Estimation of the Conditional State and Covariance With Taylor Polynomials

SIMONE SERVADIO
RENATO ZANETTI

A novel estimator is presented that expands the typical state and covariance update laws of Kalman filters to polynomial updates in the measurement. The filter employs Taylor series approximations of the nonlinear dynamic and measurement functions. All polynomials (functions approximation, state update, and covariance update) can be selected up to an arbitrary order to trade between filter's accuracy and computational time. The performance of the algorithm is tested in numerical simulations.

Manuscript received January 19, 2021; revised October 24, 2021; released for publication December 1, 2021.

Associate Editor: Paolo Braca.

S. Servadio is with the Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: simoserv@mit.edu).
R. Zanetti is with the University of Texas at Austin, Austin, TX 78705 USA (e-mail: renato@utexas.edu).

This work was sponsored by the Air Force Office of Scientific Research under Grant FA9550-18-1-0351.

I. INTRODUCTION

Estimation is the process of inferring the value of a quantity of interest from indirect, inaccurate, and noisy observations. When the quantity of interest is the (current) state of a dynamic system, the problem is often referred to as "filtering": The best estimate is obtained by "filtering out" the noise from noisy measurements. The estimate is the output given by an optimal estimator, which is a computational algorithm that processes measurements while maximizing a certain performance index. The optimal estimator makes the best use of the data, of the knowledge of the system, and of the disturbances.

For the well-known linear and Gaussian cases, the posterior distribution remains Gaussian and the Kalman Filter [21], [22] provides the mechanization to calculate its mean and covariance matrix. However, most practical problems are nonlinear in the dynamics and in the measurement equations, leading to non-Gaussian probability density functions (PDFs).

Many techniques have been developed to deal with the nonlinear estimation problem. A simple solution is based on the linearization of the dynamics and measurement equations around the most current estimate. The Extended Kalman Filter (EKF) [13] algorithm applies the KF mechanization to the linearized system. Another well-known technique to account for the system nonlinearities is the unscented transformation. The Unscented Kalman Filter (UKF) [19], [20] is able to better handle the effects of nonlinearities in the dynamics and in the measurements and, typically, achieves higher accuracy and robustness levels when compared to the EKF. The UKF applies the unscented transformation to achieve a more accurate approximation of the predicted mean and covariance matrix. The UKF is a linear estimator, i.e., the estimate is a linear function of the current measurement.

The first-order approximation of the EKF can be extended to higher order Taylor series [10], [13]. Generally, the higher the order of the Taylor series, the better the performance of the filter. The Gaussian Second Order Filter (GSOF) [18] truncates the Taylor series at second order to better account for the system's nonlinearities. Truncating the Taylor series to order c requires knowledge of the estimation error's central moments up to order $2c$ in order to calculate the Kalman gain. For example, the EKF truncates at first order, and it requires knowledge of the covariance matrices. Consequentially, the GSOF requires knowledge of the third and forth central moments of the state distribution. At each iteration, the GSOF approximates the prior PDF as Gaussian so that the third-order central moment is zero and the fourth is easily calculated from the covariance matrix. The GSOF performs a linear update based on a second-order approximation of the posterior estimation error. Linear Gaussian filters exist up to any arbitrary truncation order of the Taylor series approximation of the dynamic/measurement functions [34], [35].

Other linear filters make different types of approximations, such as Gaussian quadrature (QKF) [3], spherical cubature (CKF) [2], ensemble points (EnKF) [42], central differences (CDKF) [32], and finite differences (DDKF) [31].

All of the filters mentioned above are linear estimators, i.e., the estimate is a linear function of the current measurement. The conditional mean, which is the optimal minimum mean square error (MMSE) solution, is typically some unknown nonlinear function of the measurement whose exact computation is usually not feasible. 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) [1], [40] or Particle Filters (PFs) such as Bootstrap PF (BPF) [15], Marginalized PF (MPF) [33], Auxiliary PF (APF), Unscented PF (UPF) [44], Gaussian PF (GPF) [16], and Monte Carlo Filter PF (MCFPF) [29].

Ref. [14] derives the evolution of the conditional mean, covariance, and higher order moments of a dynamic system subject to continuous measurements. To make the solution practical, the nonlinear dynamic and measurement equations are approximated with Taylor series expansions.

Another, less studied, approach to nonlinear filtering is to expand the linear update structure to a polynomial update function of the measurement. De Santis *et al.* [11] propose an augmented state to obtain a polynomial update but preserving the linear update structure. Their work augments the measurement vector with its square to form a quadratic update [11] and was extended to polynomial updates [8]. Li *et al.* [23] propose to augment the measurement vector with uncorrelated nonlinear conversions. Similarly to [8] and [11], Liu *et al.* [26] obtain a nonlinear estimator preserving the linear structure of the measurement update. The mean square error (MSE) can be minimized by an optimal selection of the uncorrelated functions [24]. Later, Zhang and Lan [24] merged with the GSF mathematics [46]. Servadio and Zanetti [36] also implemented a quadratic update (extendable to polynomial update of any order) based on Taylor series expansions. The polynomial update requires knowledge of high-order central moments, and [36] carries these moments, exactly like the EKF carries mean and covariance. The computational demand of carrying higher order central moments (propagating forward in time and updating with measurement data) grows quickly with the truncation order of the Taylor series, the size of the state vector, and the order of the polynomial update. Ref. 38 performs a polynomial update without carrying the higher order central moments and, hence, reduces overall computational cost by approximating non-Gaussian distributions as polynomial transformation of Gaussian random variables. In doing so, all high-order central moments are easily and efficiently calculated in a closed form. Consequently, in [38], polynomial updates can be performed much more efficiently than in [36].

The update methodologies presented in [8], [11], [23], [36], and [38] produce a more precise state estimate than those produced by a linear state update. This work introduces a higher order update for the covariance matrix as well as for the state update, which results in a more accurate quantification of the uncertainty associated with the estimate. In turn, the more accurate uncertainty representation produces a more accurate estimator and, hence, a reduced estimation error.

This article is structured in the following way. First, a short background section highlights the novel contributions of the work. This is followed by the development of the new methodology and by applications to three numerical examples. Lastly, conclusions are drawn.

II. BACKGROUND

The linear update rule for mean $\hat{\mathbf{x}}^+$ and covariance matrix $\mathbf{P}^+_{\mathbf{xx}}$ are given by

$$\hat{\mathbf{x}}^+ = \hat{\mathbf{x}}^- + \mathbf{K}(\tilde{\mathbf{y}} - \hat{\mathbf{y}}^-), \tag{1}$$

$$\mathbf{P}^+_{\mathbf{xx}} = \mathbf{P}^-_{\mathbf{xx}} - \mathbf{K}\mathbf{P}_{\mathbf{yy}}\mathbf{K}^T, \tag{2}$$

where $\mathbf{K}$ is the Kalman gain, $\tilde{\mathbf{y}}$ is the measurement outcome, $\hat{\mathbf{y}}^-$ is the predicted measurement mean, $\hat{\mathbf{x}}^-$ is the prior mean, $\mathbf{P}^-_{\mathbf{xx}}$ is the covariance of the state, and $\mathbf{P}_{\mathbf{yy}}$ is the covariance of the measurement. The above equations are optimal in an MMSE only when the prior distribution and the measurement are jointly Gaussian (which implies a linear relation between the two). In general, the MMSE estimate is the conditional mean, an unknown and typically nonlinear function of the measurement outcome; Equation (1) is the statistical linear regression of the conditional mean [25], that is to say, Equation (1) is the best linear fit of the conditional mean with respect to a MSE performance index

$$\hat{\mathbf{x}}^+ \approx \mathbb{E}\left\{\mathbf{x}\middle|\mathbf{y} = \tilde{\mathbf{y}}\right\},$$

where the approximation holds to first order. Equation (2), on the other hand, is the total covariance of the estimation error:

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

but it is also the best constant approximation of the conditional covariance of the state given the measurement, also in an MSE sense.

$$\mathbf{P}^+_{\mathbf{xx}} \approx \mathbb{E}\left\{(\mathbf{x} - \mathbb{E}\{\mathbf{x}\})\,(\mathbf{x} - \mathbb{E}\{\mathbf{x}\})^T\middle|\mathbf{y} = \tilde{\mathbf{y}}\right\},$$

where the approximation holds to zeroth order.

For nonlinear dynamics/measurements, the linear update equations above are not fully recursive, and processing nonlinear measurements as a batch is more accurate than processing them individually [36]. For nonlinear systems, Bayes' rule can be applied recursively to

obtain an optimal estimator, that is to say, the quantity to be calculated recursively is the conditional PDF given the measurements outcome. Hence, a linear recursive filter can be interpreted as an approximated filter where the distribution of the state given the measurements is approximately Gaussian with mean $\hat{\mathbf{x}}^+$ and covariance matrix $\mathbf{P}^+_{\mathbf{xx}}$.

Experience has shown that the order of the statistical regression approximation of the covariance needs to be lower than that of the mean in order to obtain good numerical performance of the algorithm. A zeroth-order covariance approximation, therefore, has endured as a companion of a linear mean update rule, but it is also used in higher order update methodologies [8], [11], [23], [36], [38]. Our prior work, HOPUF-ℓ-c [38], presents a high-order polynomial state update, i.e., a higher-than-linear polynomial approximation of the conditional mean. This article presents a novel higher order polynomial covariance update to better approximate the conditional covariance than the standard zeroth-order approach.

A. The Polynomial Estimator

Gaussian filters are linear filters that approximate the distribution of the state given the measurements as Gaussian with mean $\hat{\mathbf{x}}^+$ and covariance matrix $\mathbf{P}^+_{\mathbf{xx}}$. This is equivalent to approximating the distribution of the state given the measurements as a linear transformation of a standard normal. This linear transformation is given by a shift of $\hat{\mathbf{x}}^+$ and a scale of $\sqrt{\mathbf{P}^+_{\mathbf{xx}}}$.

Our previous work (HOPUF-ℓ-c) expanded this concept by introducing a filter that approximates the distribution of the state given the measurements as a polynomial transformation of standard normal random variables and uses a higher-than-linear polynomial update function. This work introduces a novel covariance update technique and uses the HOPUF-ℓ-c state update, which is summarized here.

Let $\mathbf{x}$ be the state of the dynamic system, which is desired to be estimated, and let $\mathbf{y}$ be another random vector, sampleable, related to $\mathbf{x}$. Estimators are functions $\mathbf{g}(\mathbf{y})$ that infer the unknown value of $\mathbf{x}$ based on the known outcome of $\mathbf{y}$. Polynomial estimators are a subset of all estimators, which, using the Kronecker operator, can be written as

$$\mathbf{g}(\mathbf{y}) = \mathbf{a} + \mathbf{K}_1\mathbf{y} + \mathbf{K}_2\mathbf{y}^{[2]} + \mathbf{K}_3\mathbf{y}^{[3]} + \mathbf{K}_4\mathbf{y}^{[4]} + \cdots, \quad (3)$$

where $\mathbf{a}$ is a constant, each $\mathbf{K}_i$ is a constant matrix of appropriate dimensions, and each $\mathbf{y}^{[i]}$ is calculated using the Kronecker product

$$\mathbf{y}^{[i]} = \mathbf{y} \otimes \mathbf{y} \otimes \mathbf{y} \otimes \cdots. \quad (4)$$

In order to avoid redundancy, each repeated component of Equation (4) generated by the Kronecker product is eliminated, which means that, as an example, only one term between $y_i y_j$ and $y_j y_i$ is kept. It is convenient to derive the estimator's constants by working with deviation vectors. Deviation vectors are defined as

$$d\mathbf{x} = \mathbf{x} - \mathbb{E}\{\mathbf{x}\}, \quad (5)$$

$$d\mathbf{y}^{\{i\}} = \mathbf{y}^{[i]} - \mathbb{E}\left\{\mathbf{y}^{[i]}\right\}. \quad (6)$$

