

UNCERTAINTY ESTIMATION THROUGH POLYNOMIAL MAP INVERSION

Simone Servadio*, Renato Zanetti†

This paper investigates the estimation of the uncertainties of a system using polynomial map inversions. The measurement noise influence on the observations from the sensors can be mapped back into the initial state probability density function. Therefore, a new technique that analyzes which portion of the ‘a priori’ distribution could have generated the measurements is proposed. The algorithm is tested in different applications, under either a square map and a case with measurement deficit.

INTRODUCTION

Estimation of the state of a stochastic dynamic system from noisy measurements is a research area that has attracted considerable interest. The optimal state estimate, in a Minimum Mean Square Error sense, is the mean of the state conditioned on all measurements. For the linear and Gaussian case, the conditional distribution remains Gaussian at all times and it can be parameterized by its mean and covariance matrix. The well-known Kalman Filter [1, 2] provides the mechanization to calculate the conditional mean and covariance matrix. Systems of practical interest, however, are typically characterized by nonlinear dynamics observed via nonlinear measurements. A closed form solution to calculate the conditional mean of systems undergoing nonlinear dynamics and nonlinear measurements is typically unavailable.

Different algorithms have been proposed to approximate the conditional mean of nonlinear systems. One common approach is to simply perform a linearization of the measurement and dynamics functions centered at the current estimate, and apply the Kalman filter equations as if the system were linear; the so-called Extended Kalman Filter (EKF) [3]. However, when the system is highly nonlinear, this simple linearization fails to convergence [4].

An approach typically more robust than linearization is using the unscented transformation to better approximate the nonlinear transformation of mean and covariance matrices, the so-called Unscented Kalman Filter (UKF) [5, 6]. Both the EKF and UKF approximate the conditional mean as a linear function of the current measurement outcome. In the presence of nonlinearities, a nonlinear function of the measurement typically produces more accurate estimates. Estimation algorithms using a nonlinear function of the current measurement are computationally more expensive than the EKF and UKF as they usually require a representation of the entire conditional distribution, rather than just its mean and covariance matrix. These algorithms include the Gaussian Sum Filter [7, 8] (which approximates the conditional distribution as a Gaussian Mixture Model) and the Particle Filter [9] (which approximates it with a finite number of samples).

*Postdoctoral Associate, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology

†Assistant Professor, Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin

The algorithms mentioned above offer different approximations of the nonlinear transformation of random vectors through the measurement function. These approximations are then used to compute the estimator function. Armellin *et al.* [10] take a different approach. They collect a small batch of measurements and invert the measurement function to obtain a “measured” state. They then apply a linear estimator to this linear “measurement”. The transformation of mean and covariance matrix through the nonlinear functional inverse is still approximated, but the algorithm provides very good accuracy as it is a nonlinear function of the original measurement but only requires approximating the nonlinear transformation of mean and covariance rather than carrying the entire conditional distribution.

This work expands the results of [10] by processing measurements at every time without the need of collecting small batches of measurements at different times to make the measurement function square and invertible.

POLYNOMIAL APPROXIMATION

Different nonlinear filters are characterized from how they approximate the nonlinear transformation of random vectors. In this work nonlinear functions are approximated with their high order Taylor series expansions and then polynomial transformations of random vectors are computed.

The Taylor series and all polynomial operations are performed with Differential Algebra (DA) techniques [11]. DA supplies the tools to compute the derivatives of functions within a computer environment [12]. In DA, each variable and function is expressed as the Taylor series expansion up to a selected arbitrary order c and given a valid center for the expansions. The software implementation used to produce the numerical results of this paper is the Differential Algebra Core Engine, DACE2.0 [13]. By representing each function as an array of coefficients and exponents, the evaluation of derivatives, integration, gradients, are rapidly computed in the DA framework. Therefore, once a center point of the Taylor series expansion is selected, any function can be represented into its polynomial approximation (up to the selected order c), where the polynomial variable is the deviation from the expansion series itself.

As an illustrative example, consider variable x and the $\sin(\cdot)$ function. After selecting center $\hat{x} = 1$, the DA framework approximates $\sin(x)$ around \hat{x} as a function of the deviation from the center of the expansion; $\delta x = x - \hat{x}$ (coefficients precision has been reduced for length purposes).

$$\begin{aligned} \sin(x)|_{x=1} &= 0.84147 + 0.54030\delta x - 0.42735\delta x^2 + \dots \\ &= \sin(\hat{x}) + \mathcal{M}_{\hat{x}}(\delta x) \end{aligned} \tag{1}$$

where the series can be truncated at any order. Equation (1) shows that the first term of the series is the function evaluated at the given center. All the remaining terms depend on the initial deviation from the center, δx , and they represent the polynomial map, $\mathcal{M}_{\hat{x}}(\delta x)$, of the function connecting deviation in x to their value $\sin(x)$.

