomial filtering of discrete-time stochastic linear systems with multiplicative state noise”, IEEE Transactions on Automatic Control, doi:10.1109/9.618240, No. 8, pp. 1106-1126, Vol. 42, 1997.

[17] Servadio, S. and Zanetti, R., “Recursive Polynomial Minimum Mean-Square Error Estimation with Applications to Orbit Determination”, Journal of Guidance, Control, and Dynamics, pp. 1–16, 2020, American Institute of Aeronautics and Astronautics

[18] Isserlis, L., “On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables”, Biometrika, Vol. 12, No. 1/2, pp. 134–139, 1918, JSTOR

[19] Servadio, S., Zanetti, R. and Jones, B. A.,“Nonlinear Filtering with Polynomial Series of Gaussian Random Variables”,2020 International Conference on Information Fusion (FUSION)

[20] Raihan, D. and Chakravorty, S., “Particle Gaussian mixture filters-I”, Automatica, Vol. 98, pp. 331–340, 2018, Elsevier

[21] Ott, E., Hunt, B. R., Szunyogh, I., Zimin, A. V., Kostelich, E. J., Corazza, M., Kalnay, E., Patil, DJ. and Yorke, J. A., “A local ensemble Kalman filter for atmospheric data assimilation”, Tellus A: Dynamic Meteorology and Oceanography, Vol. 56, No. 5, pp. 415–428, 2004, Taylor & Francis

[22] Di Lizia, P., Massari, M. and Cavenago, F., “Assessment of onboard DA state estimation for spacecraft relative navigation”, Final report, ESA , 2017

[23] Cavenago, F., Di Lizia, P., Massari, M., Servadio, S. and Wittig, A., “DA-based nonlinear filters for spacecraft relative state estimation”, 2018 Space Flight Mechanics Meeting, doi: 10.2514/6.2018-1964, p. 1964, 2018

[24] Valli, M., Armellin, R., Di Lizia, P., and Lavagna, M., Nonlinear mapping of uncertainties in celestial mechanics, Journal of Guidance, Control, and Dynamics, 2013, doi: 10.2514/1.58068.

[25] Armellin, R., Di Lizia, P., Bernelli-Zazzera, F., and Berz, M., Asteroid close encounters characterization using differential algebra: the case of Apophis, Celestial Mechanics and DynamicalAstronomy, Vol. 107, No. 4, 2010, pp. 451–470, doi:10.1007/s10569-010-9283-5.

[26] Servadio, S., “High Order Filters For Relative Pose Estimation Of An Uncooperative Target”, Politecnico di Milano, Master Thesis, 2017, Milan, Italy, 20156.

[27] Makino, K. and Berz, M.,Cosy infinity version 9, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 558, No. 1, 2006, pp. 346–350, doi: 10.1016/j.nima.2005.11.109.

[28] Xiu, D., “Numerical Methods for Stochastic Computations: A Spectral Method Approach”, Princeton, New Jersey, Princeton University Press, 2010

[29] Xiu, D., “Handbook of Uncertainty Quantification”, ch. Stochastic Collocation Methods: A Survey, pp. 699-716, Springer International Publishing, 2016.

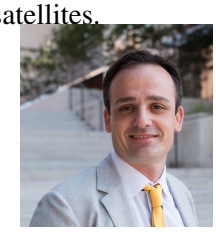
[30] Najm, H. N., “Uncertainty Quantification and Polynomial Chaos Techniques in Computational Fluid Dynamics”, Annual Reviews of Fluid Mechanics, vol. 41, pp. 35–52, 2009

[31] Jones, B. A. and Doostan, A., “Satellite Collision Probability Estimation Using Polynomial Chaos Expansions”, Advances in Space Research, vol. 52, pp. 1860–1875, 2013

[32] Uhlmann j. K., Julier, J. and Durrant-Whyte, H. F., “A new approach for the nonlinear transformation of means and covariances in linear filters”, IEEE Trans. Automatic Control, 1996

[33] Cover, T. M. and Thomas, J. A., Elements of Information Theory, Wiley-Interscience, pp. 254-255, Hoboken, New Jersey, Second Edition, 2006.

Simone Servadio is a Ph.D. candidate at The University of Texas at Austin, TX, USA. He graduated from Politecnico di Milano, Milano, ITA, with a Bachelor degree in Aerospace Engineering in 2015 and with a Master degree in Space Engineering in 2017. His research focuses on nonlinear filtering for GN%C applications, orbit determination, and relative pose estimation between satellites.



Renato Zanetti is an Assistant Professor of Aerospace Engineering at The University of Texas at Austin. Prior to joining UT he worked as an engineer at the NASA Johnson Space Center and at Draper Laboratory. Renato’s research interest include nonlinear estimation, onboard navigation, and autonomous aerospace vehicles

Brandon A. Jones received the Ph.D. degree in aerospace engineering sciences in 2010 from the University of Colorado Boulder, Boulder, CO, USA. He was a Research Assistant Professor with the University of Colorado Boulder until December 2015. He is currently an Assistant Professor with the Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX, USA. His research interests include space situational awareness, multitarget tracking, computational methods for orbit state and uncertainty propagation, and satellite navigation.