Deviations have zero mean by construction. The family of polynomial estimators defined by Equation (3) is redefined by adding and subtracting constants, in order to obtain a new, but theoretically equivalent, polynomial estimator family

$$\begin{aligned}\mathbf{g}(\mathbf{y}) &= \mathbf{a} + \mathbb{E}\{\mathbf{x}\} + \mathbf{K}_1(\mathbf{y} - \mathbb{E}\{\mathbf{y}\}) \\ &\quad + \mathbf{K}_2\left(\mathbf{y}^{[2]} - \mathbb{E}\left\{\mathbf{y}^{[2]}\right\}\right) \\ &\quad + \mathbf{K}_3\left(\mathbf{y}^{[3]} - \mathbb{E}\left\{\mathbf{y}^{[3]}\right\}\right) + \cdots \\ &= \mathbf{a} + \mathbb{E}\{\mathbf{x}\} + \mathbf{K}_1 d\mathbf{y} + \mathbf{K}_2\, d\mathbf{y}^{\{2\}} + \mathbf{K}_3\, d\mathbf{y}^{\{3\}} + \cdots \quad (7) \\ &= \mathbf{a} + \mathbb{E}\{\mathbf{x}\} + \mathcal{K} d\mathcal{Y}, \quad (8)\end{aligned}$$

where both the measurement residual with its powers, $d\mathcal{Y}$, and the matrices $\mathbf{K}_i$ are stacked:

$$\mathcal{K} = \begin{bmatrix}\mathbf{K}_1 & \mathbf{K}_2 & \mathbf{K}_3 & \dots\end{bmatrix}, \quad (9)$$

$$d\mathcal{Y} = \begin{bmatrix}d\mathbf{y}^T & d\mathbf{y}^{\{2\}T} & d\mathbf{y}^{\{3\}T} & \dots\end{bmatrix}^T. \quad (10)$$

The optimal estimator, in an MMSE sense, satisfies the orthogonality principle, from which it follows that the optimal polynomial update estimator becomes

$$\hat{\mathbf{x}} = \mathbb{E}\{\mathbf{x}\} + \mathbf{P}_{\mathbf{x}\mathcal{Y}}\mathbf{P}^{-1}_{\mathcal{Y}\mathcal{Y}} d\mathcal{Y}. \quad (11)$$

Matrices $\mathbf{P}_{\mathbf{x}\mathcal{Y}}$ and $\mathbf{P}_{\mathcal{Y}\mathcal{Y}}$ are the augmented state-measurement cross-covariance matrix and the augmented measurement covariance matrix, respectively. These matrices are constructed blockwise by using covariances $\mathbf{P}_{\mathbf{x}\mathbf{y}^{[j]}}$ and $\mathbf{P}_{\mathbf{y}^{[i]}\mathbf{y}^{[j]}}$, for any combination of i and j. As an example, $\mathbf{P}_{\mathbf{y}^{[3]}\mathbf{y}^{[4]}}$ indicates the covariance between the third-order measurement vector $\mathbf{y}^{[3]}$ and the fourth-order $\mathbf{y}^{[4]}$. 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 $\forall i, j \in \mathbb{N}_0$.

B. Differential Algebra

In this work, Gaussian random vectors undergo nonlinear (polynomial) transformations. The methodology used here to approximate these transformations is differential algebra (DA) via the Differential Algebra Core Engine (DACE2.0) software program. DA is used as a tool to implement the polynomial filter. Other approximations of nonlinear transformations are also possible but not considered here; ref. 23, for example, used the Unscented Transformation.

The theory of DA has been developed by Martin Berz in the late 1980s [7]. The DA framework is an algebra of Taylor polynomials. All functions are represented

through a matrix of coefficients and exponents rather than the classical representation with an array of floating point (FP) numbers. The DACE2.0 [27] software has a hard-coded library of the Taylor series expansion of elementary functions. As a consequence, derivatives are not computed numerically (e.g., finite differences), but evaluated directly from the Taylor polynomials. DA offers a way of working in a computer environment where the algebra of polynomials is endowed of composition of function, function inversions, explicit system solving, etc., as in the standard FP arithmetic.

DA has been proven to reduce computational costs in solving ordinary differential equations (ODEs) [28]. Once the maximum truncation order of the polynomial is selected, DA creates the Taylor polynomial expansion of the flow of ODEs as a function of the provided initial conditions. This approach can replace thousands of integrations with the computationally faster evaluation of the Taylor expansion [5]. As a result, the computational burden reduces considerably [42]. In the filtering problem, DA techniques have been used for the development of an efficient mapping of uncertainties [43] and for the evaluation of high-order moments [4]. Wittig *et al.* [45] developed a domain splitting technique that improves the state propagation when initial uncertainties are large by creating multiple polynomials.

The main concept of DA is that each function $f(\mathbf{x})$ can be expressed as a polynomial $p(\delta\mathbf{x})$, where the new variable $\delta\mathbf{x}$ is the deviation from the expansion center $\hat{\mathbf{x}}$. The polynomial $p(\delta\mathbf{x})$ is the Taylor series expansion of $f(\mathbf{x})$, centered at $\hat{\mathbf{x}}$, and truncated up to a user-selected order c.

For a detailed description of DA, its techniques, and how the DACE2.0 works in a computer environment, the reader is referred to the references.

III. THE STATE AND COVARIANCE ESTIMATION FILTER

A new filtering technique, based on a double polynomial estimator, is proposed in the DA framework. The double nature of the filter refers to the sequential estimation of the state and the covariance, where, at each time-step, the same measurement outcome is used twice to achieve matching between the conditioned state mean and its relative uncertainty spread.

Consider the generic dynamic system described by the following equations of motion and measurement equations:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + \mathbf{v}_k, \tag{12}$$

$$\mathbf{y}_{k+1} = \mathbf{h}(\mathbf{x}_{k+1}) + \mathbf{w}_{k+1}, \tag{13}$$

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 noises are assumed to be zero-mean Gaussians and uncorrelated, such that their distribution is fully described by the first two moments. For all discrete time indexes i and j

$$\mathbb{E}\left\{\mathbf{v}_i\mathbf{w}_j^T\right\} = 0, \tag{14}$$

$$\mathbb{E}\left\{\mathbf{v}_i\mathbf{v}_j^T\right\} = \mathbf{Q}_i\delta_{ij}, \tag{15}$$

$$\mathbb{E}\left\{\mathbf{w}_i\mathbf{w}_j^T\right\} = \mathbf{R}_i\delta_{ij}, \tag{16}$$

where $\mathbf{Q}_i$ is the process noise autocovariance function, while $\mathbf{R}_i$ is for the measurement noise. The initial condition of the state of the system is assumed to be Gaussian as well $\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$; however, for all other time-steps $k > 0$, the state distribution will be non-Gaussian due the nonlinearities in the dynamics.

The main result of this article, the State And Covariance Estimation Filter (SACE-c-η-μ), shares the prediction step with our previous work [38] and introduces a new update technique. The single distribution used in SACE-c-η-μ is expanded using Gaussian Multiple Models (GMMs) theory [30] to create the Multiple Models State And Covariance Estimation Filter (SACEMM-c-η-μ).

SACE-c-η-μ is composed of three different parts: the prediction, the state update, and the covariance update. The three integers c, η, and μ in SACE-c-η-μ refer to the tuning parameters of the filter: They are, respectively, the order of the Taylor polynomial approximation of $\mathbf{f}(\cdot)$ and $\mathbf{h}(\cdot)$, (c), the order of the state polynomial update, (η), and the order of the covariance polynomial update, (μ).

A. Prediction

At the beginning of each time-step, the state distribution is assumed to be Gaussian $\mathbf{x}_k \sim \mathcal{N}(\hat{\mathbf{x}}_\mathbf{k}, \mathbf{P}_k)$. The state can, therefore, be initialized in the DA framework as a first-order polynomial

$$\mathbf{x}_k = \mathbf{x}_k(\delta\mathbf{x}_k) = \hat{\mathbf{x}}_\mathbf{k} + \mathbf{S}_k\delta\mathbf{x}_k, \tag{17}$$

where $\mathbf{S}_k\mathbf{S}_k^T = \mathbf{P}_k$, and the DA variable $\delta\mathbf{x}_k = \mathbf{x}_k - \hat{\mathbf{x}}_\mathbf{k}$ expresses the deviation from the expansion center, and it is interpreted as a Gaussian with zero mean and identity covariance matrix. Therefore, matrix $\mathbf{S}_k$ (here calculated through Cholesky Decomposition) scales the coefficients of the state polynomial and results in the moments of $\mathbf{x}_k$ easily calculated from the moments of $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

The propagation function is applied directly to the state polynomial, such that the predicted state vector is

$$\mathbf{x}_{k+1}^- = \mathbf{x}_{k+1}^-(\delta\mathbf{x}_k) = \mathbf{f}\big(\mathbf{x}_k(\delta\mathbf{x}_k)\big), \tag{18}$$

where $\mathbf{x}_{k+1}^-$ indicates the Taylor series expansion of the dynamics centered at $\hat{\mathbf{x}}_k$ and truncated at the user-defined integer order c. Equation (18) is carried out in

the DA framework. Each component of $\mathbf{x}_{k+1}^{-}$ is a polynomial map (centered at $\hat{\mathbf{x}}_k$) that maps deviations ($\delta\mathbf{x}_k$) from time-step k to time-step $k+1$ and describes how the state PDF evolves in time. The predicted polynomials are lacking the influence of the process noise. Process noise can be mapped in the DA framework with the same representation reserved for the state of the system. Thus, a new DA variable $\delta\mathbf{v}_k$, interpreted again as a standard normal random vector, is introduced:

$$\mathbf{x}_{k+1}^{-}(\delta\mathbf{x}_k, \delta\mathbf{v}_k) := \mathbf{x}_{k+1}^{-}(\delta\mathbf{x}_k) + \mathbf{T}_k\delta\mathbf{v}_k, \tag{19}$$

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

Analogously, the predicted measurement is expressed as a Taylor polynomial expansion in the DA framework:

$$\mathbf{y}_{k+1} = \mathbf{y}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k) = \mathbf{h}\big(\mathbf{x}_{k+1}^{-}(\delta\mathbf{x}_k, \delta\mathbf{v}_k)\big), \tag{20}$$

where $\mathbf{y}_{k+1}$ is, again, a polynomial centered at $\hat{\mathbf{x}}_k$ with maximum order c. In Equation (20), the expansion is now with respect to both the state deviation vector ($\delta\mathbf{x}_k$) and the process noise ($\delta\mathbf{v}_k$). The influence of the measurement noise is added to the polynomials like in Equation (19). A new DA variable $\delta\mathbf{w}_{k+1}$ is introduced

$$\mathbf{y}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) := \mathbf{y}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k) + \mathbf{U}_{k+1}\delta\mathbf{w}_{k+1}, \tag{21}$$

where $\mathbf{w}_k = \mathbf{U}_k\delta\mathbf{w}_k$ and $\mathbf{U}_k\mathbf{U}_k^T = \mathbf{R}_k$. Once again, $\delta\mathbf{w}_{k+1}$ is interpreted as a standard normal random vector.

All the predicted quantities have been calculated, and they are represented as polynomial functions of standard random vectors. The number of variables is $2n+m$: n deviations map the state behavior, n map the process noise, and the remaining m map the measurement noise. The Gaussian nature of the random vectors leads to a fast evaluation of all expectation operations since, for a Gaussian PDF, central moments can be easily computed using Isserlis' formulation [17].

B. The State Polynomial Update

The second part of SACE-c-η-μ is the state polynomial update. After selecting the integer c in the prediction step, the user defines a second integer, η, which selects the order of the polynomial estimator dedicated to the state of the system.

The polynomial update evaluates the augmented Kalman gain and for high powers of the measurement polynomials. Starting from the latter,

$$\mathbf{y}_{k+1}^{[2]} = \mathbf{y}_{k+1} \otimes \mathbf{y}_{k+1}, \tag{22}$$

$$\mathbf{y}_{k+1}^{[i]} = \mathbf{y}_{k+1} \otimes \mathbf{y}_{k+1} \otimes \dots \tag{23}$$

with $i = 1, \dots, \eta$, and, once again, the redundant components are eliminated, in order to have independent measurements.

The means of the predicted state polynomials are now evaluated. Each polynomial undergoes the expectation operator, which, being a linear operator, works directly on the single monomials of the expansion [34].