It is therefore possible to operate with the Taylor approximation of functions likewise with the regular Floating Points (FP) algebra. Moreover, the DACE2.0 [14, 15] has embedded polynomial operations such as evaluation, center shifting, map inversion, and so on.

For a more complete and detailed explanation on Differential Algebra, please refer to the references [16, 17].

THE TRACE BACK NOISE FILTER - TBNC

The section describes the new filtering technique. The Trace Back Noise filter (TBNC) uses DA techniques to create the polynomial map of the flow of the dynamics, connecting state deviations at time step $k + 1$ to state deviations at time step k , and to create the polynomial map of the measurement equations, that connects deviation in the measurement space to deviations in the state space.

Consider the following equations of motion, describing a nonlinear dynamic system affected by process noise. The information about the system is a set of measurements, related to the state vector, acquired at discrete times

$$\mathbf{x}_{k+1} = \mathbf{f}_k[\mathbf{x}_k] + \boldsymbol{\nu}_k \quad (2)$$

$$\mathbf{y}_{k+1} = \mathbf{h}_{k+1}[\mathbf{x}_{k+1}] + \boldsymbol{\mu}_{k+1} \quad (3)$$

where \mathbf{x}_k is the n components long state vector at time step k , \mathbf{y}_{k+1} is the m components long measurement vector at time step $k + 1$, $\mathbf{f}_k[\cdot]$ is the process model, and $\mathbf{h}_{k+1}[\cdot]$ is the measurement function. The system is subject to process noise $\boldsymbol{\nu}_k$ and measurement noise $\boldsymbol{\mu}_{k+1}$ that are potentially non-Gaussian zero-mean random sequences that satisfy the conditions $\forall i, j > 0$:

$$\mathbb{E}\{\boldsymbol{\nu}_i\} = \mathbb{E}\{\boldsymbol{\mu}_i\} = 0 \quad (4)$$

$$\mathbb{E}\{\boldsymbol{\nu}_i \boldsymbol{\mu}_j^T\} = 0 \quad (5)$$

$$\mathbb{E}\{\boldsymbol{\nu}_i \boldsymbol{\nu}_j^T\} = \mathbf{Q}_i \delta_{i,j} \quad (6)$$

$$\mathbb{E}\{\boldsymbol{\mu}_i \boldsymbol{\mu}_j^T\} = \mathbf{R}_i \delta_{i,j} \quad (7)$$

The probability density function at time step k is approximated as Gaussian, with knowledge of the state estimate, $\hat{\mathbf{x}}_k$, and its uncertainty spread around the mean in terms of covariance matrix \mathbf{P}_k . The state polynomial \mathbf{x}_k is initialized as

$$\mathbf{x}_k(\delta \mathbf{x}_k) = \hat{\mathbf{x}}_k + \delta \mathbf{x}_k \quad (8)$$

where the DA variable $\delta \mathbf{x}_k$ indicates the deviation from the center of the series.

Time Propagation

The state polynomial is propagated in time in the DA framework according to the equation of motion

$$\mathbf{x}_{k+1}(\delta \mathbf{x}_k) = \mathbf{f}_k[\mathbf{x}_k(\delta \mathbf{x}_k)] \quad (9)$$

Differential Algebra replaces the classic integration scheme with corresponding operations in the DA framework. The results of the DA ODE integration is the propagated Taylor series expansion of the flow forward in time [18]. Therefore, Equation (9) represents the forward map of the dynamics that connects deviations at time step k to deviations at time step $k + 1$. Thanks to the Gaussian approximation of the PDF of the state at time step k , all the central moments of \mathbf{x}_k can be easily calculated using the Isserlis' formulation [19], starting from \mathbf{P}_k . Therefore, the central moments of \mathbf{x}_{k+1} are calculated by working directly on the monomials of the polynomial itself, since it is a function of the deviations $\delta \mathbf{x}_k$. The polynomial expressed by Equation (9) can therefore be written putting in evidence each single monomial: for the i th component of the state