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	...

$$\hat{\mathbf{x}}^{-} = \mathbb{E}\left\{\mathbf{x}_{k+1}^{-}\right\}. \tag{24}$$

The deviations have a Gaussian distribution with zero mean and identity covariance matrix; therefore, the expected value substitutes the relative Isserlis' moment in for each monomial, according to Table I.

For example: $\mathbb{E}\left\{\alpha\delta x_1^8\delta x_2^4\delta x_4^6\delta v_2^2\delta w_3^4\right\} = 4725\alpha$. The predicted means of the measurement polynomials are similarly evaluated using Equation (24):

$$\hat{\mathbf{y}}_{k+1} = \mathbb{E}\left\{\mathbf{y}_{k+1}\right\}, \tag{25}$$

$$\hat{\mathbf{y}}_{k+1}^{[2]} = \mathbb{E}\left\{\mathbf{y}_{k+1}^{[2]}\right\}, \tag{26}$$

$$\hat{\mathbf{y}}_{k+1}^{[i]} = \mathbb{E}\left\{\mathbf{y}_{k+1}^{[i]}\right\}, \tag{27}$$

where, once again, $i = 1, \dots, \eta$.

The augmented measurement covariance $\mathbf{P}_{\mathcal{Y}\mathcal{Y},[\eta]}$ is evaluated blockwise. The matrix is guaranteed to be nonsingular because redundant rows and columns have been eliminated. The matrix is symmetric and each block is evaluated as

$$\mathbf{P}_{\mathbf{y}^{[i]}\mathbf{y}^{[j]}} = \mathbb{E}\left\{\left(\mathbf{y}_{k+1}^{[i]} - \hat{\mathbf{y}}_{k+1}^{[i]}\right)\left(\mathbf{y}_{k+1}^{[j]} - \hat{\mathbf{y}}_{k+1}^{[j]}\right)^{\mathrm{T}}\right\}, \tag{28}$$

$\forall i, j = 1, \dots, \eta$. Every time a polynomial multiplies itself, the maximum truncation order of the Taylor series doubles. For example, the evaluation of $\mathbf{P}_{\mathbf{y}^{[5]}\mathbf{y}^{[3]}}$ applies the expectation operator to a polynomial with monomials up to order $8c$. The augmented state-measurement cross-covariance matrix $\mathbf{P}_{\mathbf{x}\mathcal{Y},[\eta]}$ is evaluated blockwise, and each block is evaluated as

$$\mathbf{P}_{\mathbf{x}\mathbf{y}^{[i]}} = \mathbb{E}\left\{(\mathbf{x}_{k+1}^{-} - \hat{\mathbf{x}}_{k+1}^{-})\left(\mathbf{y}_{k+1}^{[i]} - \hat{\mathbf{y}}_{k+1}^{[i]}\right)^{\mathrm{T}}\right\}, \tag{29}$$

$\forall i = 1, \dots, \eta$. The subscript $[\eta]$ specifies that the covariance matrices are created with measurement powers up to order η. From these covariances, it is now possible to evaluate the augmented Kalman gain

$$\mathcal{K} = \mathbf{P}_{\mathbf{x}\mathcal{Y},[\eta]}\mathbf{P}_{\mathcal{Y}\mathcal{Y},[\eta]}^{-1}. \tag{30}$$

Denote with $\tilde{\mathbf{y}}_{k+1}$ the numerical outcome of the random vector $\mathbf{y}_{k+1}$, its powers are evaluated using the Kronecker product

$$\tilde{\mathbf{y}}_{k+1}^{[2]} = \tilde{\mathbf{y}}_{k+1} \otimes \tilde{\mathbf{y}}_{k+1}, \tag{31}$$

$$\tilde{\mathbf{y}}_{k+1}^{[i]} = \tilde{\mathbf{y}}_{k+1} \otimes \tilde{\mathbf{y}}_{k+1} \otimes \cdots \tag{32}$$

with $i = 1, \ldots, \eta$, and, once again, the redundant components are eliminated. The polynomial update exploits the influence of high powers from the measurement outcome. The measurement residual is developed to create the augmented innovation vector

$$d\tilde{\mathcal{Y}}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) = \begin{bmatrix} \tilde{\mathbf{y}}_{k+1} - \mathbf{y}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) \\ \tilde{\mathbf{y}}^{[2]}_{k+1} - \mathbf{y}^{[2]}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) \\ \cdots \\ \tilde{\mathbf{y}}^{[\eta]}_{k+1} - \mathbf{y}^{[\eta]}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) \end{bmatrix}. \tag{33}$$

The updated distribution (polynomial) of the state is given by

$$\begin{aligned} &\mathbf{x}^{+}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) \\ &\quad = \mathbf{x}^{-}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k) + \mathcal{K} d\tilde{\mathcal{Y}}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}), \end{aligned} \tag{34}$$

and the posterior estimate is its mean

$$\hat{\mathbf{x}}^{+}_{k+1} = \mathbb{E}\left\{\mathbf{x}^{+}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1})\right\} \tag{35}$$

evaluated, through Isserlis's moments, monomial by monomial using Table I.

Equation (34) shows that the state polynomials are a function of the three different deviations: the state deviation, the process noise, and the measurement noise. Furthermore, the new order of the polynomial is increased by a factor η, dictated by the order of the polynomial update. If the order of the polynomial approximation of the prior distribution ($\mathbf{x}^{-}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k)$) is c, then the order of the posterior polynomial ($\mathbf{x}^{+}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1})$) is ηc. The higher the polynomial order, the higher the number of moments to be calculated by Table I, which leads to a higher computational burden.

C. The Covariance Polynomial Update

The third, and last, part of SACE-c-η-μ is the covariance polynomial update. After having estimated the state of the system, SACE-c-η-μ applies a second polynomial estimator to identify the value of the state covariance conditioned to the measurements. Therefore, the user defines one last integer parameter, μ, that specifies the order of the covariance polynomial update. Unlike previous tuning parameters, μ cannot be freely chosen, but it has to respect the inequality $\mu < \eta$. The covariance cannot have an higher update order than the state.

The covariance matrix is obtained as

$$\mathbf{P}_{\mathbf{xx},k+1} = \mathbb{E}\left\{(\mathbf{x}^{+}_{k+1} - \hat{\mathbf{x}}^{+}_{k+1})(\mathbf{x}^{+}_{k+1} - \hat{\mathbf{x}}^{+}_{k+1})^{\mathrm{T}}\right\}. \tag{36}$$

This value shows the average spread of the posterior distribution among all different possible outcomes, $\tilde{\mathbf{y}}$, of the random variable $\mathbf{y}$. Equation (36) is the equivalent of the classical covariance update formulation, Equation (2), that is used in the most common filters such as EKF, UKF, QKF, CBF, Central Difference Filter, and GSOF. Therefore, even if correct, using the average error covariance does not extract all the possible information from the measurement outcome. Similar to the polynomial formulation for estimating the state presented in Equation (7), Equation (36) can be seen as a zeroth-order polynomial estimator of the covariance matrix.

A new approach is, therefore, presented in which the estimate of the covariance is performed to order higher than zero. Define a polynomial vector, $\boldsymbol{\rho}_{k+1}$, as the covariance polynomial

$$\boldsymbol{\rho}^{-}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1}) = (\mathbf{x}^{+}_{k+1} - \hat{\mathbf{x}}^{+}_{k+1}) \otimes (\mathbf{x}^{+}_{k+1} - \hat{\mathbf{x}}^{+}_{k+1}), \tag{37}$$

where, in order to reduce the computational burden, the redundant terms of the symmetric covariance matrix have been eliminated, e.g., the upper diagonal terms are removed. The covariance polynomial maximum order is $2\eta c$, being the square of the posterior distribution. The mean of $\boldsymbol{\rho}_{k+1}$ is exactly the vectorized version of the covariance matrix expressed in Equation (36):

$$\hat{\boldsymbol{\rho}}^{-}_{k+1} = \mathbb{E}\left\{\boldsymbol{\rho}^{-}_{k+1}(\delta\mathbf{x}_k, \delta\mathbf{v}_k, \delta\mathbf{w}_{k+1})\right\}, \tag{38}$$

$$= \mathrm{stack}(\mathbf{P}_{\mathbf{xx},k+1}), \tag{39}$$

where the stack() operator indicates the vectorization of a matrix, performed by stacking columns on top of each other. The covariance update is treated in the same manner as the state vector: adding to a known prior a polynomial function of the measurement outcome $\tilde{\mathbf{y}}_{k+1}$. This second polynomial update provides an updated covariance value that better represents the state estimate's uncertainty.

The starting point is the already computed augmented measurement covariance matrix $\mathbf{P}_{\mathcal{Y}\mathcal{Y},[\mu]}$. The constrain $\mu < \eta$ makes $\mathbf{P}_{\mathcal{Y}\mathcal{Y},[\mu]}$ a subset of $\mathbf{P}_{\mathcal{Y}\mathcal{Y},[\eta]}$, obtained by selecting the first μ rows and columns. The cross-covariance matrix $\mathbf{P}_{\rho\mathcal{Y},[\mu]}$ is evaluated block-wise:

$$\mathbf{P}_{\rho\mathcal{Y},[\mu]} = \left[\mathbf{P}_{\rho\mathbf{y}}\ \mathbf{P}_{\rho\mathbf{y}^{[2]}}\ \mathbf{P}_{\rho\mathbf{y}^{[3]}}\ \cdots\right], \tag{40}$$

similarly to $\mathbf{P}_{\mathbf{x}\mathcal{Y},[\eta]}$. Each block is obtained as

$$\mathbf{P}_{\rho\mathbf{y}^{[i]}} = \mathbb{E}\left\{(\rho^{-}_{k+1} - \hat{\rho}^{-}_{k+1})\left(\mathbf{y}^{[i]}_{k+1} - \hat{\mathbf{y}}^{[i]}_{k+1}\right)^{\mathrm{T}}\right\} \tag{41}$$

with $i = 1, \ldots, \mu$. The Kalman gain associated with the covariance correction is calculated as

$$\mathcal{G} = \mathbf{P}_{\rho\mathcal{Y},[\mu]}\mathbf{P}^{-1}_{\mathcal{Y}\mathcal{Y},[\mu]}. \tag{42}$$

The covariance is updated to its posterior estimate as

$$\hat{\boldsymbol{\rho}}^{+}_{k+1} = \hat{\boldsymbol{\rho}}^{-}_{k+1} + \mathcal{G}\begin{bmatrix} \tilde{\mathbf{y}}_{k+1} - \hat{\mathbf{y}}_{k+1} \\ \tilde{\mathbf{y}}^{[2]}_{k+1} - \hat{\mathbf{y}}^{[2]}_{k+1} \\ \cdots \\ \tilde{\mathbf{y}}^{[\mu]}_{k+1} - \hat{\mathbf{y}}^{[\mu]}_{k+1} \end{bmatrix}, \tag{43}$$

where the influence of the measurement is weighted by the augmented Kalman gain. Before starting the next iteration, vector $\hat{\boldsymbol{\rho}}^{+}_{k+1}$ is brought back to its matrix formulation

$$\hat{\mathbf{P}}_{\mathbf{xx},k+1} = \mathrm{matrix}(\hat{\boldsymbol{\rho}}^{+}_{k+1}), \tag{44}$$

where the matrix() operator is the inverse of the stack() operator.

The updated posterior distribution can be approximated as Gaussian with mean $\hat{\mathbf{x}}^{+}_{k+1}$ and covariance matrix $\hat{\mathbf{P}}_{\mathbf{xx},k+1}$ to start the next iteration from Equation (17), where the DA variables related to the noises are discarded and a new state deviation vector is initialized.

SACE-c-η-μ contains three tuning parameters to enhance the performance of classic estimators. In fact, SACE-1-1-0 reduces to the EKF, and SACE-2-1-0 is the GSOF. The polynomial estimator better weights the information from the measurement by computing high-order central moments. The increase in accuracy is paid by an increase in computational effort, which practically limits the filter's order selection. The highest polynomial order the filter has to compute (in the evaluation of $\mathbf{P}_{\rho\mathbf{y}^{[\mu]}}$) is $(2\eta+\mu)c$.

The computational time required by the filter depends on the selection of its three tuning parameters and on the dimension of the state vector. SACE-c-η-μ is not suitable for extremely large systems because of the exponential grow in the number of monomials in the Taylor expansion [9]. An in-depth analysis of the computational time of filters developed in the DACE2.0 framework is presented in [12]. The reference portraits an exhaustive analysis of the execution time on the BeagleBone Black (BBB) Single Board Computer, with particular focus on the duty cycles of filter execution on BBB and its dependency on the Taylor truncation order.

IV. THE MULTIPLE MODELS SPACE AND COVARIANCE ESTIMATION FILTER

SACE-c-η-μ approximates the time propagation of the state with one single polynomial representation of the flow. However, as the Taylor polynomial series gets farther away from the expansion center, it becomes less accurate. Therefore, when the initial uncertainties of the state distribution are extremely large, a single polynomial map may not be sufficient to truthfully describe the predicted PDF [45]. Splitting the initial uncertainties in multiple (smaller) subdomains aids the filter in reaching convergence. Thus, a second filter called SACEMM-c-η-μ merges SACE-c-η-μ with the GMM formulation. In the DA framework, multiple models translate into multiple polynomials.

A. Initialization

The initial state distribution is assumed to be Gaussian $\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_\mathbf{0}, \mathbf{P}_0)$. The initialization of the models follows an analogy with the unscented transformation [20]. Therefore, the initial domain is divided into $\theta = 2n+1$ models, where n is the number of states. Each ith model is a Gaussian with mean $\hat{\mathbf{x}}_{0,\{i\}}$ and covariance $\mathbf{P}_{0,\{i\}}$. Being symmetric, the state covariance matrix can be elaborated into its eigenvalue decomposition

$$\mathbf{P}_0 = \mathbf{V}\mathbf{D}\mathbf{V}^T, \tag{45}$$

where $\mathbf{V}$ is the matrix of eigenvectors that describes the orientation of the uncertainty ellipsoids, and the diagonal matrix of eigenvalues $\mathbf{D}$ describes the magnitude of the uncertainties. The mean of each Gaussian kernel is selected as

$$\hat{\mathbf{x}}_{0,\{0\}} = \hat{\mathbf{x}}_0, \tag{46}$$

$$\hat{\mathbf{x}}_{0,\{j\}} = \hat{\mathbf{x}}_0 + \mathbf{V}\mathbf{D}_j, \qquad j = 1,\dots,n, \tag{47}$$

$$\hat{\mathbf{x}}_{0,\{j\}} = \hat{\mathbf{x}}_0 + \mathbf{V}\mathbf{D}_{j-n}, \quad j = n+1,\dots,2n, \tag{48}$$

where $\mathbf{D}_j$ indicates the jth column of the matrix. The centers of the models lie on the principal axes and their initial weights are proportional to their probability with respect to the initial distribution

$$\omega_{0,\{i\}} = \frac{(2\pi)^{-n/2}}{\mathcal{W}_0\sqrt{\det\mathbf{P}_0}} \exp\left(-\frac{1}{2}\left(\hat{\mathbf{x}}_{0,\{i\}}-\hat{\mathbf{x}}_0\right)^T\mathbf{P}_0^{-1}\left(\hat{\mathbf{x}}_{0,\{i\}}-\hat{\mathbf{x}}_0\right)\right), \tag{49}$$

$$\mathcal{W}_0 = \sum_{i=0}^{\theta-1}\omega_{0,\{i\}}, \tag{50}$$

where $\mathcal{W}_0$ normalizes the weights such that their sum is unity. The models are assumed to share the same covariance, and they all have the same initial level of uncertainty

$$\mathbf{P}_{0,\{j\}} = \mathbf{P}_0 + \mathbf{x}_0\mathbf{x}_0^T - \sum_{i=0}^{\theta-1}\omega_{0,\{i\}}\mathbf{x}_{0,\{i\}}\mathbf{x}_{0,\{i\}}^T \tag{51}$$

with $j = 0,\dots,\theta-1$. Therefore, at the beginning of the first iteration, the initial Gaussian distribution has been divided into θ smaller Gaussian kernels $\mathbf{x}_{0,\{i\}} \sim \mathcal{N}(\hat{\mathbf{x}}_{0,\{i\}}, \mathbf{P}_{0,\{i\}})$ with the same covariance matrix and means on the principal axes, selected as sigma points from the unscented transformation.

B. Prediction

The models have been initialized as Gaussian kernels. SACEMM-c-η-μ applies SACE-c-η-μ on each kernel like it were operating by its own. As a consequence, θ different polynomials are created in the DA frameworks, and θ polynomial maps of the flow describe the time propagation of the state.

$$\mathbf{P}_{k,\{i\}} = \mathbf{S}_{k,\{i\}}\mathbf{S}_{k,\{i\}}^T, \tag{52}$$

$$\delta\mathbf{x}_{k,\{i\}} = \mathbf{x}_{k,\{i\}} - \hat{\mathbf{x}}_{k,\{i\}}, \tag{53}$$

$$\mathbf{x}_{k,\{i\}} = \mathbf{x}_{k,\{i\}}(\delta\mathbf{x}_{k,\{i\}}) = \hat{\mathbf{x}}_{k,\{i\}} + \mathbf{S}_{k,\{i\}}\delta\mathbf{x}_{k,\{i\}}, \tag{54}$$

$$\mathbf{x}^{-}_{k+1,\{i\}} = \mathbf{x}^{-}_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}) = \mathbf{f}\big(\mathbf{x}_{k,\{i\}}(\delta\mathbf{x}_{k,\{i\}})\big), \tag{55}$$

with $i = 0,\dots,\theta-1$. Multiple Taylor series expansions improve the approximation accuracy of the polynomial

maps since, at the boundaries, deviations are closer to their relative centers. Following SACE-c-η-μ for each model, the process noise is mapped on each polynomial expansion:

$$\mathbf{v}_{k,\{i\}} = \mathbf{T}_k \delta\mathbf{v}_{k,\{i\}}, \tag{56}$$

$$\mathbf{x}^-_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}, \delta\mathbf{v}_{k,\{i\}}) := \mathbf{x}^-_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}) + \mathbf{T}_k \delta\mathbf{v}_{k,\{i\}}, \tag{57}$$

and a measurement polynomial is evaluated for each kernel:

$$\mathbf{w}_{k+1,\{i\}} = \mathbf{U}_{k+1} \delta\mathbf{w}_{k+1,\{i\}}, \tag{58}$$

$$\begin{aligned}\mathbf{y}_{k+1,\{i\}} &= \mathbf{y}_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}, \delta\mathbf{v}_{k,\{i\}}),\\ &= \mathbf{h}\big(\mathbf{x}^-_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}, \delta\mathbf{v}_{k,\{i\}})\big).\end{aligned} \tag{59}$$

Measurement noise is added in the DA framework

$$\begin{aligned}&\mathbf{y}_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}, \delta\mathbf{v}_{k,\{i\}}, \delta\mathbf{w}_{k+1,\{i\}}) :=\\ &\quad \mathbf{y}_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}, \delta\mathbf{v}_{k,\{i\}}) + \mathbf{U}_{k+1}\delta\mathbf{w}_{k+1,\{i\}}\end{aligned} \tag{60}$$

such that the prediction step is completed for each Gaussian kernel.

C. The State and Covariance Polynomial Update

The prediction step has been exploited by the introduction of multiple polynomials. In the update step, each kernel undergoes the polynomial update for the state and for the covariance described by SACE-c-η-μ. Therefore, after having selected η and μ as the orders for the polynomial estimators, the state posterior estimate and the conditional covariance of each model are evaluated as

$$\mathbf{x}^+_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}, \delta\mathbf{v}_{k,\{i\}}, \delta\mathbf{w}_{k+1,\{i\}}) = \mathbf{x}^-_{k+1,\{i\}} + \mathcal{K}_{\{i\}} d\tilde{\mathcal{Y}}_{\{i\}}, \tag{61}$$

$$\hat{\mathbf{x}}^+_{k+1,\{i\}} = \mathbb{E}\left\{\mathbf{x}^+_{k+1,\{i\}}(\delta\mathbf{x}_{k,\{i\}}, \delta\mathbf{v}_{k,\{i\}}, \delta\mathbf{w}_{k+1,\{i\}}\right\}, \tag{62}$$

and

$$\hat{\boldsymbol{\rho}}^+_{k+1,\{i\}} = \hat{\boldsymbol{\rho}}^-_{k+1,\{i\}} + \mathcal{G}_{\{i\}} \begin{bmatrix} \tilde{\mathbf{y}}_{k+1,\{i\}} - \hat{\mathbf{y}}_{k+1,\{i\}} \\ \tilde{\mathbf{y}}^{[2]}_{k+1,\{i\}} - \hat{\mathbf{y}}^{[2]}_{k+1,\{i\}} \\ \cdots \\ \tilde{\mathbf{y}}^{[\mu]}_{k+1,\{i\}} - \hat{\mathbf{y}}^{[\mu]}_{k+1,\{i\}} \end{bmatrix}, \tag{63}$$

$$\hat{\mathbf{P}}_{\mathbf{xx},k+1,\{i\}} = \text{matrix}(\hat{\boldsymbol{\rho}}^+_{k+1,\{i\}}), \tag{64}$$

with $i = 0, \ldots, \theta - 1$. Every Kalman gain and expectation has been calculated according to the polynomial estimator theory and using Table I, since each deviation is interpreted as a standard normal random vector.

The influence of each ith Gaussian to the posterior PDF needs to be updated as well. The posterior distribution of the probability of each Gaussian given the measurements can be evaluated using Bayes' rule. Therefore, the updated weight of each model is proportional to its measurement likelihood. Let us define with $P(\mathbf{y}_{k+1}|i, \mathbf{Y}_k)$ the probability of $\tilde{\mathbf{y}}_{k+1}$ to be the outcome from the ith Gaussian:

$$\begin{aligned}P(\tilde{\mathbf{y}}_{k+1}|i, \mathbf{Y}_k) &= \frac{(2\pi)^{-m/2}}{\sqrt{\det \mathbf{P}_{\mathbf{yy},\{i\}}}}\\ &\exp\left(-\frac{1}{2}\left(\tilde{\mathbf{y}}_{k+1} - \hat{\mathbf{y}}_{k+1,\{i\}}\right)\mathbf{P}^{-1}_{\mathbf{yy},\{i\}}\left(\tilde{\mathbf{y}}_{k+1} - \hat{\mathbf{y}}_{k+1,\{i\}}\right)\right),\end{aligned} \tag{65}$$

where $\mathbf{Y}_k$ indicates all the measurement realizations up to time-step k. The weight update formulation is derived, for the ith kernel, as

$$\begin{aligned}\omega_{k+1,\{i\}} &= P(i|\mathbf{Y}_{k+1})\\ &= P(i|\tilde{\mathbf{y}}_{k+1}, \mathbf{Y}_k)\\ &= \frac{P(i, \tilde{\mathbf{y}}_{k+1}|\mathbf{Y}_k)}{P(\tilde{\mathbf{y}}_{k+1}|\mathbf{Y}_k)}\\ &= \frac{P(i, \tilde{\mathbf{y}}_{k+1}|\mathbf{Y}_k)}{\sum_{j=0}^{\theta-1} P(j, \tilde{\mathbf{y}}_{k+1}|\mathbf{Y}_k)}\\ &= \frac{P(\tilde{\mathbf{y}}_{k+1}|i, \mathbf{Y}_k)P(i|\mathbf{Y}_k)}{\sum_{j=0}^{\theta-1} P(j, \tilde{\mathbf{y}}_{k+1}|\mathbf{Y}_k)}\\ &= \frac{P(\tilde{\mathbf{y}}_{k+1}|i, \mathbf{Y}_k)}{\sum_{j=0}^{\theta-1} \omega_{k,\{j\}} P(\tilde{\mathbf{y}}_{k+1}|j, \mathbf{Y}_k)}\omega_{k,\{i\}},\end{aligned} \tag{66}$$

where the denominator normalizes the weights such that they sum to unity. Equation (66) is recursive and modifies the importance of each model based on how likelihood it could have generated the measurement outcome.