$$\mathbf{x}_{k+1,i}(\delta \mathbf{x}_k) = \mathbf{f}_{k,i}[\hat{\mathbf{x}}_k] + \sum_{r=1}^c \frac{1}{r!} \sum \frac{\partial^r \mathbf{f}_{k,i}[\mathbf{x}_k]}{\partial \mathbf{x}_1^{\gamma_1} \dots \partial \mathbf{x}_n^{\gamma_r}} \delta \mathbf{x}_1^{\gamma_1}(k) \dots \delta \mathbf{x}_n^{\gamma_r}(k) \quad (10)$$

where the second summation is over all the possible permutations of $\gamma_i \in \{1, \dots, n\}$ with $i \in \{1, \dots, r\}$, and it indicates all the derivatives up to the selected order c . Hence, the summation of $\mathbf{f}_{k,i}[\mathbf{x}_k]$ expresses all the higher-order partial derivatives of the solution of the flow, which directly connect deviations from time step k to time step $k+1$. Consequently, each function needs to be c -times differentiable in the selected support.

The predicted mean is calculated by applying the expectation operator on the propagated polynomial;

$$\hat{\mathbf{x}}_{k+1}^- = \mathbb{E} \{ \mathbf{x}_{k+1}(\delta \mathbf{x}_k) \} = \mathbf{f}_{k,i}[\hat{\mathbf{x}}_k] + \sum_{r=1}^c \frac{1}{r!} \sum \frac{\partial^r \mathbf{f}_{k,i}[\mathbf{x}_k]}{\partial \mathbf{x}_1^{\gamma_1} \dots \partial \mathbf{x}_n^{\gamma_r}} \mathbb{E} \{ \delta \mathbf{x}_1^{\gamma_1}(k) \dots \delta \mathbf{x}_n^{\gamma_r}(k) \} \quad (11)$$

where, being a linear operator, the expectations work directly on each single monomial of the series. The predicted mean is therefore evaluated by substituting, to each deviation product, the relative Isserlis' moment evaluated from \mathbf{P}_k , since the polynomial map is a function of the original deviation initialized in Equation (8).

The predicted state covariance matrix is evaluated likewise the mean, by considering the deviation of the propagated polynomial with respect to the estimated mean.

$$\mathbf{P}_{k+1}^- = \mathbb{E} \left\{ (\mathbf{x}_{k+1}(\delta \mathbf{x}_k) - \hat{\mathbf{x}}_{k+1}^-) (\mathbf{x}_{k+1}(\delta \mathbf{x}_k) - \hat{\mathbf{x}}_{k+1}^-)^T \right\} + \mathbf{Q} \quad (12)$$

where the argument of the expectation are polynomials in $\delta \mathbf{x}_k$ whose expected value are, again, the central moments of the initial Gaussian PDF. The state distribution has been propagated in time in terms of its mean and covariance, having considered the influence of the process noise.

Measurement Update

The time prediction step is followed by the measurement update. The update starts by initializing a new polynomial, with a new DA variable (deviation) around the current predicted mean.

$$\mathbf{x}_{k+1}(\delta \mathbf{x}_{k+1}) = \hat{\mathbf{x}}_{k+1}^- + \delta \mathbf{x}_{k+1} \quad (13)$$

The measurement equation is applied to the state polynomial, following the same procedure used in the time propagation step:

$$\mathbf{y}_{k+1}(\delta \mathbf{x}_{k+1}) = \mathbf{h}_{k+1}[\mathbf{x}_{k+1}(\delta \mathbf{x}_{k+1})] = \bar{\mathbf{y}}_{k+1} + \mathcal{M}_{\hat{\mathbf{x}}_{k+1}^-}(\delta \mathbf{x}_{k+1}) \quad (14)$$

The c th order measurement polynomial $\mathbf{y}_{k+1}(\delta \mathbf{x}_{k+1})$ has been divided into its center (or constant part) $\bar{\mathbf{y}}_{k+1}$, and the polynomial map of deviations, $\mathcal{M}_{\hat{\mathbf{x}}_{k+1}^-}(\delta \mathbf{x}_{k+1})$, centered at $\hat{\mathbf{x}}_{k+1}^-$, that connects deviations from the state space to deviations in the measurement space. Let initially assume a square invertible map, where $m = n$, which implies at every time there are as many measurements as states. Under these conditions, the map can be inverted using standard DA techniques [20] to obtain

$$\mathcal{M}_{\bar{\mathbf{y}}_{k+1}}(\delta \mathbf{y}_{k+1}) = \left(\mathcal{M}_{\hat{\mathbf{x}}_{k+1}^-}(\delta \mathbf{x}_{k+1}) \right)^{-1} \quad (15)$$