The filtering algorithm has ended, and it can start the following iteration from $\hat{\mathbf{x}}^+_{k+1,\{i\}}$, $\hat{\mathbf{P}}_{\mathbf{xx},k+1,\{i\}}$, and $\omega_{k+1,\{i\}}$ for each model. However, the weighted state estimate, $\bar{\mathbf{x}}$, and covariance, $\bar{\mathbf{P}}$, are calculated for downstream users, and they are used to assess the performance of the filtering technique.

$$\bar{\mathbf{x}} = \sum_{i=0}^{\theta} \omega_{k+1,\{i\}} \hat{\mathbf{x}}^+_{k+1,\{i\}}, \tag{67}$$

$$\bar{\mathbf{P}} = -\bar{\mathbf{x}}\bar{\mathbf{x}}^T + \sum_{i=0}^{\theta} \omega_{k+1,\{i\}} \left(\hat{\mathbf{P}}_{\mathbf{xx},k+1,\{i\}} + \hat{\mathbf{x}}^+_{k+1,\{i\}}\hat{\mathbf{x}}^{+T}_{k+1,\{i\}}\right). \tag{68}$$

Once again, for basic parameters, SACEMM-c-η-μ reduces to well-known filters: In fact, picking SACEMM-1-1-0 reduces to the GSF. The computational complexity of SACEMM-c-η-μ is approximately θ times bigger when compared to SACE-c-η-μ. Therefore, it is advised to operate the multiple-model technique when the initial state uncertainties are particularly large, or when the time-step is long enough that one polynomial approximation is not sufficient to adequately represent the flow of the dynamics. Therefore, for problems with high initial uncertainty, SACEMM-c-η-μ can be used for the first few iteration steps and then replaced with SACE-c-η-μ once the sate error covariance has decreased.

V. NUMERICAL EXAMPLES

The proposed filtering techniques have been applied to three different scenarios. First, a scalar application gives a visual representation of how the new update algorithm works and highlights the innovative features as compared to other estimators. The second problem consists of a tracking application where the system undergoes the highly nonlinear dynamics of a Lorenz96 system. The third application uses Lorenz63 dynamics to underline the benefits of the multiple model filtering technique.

A. Scalar Problem

A simple scalar problem is presented here to highlight the improvements of the new filtering technique by estimating the conditional covariance. It has already been proven that high-order polynomial estimators are a better approximation of the true MMSE [38]. However, the presented example underlines the matching between state and covariance for each different realization of the measurement.

Define a normal prior state distribution $x \sim \mathcal{N}(1, 0.02)$ and a measurement

$$y = 1/x + \nu, \tag{69}$$

where $\nu \sim \mathcal{N}(0, 0.003)$ is independent of x and represents the measurement noise.

Fig. 1 shows the true joint distribution of x and y represented using 10^5 points (gray dots in the figure). The figure compares SACE-c-η-μ and SACEMM-c-η-μ with a few common estimators: the EKF, the UKF, the GSF, the Iterated Extended Kalamn Filter (IEKF) [6], the PF, and the high-order EKF (DAHO-k) [43]. The first row of graphs (EKF, UKF, DAHO-3) contains linear estimators; therefore, their representation on the (x, y) plane is a straight line, shown in red. The slope of the red line is the Kalman gain, whose optimal value is $\mathbf{P_{xy}}\mathbf{P_{yy}}^{-1}$. The different slopes shown by the different linear estimators

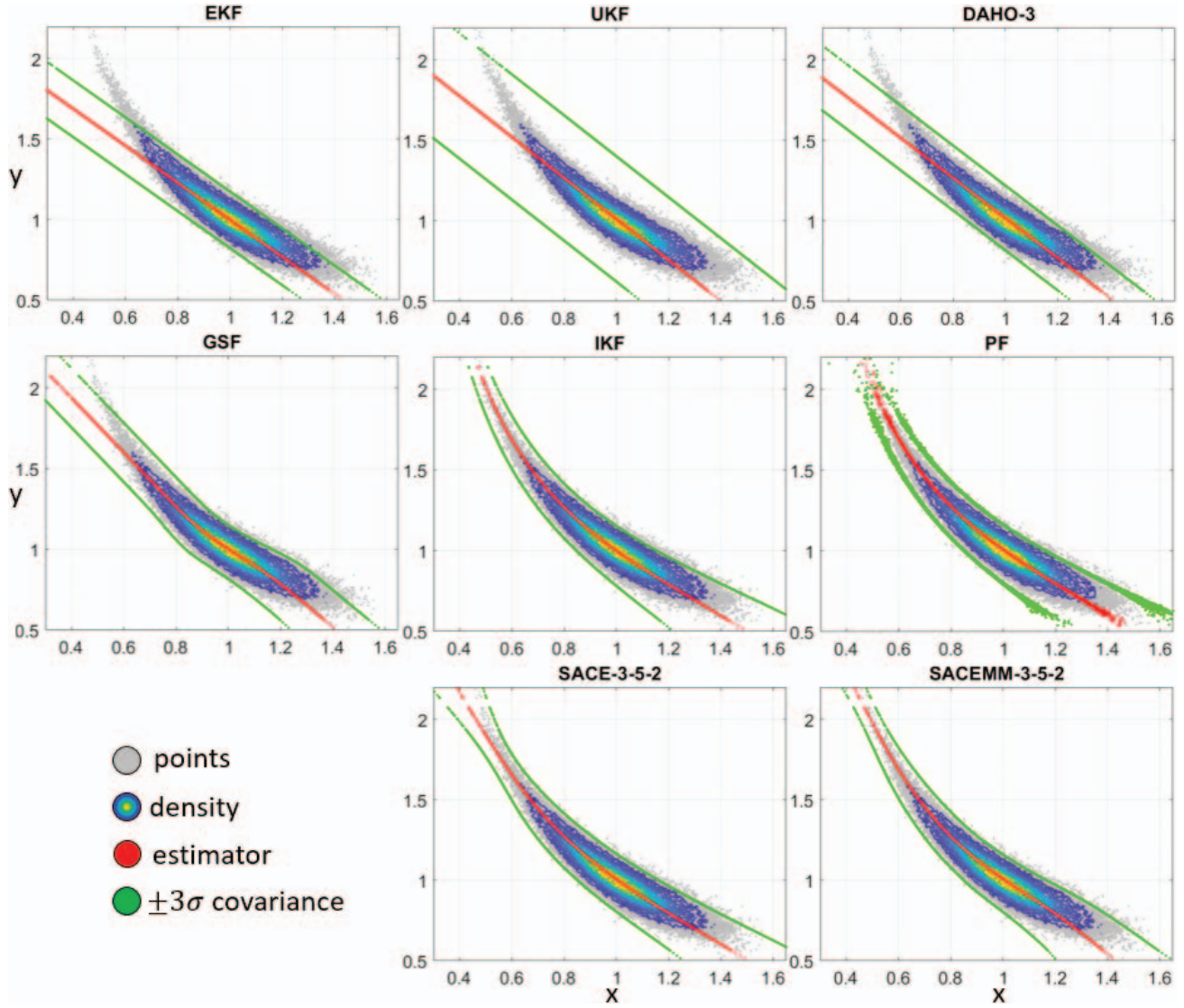

Fig. 1. Comparison among different estimators. Posterior distribution (gray), the estimator functions (red), and their confidence levels (green).

are due to the different approximations each linear filter employs to evaluate the moments. The EKF applies basic linearization (Jacobians), the UKF uses the unscented transformation, and DAHO-3 uses Taylor polynomials up to the third order. The green lines depict the filter's own assessment of the estimation error uncertainty as a $\pm 3\sigma$ boundary. The different evaluations of the moments lead to different values on the estimation of the variance, as it follows Equation (2). The green lines share the same slope of the corresponding red line: They are just translated left (and right) by 3σ. These linear filters estimate the same uncertainty level regardless of the measurement outcome, and the predicted covariance value is the mean among all the possible different realizations.

The second row of graphs in Fig. 1 shows nonlinear estimators. The GSF has been implemented with three models, which allows the estimator function, red line, to follow the curved shape of the posterior distribution. However, when the likelihood of one model becomes predominant with respect to the others, the GSF behaves similarly to the EKF: This aspect is mostly evident near the tails of the distribution. The estimated covariance of the GSF is a function of the measurement because it is evaluated as a weighted mean among all the models, whose importance weight is based on their likelihood. However, the $\pm 3\sigma$ green lines show the same problems of linear estimators: The lines are able to change slope when the models have approximately the same weight; otherwise, they are straight. Furthermore, since the GSF can be intended as multiple EKFs with reduced subdomains, the filter shows the same behavior of the linear estimator at the edges of the posterior PDF. The IEKF performs multiple updates to repeatedly calculate the measurement Jacobian each time linearizing with respect to the most current estimate. The IEKF minimizes a nonlinear least-squares performance index that, for appropriate probability distributions functions, approximates the maximum *a posteriori* (MAP) estimate. As such, the IEKF is a nonlinear estimator, and its red line follows the bend of the posterior distribution, setting on the most likely value of x for each measurement outcome y. The $\pm 3\sigma$ green lines correctly bound the distribution; however, the IEKF is not necessary an unbiased filter, and choosing the peak of the posterior distribution does not necessarily minimize the MSE. Hence, the IEKF's MSE is often larger than filters based on the MMSE principle [37]. The third nonlinear estimator presented in Fig. 1 is the PF. PFs are accurate nonlinear estimators that use an ensemble of weighted particles to calculate the state estimate. The weight of each particle depends on its measurement likelihood. Both the state estimate and the predicted error covariance are (nonlinear) functions of the measurements. The graph shows that the PF estimates do not form well-defined lines, but the state and covariance estimate values depend on the randomness of the data. In other words, while in the EKF the state estimates from two separate updates with the same measurement outcome give exactly the same value, two PF estimates depend on the randomness of the initial ensemble used to generate them. Consequently, the green and red "lines" of the PF become thicker while moving towards the tails of the posterior distribution.

In the third row, SACE-3-5-2 and SACEMM-3-5-2 are reported. The fifth-order polynomial estimator is able to follow the curved shape of the joint distribution, and it accurately approximates the true MMSE. The optimal MMSE is the conditional mean, which visually is the line that divides in half the distribution of y, as horizontal spread of points, for each value of x. Therefore, while EKF, UKF, and DAHO-3 can be interpreted as different linear approximations of the true MMSE, SACE-3-5-2 represents a fifth-order approximation, which shows a more accurate result. By increasing the estimator order η to infinity, SACE-c-η-μ would asymptotically reach the true MMSE. The green lines related to SACE-3-5-2 show how the uncertainty level has become a (nonlinear) function of the measurement. The $\pm 3\sigma$ boundary increases and tightens depending on the horizontal spread of samples around the estimator function. For example, when the current measurement is $y = 1$, SACE-3-5-2 gives its estimate with a level of uncertainty that matches the spread of the gray points on the line $y = 1$. When the sensor gives $y = 2$, SACE-3-5-2 outputs a level of confidence in its estimate higher than in the previous case, since the spread of the gray samples around its estimate at $y = 2$ is tighter. Therefore, the estimated covariance of the filter is a function of the measurement, and the performance improves drastically because the uncertainty level always matches the estimate, providing a more reliable outcome. There appear to be no influential benefits in applying the multiple model polynomial estimator: SACEMM-3-5-2 behaves similarly to its single-model counterpart and shares the same features. However, at the tails of the distribution, SACEMM-3-5-2 estimated conditional covariance better follows the distribution of the samples.

The accuracy level reached by each filter is compared in Fig. 2, where the results of a RMSE analysis is reported.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N_{\mathrm{samples}}} (x_i - \hat{x}_i^+)^2}{N_{\mathrm{samples}}}}. \tag{70}$$

The RMSE of each estimator is evaluated using the entire set of 10^5 points. The bars show that SACE-3-5-2 is the most accurate filter, while the linear estimators are the least. However, a more precise approximation of the measurement equation leads to a smaller RMSE and to a more precise estimate, as proven by DAHO-3 (third-order Taylor polynomial) being the most accurate among the other linear estimators. The IEKF shares the same accuracy level as DAHO-3, while the other nonlinear estimators have lower RMSE. Two PF implementations are shown with different numbers of particles: 10^3

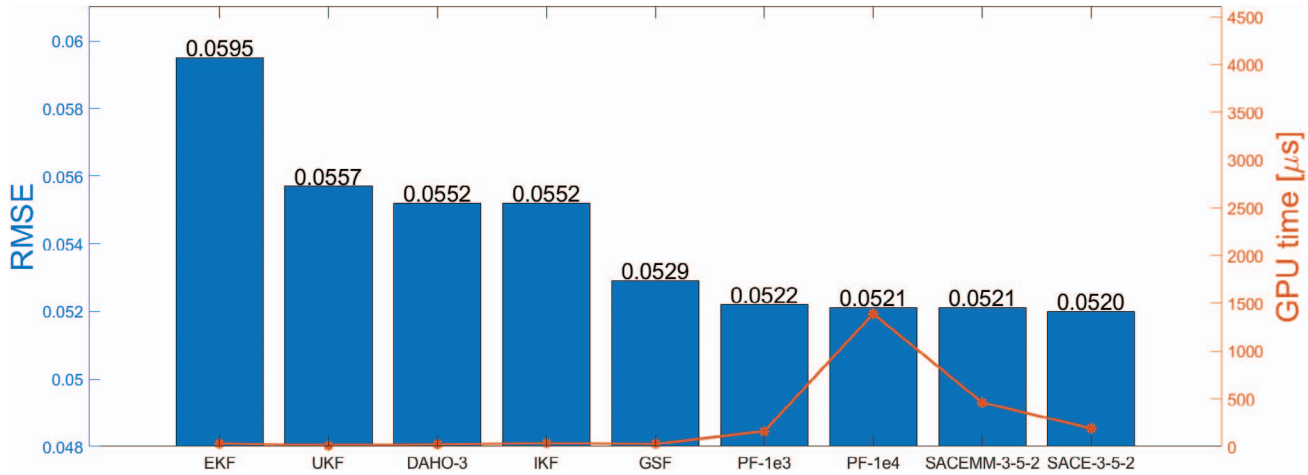

Fig. 2. Comparison of RMSE and computational time among different estimators.

and 10^4. It has error levels comparable with SACE-3-5-2, and the PF with 10^4 particles has a heavier computational burden. Fig. 2 reports, in orange, the average GPU time of each estimator, evaluated among all the 10^5 runs shown in Fig. 1. As expected, PF-1e4 has the highest computational time, while linear estimators have the lowest. SACE-3-5-2 achieves the best accuracy levels comparable to sample-based filters in a shorter amount of time, although the performance of all nonlinear filters is very similar in this simple motivating example.

The proposed scalar problem shows no significant difference between SACE-c-η-μ and SACEMM-c-η-μ. Let us increase the prior uncertainty level to $x \sim \mathcal{N}(1, 0.03)$ in order to underline the benefits of having multiple models. Fig. 3 shows the estimator function and confidence level of the two filters, along with the joint distribution. SACE-3-5-2 outputs unphysical results for the predicted conditional covariance of the state: a negative value of σ^2. The initial prior uncertainties are excessively large to allow the filter to work properly. On the left tail of the join distribution, the variance becomes negative and that is represented by the green lines overlapping the red one, to show that the filtering algorithm is not functioning correctly. SACEMM-3-5-2, on the other hand, has no issues in estimating correctly both the state and the covariance for all possible outcomes of the measurement. The green lines bound the samples of the joint distribution narrowing and widening as needed. The correct result is connected to the reduced initial covariance associated with each model, which increases the filter robustness and performance.

The proposed problem underlines a couple of characteristics of the proposed algorithms. Unlike the linear and Gaussian cases, the conditional covariance and the estimation error covariance are different. Linear filters employ the estimation error covariance, which expresses the average spread of the estimation error over all possible measurement realizations. This is a good metric, but once a measurement is actually available to process, the covariance conditioned on the actual measurement outcome is a more informative quantity, because it provides the spread of the estimation error for the actual value of y. In fact, the conditional covariance is a (nonlinear) function of the measurement whose evaluation is usually not feasible. SACE-c-η-μ and SACEMM-c-η-μ use a polynomial estimator to approximate the function, achieving better results with respect to filters that do not.

B. Lorenz96 System

The performance of the proposed filter is tested on a Lorenz96 example [30], where the state dynamics are

$$\frac{dx_i(t)}{dt} = x_{i-1}(t)(x_{i+1}(t) - x_{i-2}(t)) - x_i(t) + F + v_i(t), \tag{71}$$

with $i = 1, \ldots, 4$, since $\mathbf{x}(t)$ is selected to be 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 with value chosen equal to 8, since it introduces 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 $\sigma_v = 10^{-3}$. The dynamics are propagated at 2 Hz for a total of 20 s. The measurements are obtained at each time-step according to the following model:

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}(t_k) + \boldsymbol{\mu}_k, \quad \mathbf{H}_{i,j} = \begin{cases} 1 & j = 2i - 1 \\ 0 & \text{otherwise} \end{cases} \tag{72}$$

with $i = \{1, 2\}$ and $j = \{1, 2, 3, 4\}$. In other words, the sensors observe the components of the state with odd indices. Measurement noises are assumed to be Gaussian and uncorrelated within each other and with the process noise. The standard deviation is selected as $\sigma_{\boldsymbol{\mu}} = 0.5$: This value is particularly high, and filters based on linear estimators are not able to track the state of the system and achieve convergence [38].

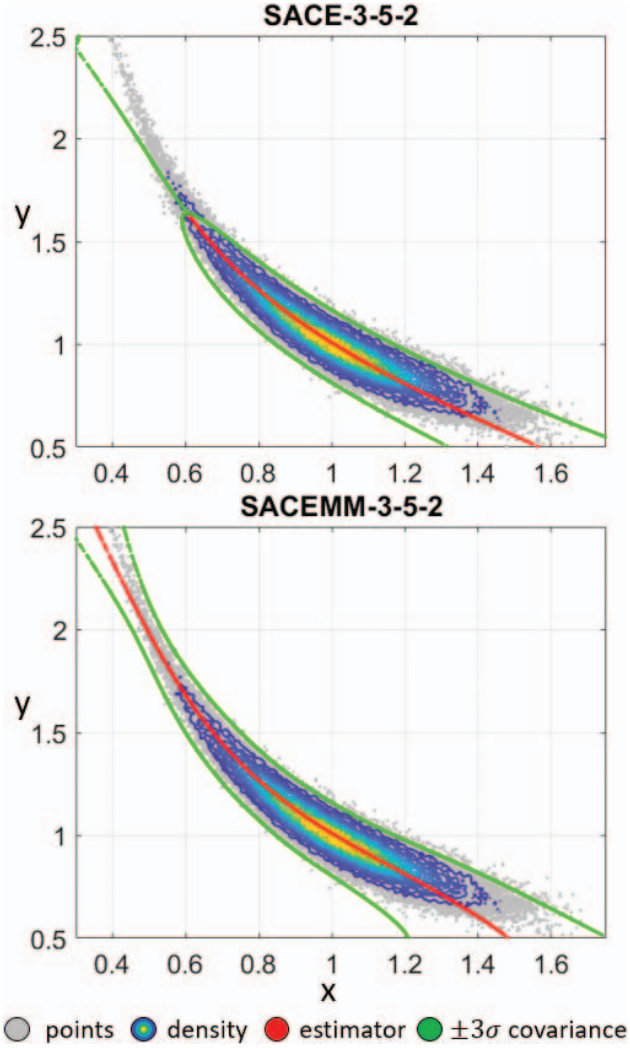

Fig. 3. SACE-3-5-2 versus SACEMM-3-5-2. Posterior distribution (gray), the estimator functions (red), and their confidence levels (green).

Fig. 4 shows the Monte Carlo analysis results performed with SACE-2-3-2 on the presented application. The figure shows, for each ith component of the state, the estimation error of each realization (gray lines), calculated as

$$\epsilon_{j,i} = x_{j,i} - \hat{x}_{j,i}, \tag{73}$$

for each jth time-step. A total of 100 realizations are reported. Fig. 4 describes the error means, in black, and the error standard deviations, as 3σ values, in blue. The black lines show that SACE-c-η-μ is an unbiased filter, as expected from the theory of MMSE estimation. The predicted error standard deviation, continuous blue line, is evaluated directly from the updated covariance matrix, by taking the square root of the diagonal terms. The effective performance of the filter is assessed by the sample standard deviation of the Monte Carlo estimation errors, dashed blue lines. At each time-step, the actual error covariance of the filter is evaluated by working directly on the samples. The consistency of SACE-2-3-2 is established by the overlapping of the dashed and continuous blue lines, which proves that the filter can correctly predict its own uncertainty levels.

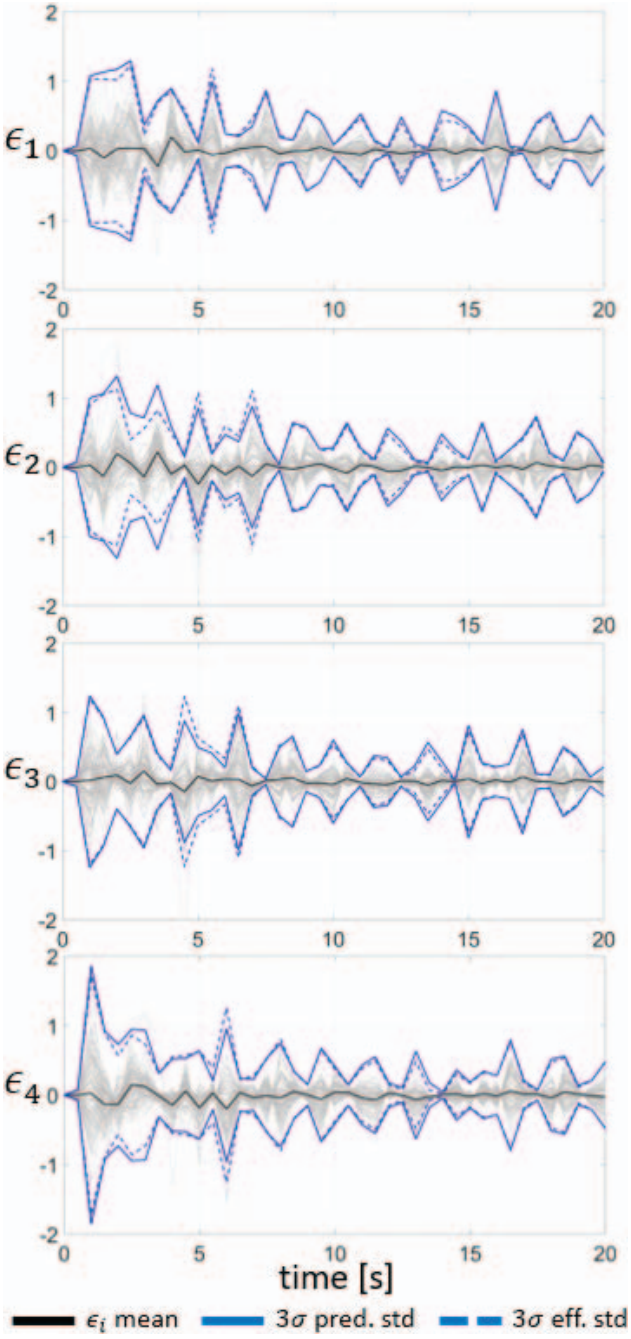

Fig. 4. Monte Carlo performance analysis with SACE-2-3-2: 100 runs.