That is, the inverted polynomial map connects deviations from the measurement space to the state space. Let $\tilde{\mathbf{y}}_{k+1}$ be the actual noisy observation coming from the sensors, whose value is influenced by measurement noise with known distribution. The intuition behind the TBN c filter is depicted in

Figure 1. The figure shows the Taylor polynomial maps. The forward map $\mathbf{h}[\mathbf{x}]$, in red, describes the approximation of the the measurement distribution from the state predicted PDF. The map can be inverted, in blue, to trace back deviations from the measurement space to deviations in the state space. Considering the observation $\tilde{\mathbf{y}}_{k+1}$ from the sensors and the known distribution of the measurement noise, it is possible to highlight the portion of the measurement PDF that generated the observation received. Assuming Gaussian measurement noise with covariance matrix \mathbf{R} , and centering the Gaussian at $\tilde{\mathbf{y}}_{k+1}$, we construct a distribution (the blue ellipse in Figure 1) of the true measurement. The true measurement is the perfect measurement value in the absence of measurement noise. Applying the inverse map to this measurement distribution we obtain the distribution of the state that could have generated the measurements, the orange region in the figure.

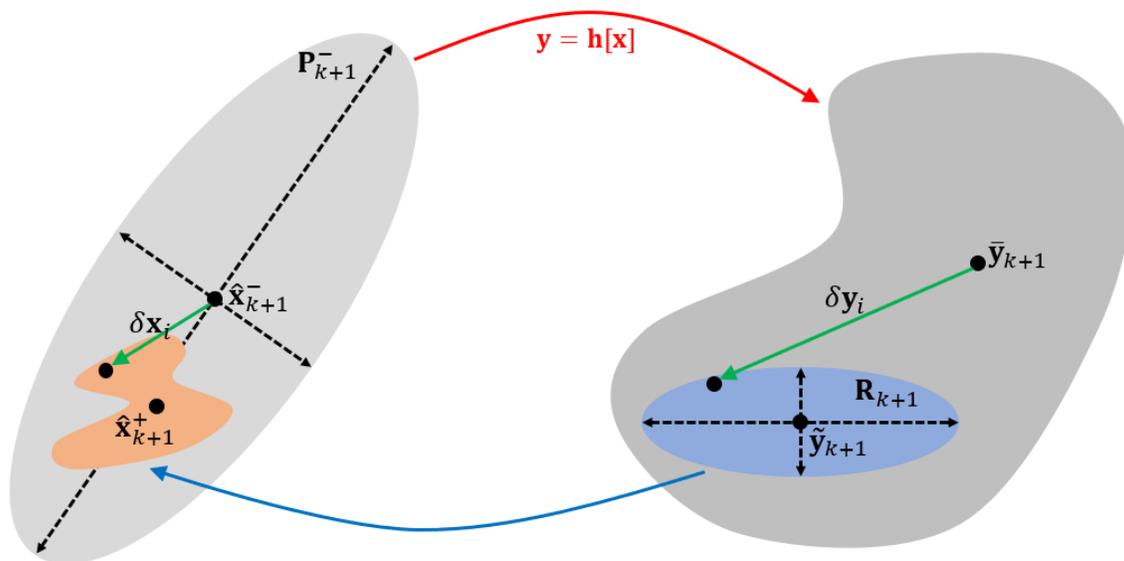


Figure 1. Sketch of the Taylor maps involved in the TBNc filter and the connection between deviations in the state space and in the measurement space.

The procedure described by Figure 1 is achieved numerically by sampling particles in the measurement space. Considering Gaussian measurement noise, N particles are drawn from $\mathcal{N}(\tilde{\mathbf{y}}_{k+1}, \mathbf{R}_{k+1})$. It is now possible to evaluate the deviation of each particle from the center of the polynomial map

$$d\mathbf{y}^{[i]} = \mathbf{y}^{[i]} - \bar{\mathbf{y}}_{k+1} \quad (16)$$

where $\mathbf{y}^{[i]}$ is the measurement particle, with $i = 1, \dots, N$. Each particle has the same weight, since they are drawn directly from the measurement noise PDF. Using the inverted map, it is possible to solve for the state deviation that generates each sample $\mathbf{y}^{[i]}$

$$d\mathbf{x}^{[i]} = \mathcal{M}_{\bar{\mathbf{y}}_{k+1}}(d\mathbf{y}^{[i]}) \quad (17)$$