The performance comparison among different filters is shown in Fig. 5 through another Monte Carlo analysis conducted with 100 runs. The figure shows, for each filter, the comparison between the effective and predicted error covariance. The continuous lines represent the filter's own estimate of the error standard deviation, calculated directly from the updated covariance matrix as the square root of its trace:

$$\bar{\sigma} = \sqrt{\mathrm{tr}(\hat{\mathbf{P}}_{\mathbf{xx}})}. \tag{74}$$

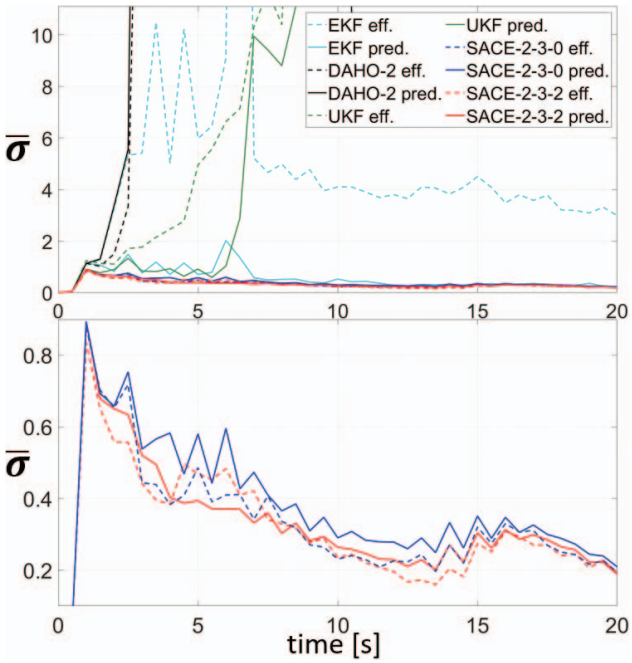

Fig. 5. Lorenz96: covariance comparison among different filters.

The dashed lines represent the effective error standard deviation derived from the Monte Carlo analysis. A consistent filter has the matching between its dashed and continuous lines, meaning that the estimated uncertainty level reflects the actual error standard deviation. The top graph, in Fig. 5 shows how linear estimators, the EKF, UKF, and DAHO-2, diverge (and break down) while trying to track the state of the system. The measurement noise level is excessively large, and a linear dependence on the measurement outcome is not sufficient to achieve a correct estimate. The EKF lines also represent the behavior of the IEKF: Since the measurement is a linear measurement, the IEKF reduces to the EKF. The UKF and DAHO-2 use, respectively, the unscented transformation and second-order Taylor polynomial to improve the prediction step of the filter and have a more accurate propagated state prior distribution. However, the update step is still linear and highly influenced by the noise standard deviation that prevents the evaluation of a reliable Kalman gain. The polynomial estimator better weights the information from the measurements using high-order moments and it achieves convergence and consistency. Therefore, SACE-2-3-0, in blue, and SACE-2-3-2, in red, correctly estimate the state of the system along the whole simulation. The bottom graph in Fig. 5 zooms in on the performance of SACE-c-η-μ for the two different sets of parameters. SACE-2-3-0 shows a filter whose estimate is a polynomial function of the measurement, and its estimated covariance is evaluated as a mean among all possible resolutions; it is not influenced by the measurement outcome. SACE-2-3-2, on the other hand, improves accuracy by estimating the covariance, giving it the same importance reserved for the state. Thus, the red lines settle below the blue ones for the whole simulation, since the predicted error standard deviation better matches the conditional mean.

C. Lorenz63 System

The performance of the proposed algorithms is also tested on a Lorenz63 application [30], [41], a challenging nonlinear system without process noise. The absence of process noise causes impoverishments in PFs, typically resulting in unsatisfactory performance. The state of the system undergoes the following dynamics:

$$\frac{dx_1(t)}{dt} = \alpha(x_2(t) - x_1(t)), \tag{75}$$

$$\frac{dx_2(t)}{dt} = x_1(t)(\gamma - x_3(t)) - x_2(t), \tag{76}$$

$$\frac{dx_3(t)}{dt} = x_1(t)x_2(t) - \beta x_3(t), \tag{77}$$

where $\alpha = 10$, $\beta = 8/3$, and $\gamma = 28$. For this selection of parameters, the Lorenz system has chaotic solutions. Almost all initial points will tend to the invariant set, the Lorenz attractor. In the presented application, the initial condition is assumed to be Gaussian with mean $\hat{\mathbf{x}} = \begin{bmatrix}10 & 10 & 10\end{bmatrix}^T$ and diagonal covariance matrix, with the same standard deviation for each component of the state: $\sigma_{\mathbf{x}} = 2.5$. The state is integrated in time at 30 Hz, with observations taken at each time-step. The measurement model consists of the range of the state from origin

$$y_k = \sqrt{x_1(t_k)^2 + x_2(t_k)^2 + x_3(t_k)^2} + \mu_k, \tag{78}$$

where measurement noise is assumed to be Gaussian with zero mean and standard deviation $\sigma_\mu = 1$.

Fig. 6 shows, on the top, one of the trajectories described by the state of the system, in its three components. The Lorenz attractor has two main lobes symmetric with respect to the x_3 axis: The resulting pathway has been labeled a "butterfly" shape. A Monte Carlo analysis with 1000 realizations with SACEMM-2-5-2 is reported at the bottom of Fig. 6. For each ith component of the state, the estimation error of each realization is calculated according to Equation (73), and reported in gray. Analogously with the previous application, the continuous blue lines represent the predicted error standard deviations, as 3σ values, of each component, while the dashed blue lines are the effective error standard deviations, again as 3σ values, calculated directly from the Monte Carlo realizations at each time-step. The overlapping between the dashed and the continuous lines indicates that SACEMM-2-3-2 is a consistent filter able to correctly estimate its own uncertainties. The black lines are the error means, and they prove the unbiased nature of the proposed filtering technique, as expected from the MMSE theory.

The performance of the filters have been assessed through a covariance comparison carried out with multiple Monte Carlo analyses, each performed with 1000

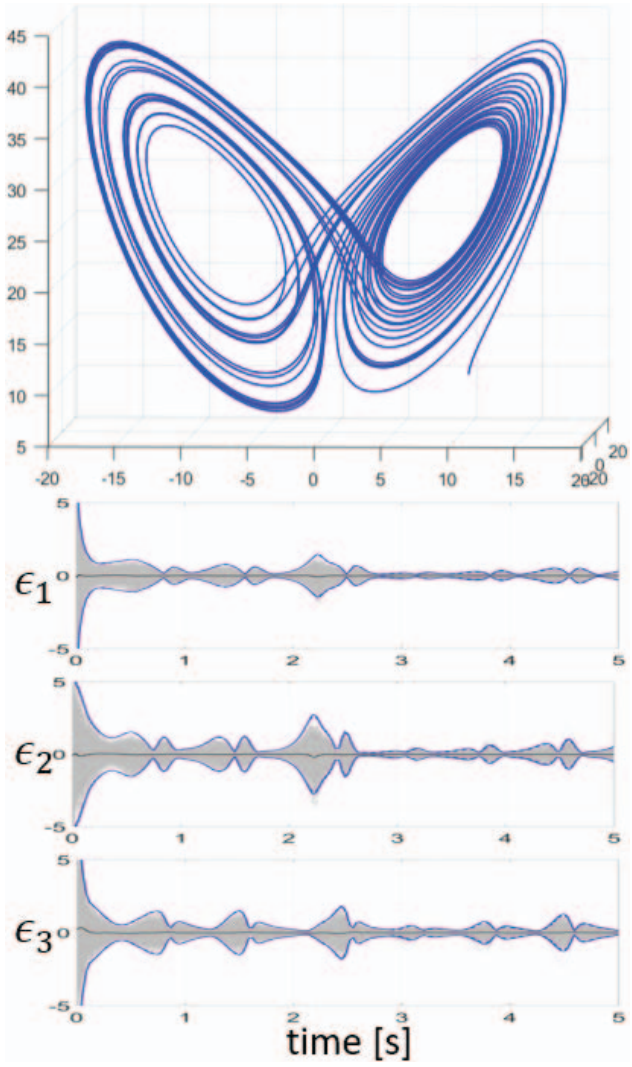

Fig. 6. Trajectory and SACEMM-2-3-2 Monte Carlo analysis results: 1000 runs.

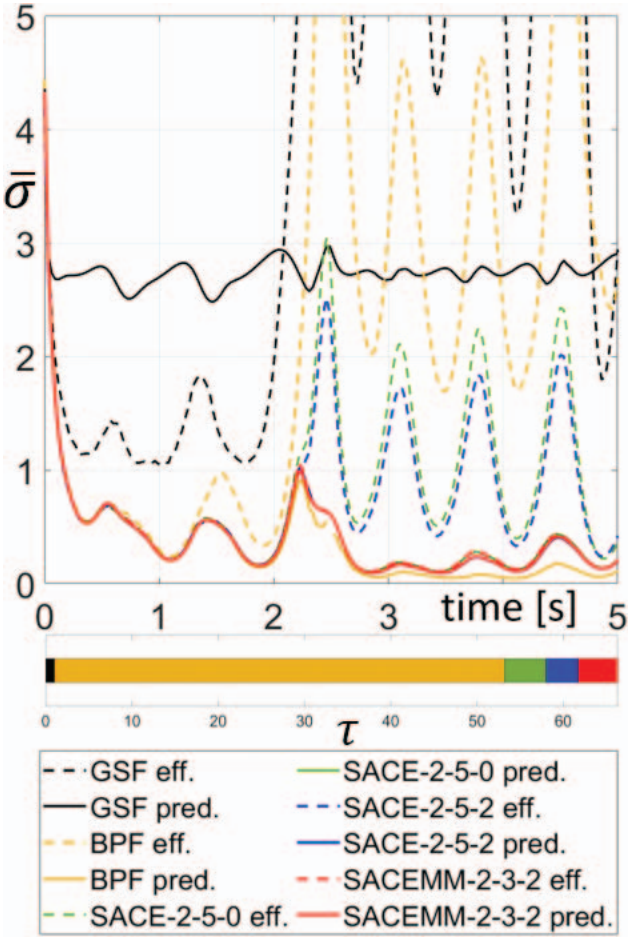

Fig. 7. Lorenz63: covariance comparison and time analysis among different filters.

runs. Fig. 7 reports, for each filter, the effective and the predicted error standard deviations. As shown in previous analysis, the dashed lines represent the actual uncertainty level of the filter, while the continuous lines are the filter's own uncertainty estimate, evaluated according to Equation (74). Fig. 7 reports SACE-c-η-μ and SACEMM-c-η-μ with different sets of parameters. For the basic selection of SACEMM-1-1-0, the filter reduces to the GSF, where the dynamics are linearized around the current center of each model, and the update is a linear estimator. The GSF is reported with black lines, and it fails to estimate the state of the system. The effective covariance indicates divergence and goes out of scale with respect to the predicted standard deviation.

The state of the system is also estimated with a 10^4 particles BPF, shown in orange, and the IEKF. The IEKF diverges rapidly and is not reported in the figure since the errors quickly reach out-of-scale large values. The linearization of the dynamics employed by the IEKF is not sufficient to correctly propagate the state covariance forward in time. The divergence of the IEKF might be connected to the poor time propagation. However, this issue might be alleviated by using the Levenberg–Marquardt algorithm [39]. The BPF performs better than the GSF but shows convergency problems, and it is not able to achieve an accurate estimate of the state. The BPF has issues due to the lack of process noise in the dynamics. After resampling, the propagated particles are not spread enough to be an appropriate representation of the prior uncertainty in order to accurately perform the measurement update.

SACE-c-η-μ is analyzed with the traditional zeroth-order covariance estimation, SACE-2-5-0 shown in green, and with a second-order covariance polynomial estimator, SACE-2-5-2 in blue. The two filters behave similarly: They both show convergence with consistency for the first half of the simulation, and they diverge for the remaining half. At time-step $t = 2.2$ s, the state of the system is near the origin, in between the two lobes of the Lorenz attractor. This point is critical because, due to uncertainty, the estimated state may select the incorrect lobe, while the true state follows the other. The measurement model, consisting solely of the range, gives no beneficial information regarding the lobe selection: Thus, the correction terms in the update step

do not help tracking the state along the correct path. Consequently, in some realizations of the Monte Carlo analysis, the filter is tracking the state of the system as if it were on the incorrect lobe. The radial nature of the range measurement provides no information to the estimator about correcting the estimated state because of the symmetric nature of the "butterfly" trajectory. Therefore, both SACE-2-5-0 and SACE-2-5-2 show inconsistency after the critical point, and the effective standard deviation is bigger than the predicted one. However, it is worth noticing that the dashed blue line settles below the dashed green line, indicating an increase in accuracy achieved due to the estimated covariance being connected with the measurement outcome. Lastly, SACEMM-2-3-2 is reported in red, and it is the only filter that shows convergency and consistency during the whole length of the simulation. The introduction of multiple models improves accuracy, especially around the critical point, where smaller subdomains make it easier for the filter to follow the right path along the correct lobe. If a model separates from the others, following the incorrect lobe, then it is weighted down in order to ensure a correct estimation. The division of the system uncertainties in smaller subdomains helps the filter track the correct trajectory, while the high-order polynomial update ensures excellent accuracy levels. SACEMM-c-η-μ has better performance than SACE-c-η-μ when the initial uncertainties of the state of the system are exceptionally high and when the propagated state PDF is multimodal.

The second part of Fig. 7 reports an analysis on the computational time requested by each filter. The parameter τ is evaluated as

$$\tau = \frac{\mathcal{T}_i}{\mathcal{T}_{\text{GSF}}}, \tag{79}$$

where $\mathcal{T}_i$ is the computational time of the ith filter, with $i = \{\text{GSF, BPF, SACE-2-5-0, SACE-2-5-2, SACEMM-2-3-2}\}$. Therefore, the τ bar expresses the relative computational effort among the different filters for this application. The τ analysis shows that the BPF is the computationally heaviest filter, while the computational time requested by SACE-c-η-μ and SACEMM-c-η-μ changes depending on the selection of their parameters.

VI. CONCLUSIONS

A novel filter based on a double estimator has been presented. The new technique estimates the conditional mean and the conditional covariance of the posterior distribution by applying, sequentially, two polynomial estimators, using the same measurement outcome. The new approach better matches the estimated state with its error standard deviation, which is now a polynomial function of the measurement. Therefore, the newly proposed filter is able to reduce the error uncertainty when the posterior distribution gets narrower around a low probability realization of the measurement. In turn, the better representation of the uncertainty produces a better estimate of the state during the subsequent measurement updates.

Three numerical examples have been reported. The scalar application gives a visual representation of the benefits of the polynomial approximation of the true MMSE and its covariance. Thus, the higher the order of the updates, the more precise the relative state estimate and its covariance. The vectorial application underlines the benefits of predicting the covariance by considering its estimation as working with an augmented state. The new state estimate improves in accuracy and a smaller error standard deviation is obtained. The multiple-model filter is more robust against high initial standard deviations and multimodal distributions.

REFERENCES

[1] D. Alspach and H. Sorenson
"Nonlinear Bayesian estimation using Gaussian sum approximations,"
IEEE Trans. Autom. Control, vol. 17, no. 4, pp. 439–448, Aug. 1972.

[2] I. Arasaratnam and S. Haykin
"Cubature Kalman filters,"
IEEE Trans. Autom. Control, vol. 54, no. 6, pp. 1254–1269, Jun. 2009.

[3] I. Arasaratnam, S. Haykin, and R. J. Elliot
"Discrete-time nonlinear filtering algorithms using Hauss–Hermite quadrature,"
Proc. IEEE, vol. 95, no. 5, pp. 953–977, May 2007.

[4] R. Armellin and P. Di Lizia
"Probabilistic optical and radar initial orbit determination,"
J. Guid., Control, Dyn., vol. 41, no. 1, pp. 101–108, 2018.

[5] R. Armellin, P. Di Lizia, F. Bernelli-Zazzera, and M. Berz
"Asteroid close encounters characterization using differential algebra: The case of Apophis,"
Celestial Mech. Dynamical Astron., vol. 107, no. 4, pp. 451–470, 2010.

[6] B. Bell and F. W. Cathey
"The iterated Kalman filter update as a Gauss–Newton method,"
IEEE Trans. Autom. Control, vol. 38, no. 2, pp. 294–297, Feb. 1993.

[7] M. Berz
Modern Map Methods in Particle Beam Physics. New York, NY, USA: Academic Press, 1999.

[8] F. Carravetta, A. Germani, and N. Raimondi
"Polynomial filtering of discrete-time stochastic linear systems with multiplicative state noise,"
IEEE Trans. Autom. Control, vol. 42, no. 8, pp. 1106–1126, Aug. 1997.

[9] F. Cavenago, P. Di Lizia, M. Massari, and A. Wittig
"On-board spacecraft relative pose estimation with high-order extended Kalman filter,"
Acta Astronautica, vol. 158, pp. 55–67, 2019.

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

[11] A. De Santis, A. Germani, and M. Raimondi
"Optimal quadratic filtering of linear discrete-time non-Gaussian systems,"
IEEE Trans. Autom. Control, vol. 40, no. 7, pp. 1274–1278, Jul. 1995.

[12] P. Di Lizia, M. Massari, F. Cavenago, and A. Wittig
"Assessment of onboard DA state estimation for spacecraft relative navigation," ESA, Paris, France, Contract IPL-PTE/LF/as/517.2016, Final Rep., 2017.

[13] A. Gelb
Applied Optimal Estimation. Cambridge, MA, USA: MIT Press, 1974.

[14] J. R. Fisher
"Optimal nonlinear filtering,"
Advances in Control Systems, vol. 5, pp. 197–300, 1967.

[15] N. J. Gordon, D. J. Salmond, and A. FM. Smith
"Novel approach to nonlinear/non-Gaussian Bayesian state estimation,"
IEE Proc. F-Radar Signal Process., vol. 140, no. 2, pp. 107–113, 1993.

[16] F. Hutter and R. Dearden
"The Gaussian particle filter for diagnosis of non-linear systems,"
IFAC Proc. Volumes, vol. 36, no. 5, pp. 909–914, 2003.

[17] L. Isserlis
"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.

[18] A. H. Jazwinski
Stochastic Processes and Filtering Theory (Mathematics in Sciences and Engineering 64). New York, NY, USA: Academic Press, 1970.

[19] S. J. Julier and J. K. Uhlmann
"Unscented filtering and nonlinear estimation,"
Proc. IEEE, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[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 Trans. Autom. Control, vol. 45, no. 3, pp. 477–482, Mar. 2000.

[21] R. E. Kalman
"A new approach to linear filtering and prediction problems,"
J. Basic Eng., vol. 82, no. 1, pp. 35–45, 1960.

[22] R. E. Kalman and R. S. Bucy
"New results in linear filtering and prediction theory,"
J. Basic Eng., vol. 83, no. 1, pp. 95–108, 1961.

[23] J. Lan and X. R. Li
"Nonlinear estimation based on conversion-sample optimization,"
Automatica, vol. 121, pp. 1873–2836, 2020.

[24] J. Lan and X. R. Li
"Nonlinear estimation by LMMSE-based estimation with optimized uncorrelated augmentation,"
IEEE Trans. Signal Process., vol. 63, no. 16, pp. 4270–4283, Aug. 2015.

[25] 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'"
IEEE Trans. Autom. Control, vol. 47. no. 8, pp. 1406–1408, 2002.

[26] Y. Liu, X. R. Li, and H. Chen
"Generalized linear minimum mean-square error estimation with application to space-object tracking,"
in *Proc. Asilomar Conf. Signals, Syst. and Comput.*, 2013, pp. 2133–2137.

[27] K. Makino and M. Berz
"Cosy infinity version 9, nuclear instruments and methods in physics research section A: Accelerators,"
Spectrometers, Detectors Associated Equip., vol. 558, no. 1, pp. 346–350, 2006.

[28] M. Massari, P. Di Lizia, F. Cavenago, and A. Wittig
"Differential Algebra software library with automatic code generation for space embedded applications," in *Proc. AIAA Inf. Syst.-AIAA Infotech@ Aerospace*, 2018, p. 0398.

[29] M. Murata, H. Nagano, and K. Kashino
"Monte Carlo filter particle filter,"
in *Proc. Eur. Control Conf.*, 2015, pp. 2836–2841.

[30] D. Raihan and S. Chakravorty
"Particle Gaussian mixture filters-I,"
Automatica, vol. 98, pp. 331–340, 2018.

[31] O. Ravn, M. Norgaard, and N. K. Poulsen
"New developments in state estimations for nonlinear systems,"
Automatica, vol. 36, no. 11, pp. 1627–1638, 2000.

[32] T. S. Schei
"A finite-difference method for linearization in nonlinear estimation algorithms,"
Automatica, vol. 33, no. 11, pp. 2053–2058, 1997.

[33] T. Schon, F. Gustafsson, and P.-J. Nordlund
"Marginalized particle filters for mixed linear/nonlinear state-space models,"
IEEE Trans. Signal Process., vol. 53, no. 7, pp. 2279–2289, Jul. 2005.

[34] S. Servadio
"High order filters for relative pose estimation of an uncooperative target," M.S. thesis, Politecnico di Milano, Milan, Italy, 2017.

[35] S. Servadio and R. Zanetti
"Differential algebra-based multiple Gaussian particle filter for orbit determination," *J. Optim. Theory Appl.* (2021). https://doi.org/10.1007/s10957-021-01934-8.

[36] S. Servadio and R. Zanetti
"Recursive polynomial minimum mean-square error estimation with applications to orbit determination,"
J. Guid., Control, Dyn., AIAA, vol. 45, no. 5, pp. 939–954, 2020.

[37] S. Servadio, R. Zanetti, and R. Armellin
"Maximum a posteriori estimation of hamiltonian systems with high order series expansions,"
in *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference held August 11–15, 2019, Portland, Maine, U.S.A.*, Volume 171 of Advanced in the Astronautical Sciences, Univelt, San Diego. pp. 2843–2858.

[38] S. Servadio, R. Zanetti, and B. A. Jones
"Nonlinear filtering with a polynomial series of G aussian random variables,"
IEEE Trans. Aerosp. Electron. Syst., vol. 57, no. 1, pp. 647–658, Feb. 2021.

[39] M. A. Skoglund, G. Hendeby, and D. Axehill
"Extended Kalman filter modifications based on an optimization view point," in *Proc. 18th Int. Conf. Inf. Fusion*, 2015, pp. 1856–1861, 2015.

[40] H. W. Sorenson and D. L. Alspach
"Recursive Bayesian estimation using Gaussian sums,"
Automatica, vol. 7, no. 4, pp. 465–479, 1971.

[41] G. Terejanu, P. Single, T. Singh, and P. D. Scott
"Adaptive Gaussian sum filter for nonlinear Bayesian estimation,"
IEEE Trans. Autom. Control, vol. 56, no. 9, pp. 2151–2156, Sep. 2011.

[42] M. Valli, R. Armellin, P. Di Lizia, and M. Lavagna
"A Gaussian particle filter based on differential algebra for spacecraft navigation,"
in *Proc. 63rd Int. Astronautical Congr.*, International Astronautical Federation, 2012, pp. 5178–5187.

[43] M. Valli, R. Armellin, P. Di Lizia, and M. Lavagna
"Nonlinear mapping of uncertainties in celestial mechanics,"
J. Guid., Control, Dyn., vol. 36, no. 1, pp. 48–63, 2012.

[44] R. van der Merwe, A. Doucet, N. De Freitas, and E. A. Wan
"The unscented particle filter,"
in *Proc. Advances Neural Inf. Process. Syst.*, 2001, pp. 584–590.

[45] A. Wittig, P. Di Lizia, R. Armellin, K. Makino, F. Bernelli-Zazzera, and M. Berz
"Propagation of large uncertainty sets in orbital dynamics by automatic domain splitting,"
Celestial Mechanics Dynamical Astron., vol. 122, no. 3, pp. 239–261, 2015.

[46] Y. Zhang and J. Lan
"Gaussian sum filtering using uncorrelated conversion for nonlinear estimation,"
in *Proc. 20th Int. Conf. Inf. Fusion*, 2017, pp. 1–8.

Simone Servadio is a Postdoctoral Associate with the Massachusetts Institute of Technology, Cambridge, MA, USA. He received the Doctorate degree from the University of Texas at Austin, Austin, TX, USA, studying nonlinear estimation, orbit determination, attitude estimation and control, tracking, and GNC applications. His current research interests include astrodynamics and the Koopman operator, with particular interest for stable and unstable manifolds, periodic orbits, and the circular restricted three-body problem.

Renato Zanetti is an Assistant Professor with the Aerospace Engineering, University of Texas at Austin, Austin, TX, USA. Prior to joining the University of Texas at Austin, he worked as an Engineer with the NASA Johnson Space Center and with Draper Laboratory. His research interests include nonlinear estimation, onboard navigation, and autonomous aerospace vehicles.