The corresponding particles in the state space are simply calculated by adding the center of the state polynomial

$$\mathbf{x}^{[i]} = \hat{\mathbf{x}}_{k+1}^- + d\mathbf{x}^{[i]} \quad (18)$$

with $i = 1, \dots, N$. These particles describe the orange region in Figure 1 and they circumscribe the region where the true state lies. However, these particles are missing information from the

prior distribution, assumed as a Gaussian with mean $\hat{\mathbf{x}}_{k+1}^-$ and covariance \mathbf{P}_{k+1}^- . Therefore, at each particle is assigned a weight proportional to its likelihood given the prior

$$w_+^{[i]} = \mathcal{N}\left(\mathbf{x}^{[i]}; \hat{\mathbf{x}}_{k+1}^-, \mathbf{P}_{k+1}^-\right) \quad (19)$$

After normalizing the N weights $w_+^{[i]}$, the estimate and the updated state covariance are calculated as a weighted mean among all the particles:

$$\hat{\mathbf{x}}_{k+1}^+ = \sum_{i=1}^N w_+^{[i]} \mathbf{x}^{[i]} \quad (20)$$

$$\mathbf{P}_{k+1}^+ = \sum_{i=1}^N w_+^{[i]} (\mathbf{x}^{[i]} - \hat{\mathbf{x}}_{k+1}^+) (\mathbf{x}^{[i]} - \hat{\mathbf{x}}_{k+1}^+)^T \quad (21)$$

The filter is ready to perform the following iteration, constructing a new state polynomial from $\hat{\mathbf{x}}_{k+1}^+$, and central moments from \mathbf{P}_{k+1}^+ with the Isserlis' formulation.

Non-Square Mapping

The main drawback of the filter described in the previous section is that the measurement map needs to be square in order to be invertible, requirement that translates in the $m = n$ constraint. In [10], the authors overcome the issue by collecting measurements at different times until the required amount of measurements is reached. However, this solution requires the number of states to be a multiple of the number of measurement (as shown in the applications in [10]) and it leads to the creation of complex polynomial maps that undergo multiple propagations and functions among different time steps. TBNc offers a different solution that allows to perform the measurement update at each iteration. When the filter is working with fewer measurements than the needed number, the algorithm creates fictitious polynomials, \mathbf{y}_F , that make the map invertible. Therefore, the measurement vector is divided into two parts:

$$\mathbf{y}_{aug,k+1}(\delta\mathbf{x}_{k+1}) = \left[\mathbf{y}_{k+1}^T(\delta\mathbf{x}_{k+1}) \quad \mathbf{y}_F^T(\delta\mathbf{x}_{k+1}) \right]^T \quad (22)$$

The first m components are the polynomials evaluated according to the measurement equation, while the remaining $(n - m)$ polynomials are adequately constructed by the filter itself according to some chosen function $\mathbf{g}[\cdot]$:

$$\mathbf{y}_F(\delta\mathbf{x}_{k+1}) = \mathbf{g}[\mathbf{x}_{k+1}(\delta\mathbf{x}_{k+1})] \quad (23)$$

The fictitious measurement function $\mathbf{g}[\cdot]$ can be chosen freely. It is convenient to select measurements equal to single components of the state, in order not to increase the computational burden of the filtering technique. The measurement vector from the sensors $\tilde{\mathbf{y}}$ needs to be augmented as well in order to be dimensionally consistent with the number of polynomials:

$$\tilde{\mathbf{y}}_{aug,k+1} = \left[\tilde{\mathbf{y}}_{k+1}^T \quad \tilde{\mathbf{y}}_{F,k+1}^T \right]^T \quad (24)$$

where the values of $\tilde{\mathbf{y}}_{F,k+1}^T$ are chosen as

$$\tilde{\mathbf{y}}_{F,k+1} = \mathbb{E} \{ \mathbf{y}_F(\delta\mathbf{x}_{k+1}) \} \quad (25)$$

which are calculated as in Equation (11). In the case where the additional measurement polynomials are a linear combination of the components of the state, the fictitious measurements can be evaluated directly from the mean of the components of the state vector.

The measurement polynomial map is now invertible according to Equation (15). The TBN_c measurement update continues by sampling N particles in the augmented measurement space. The first m components of $\mathbf{y}_{aug}^{[i]}$ are drawn according to $\mathcal{N}(\tilde{\mathbf{y}}_{k+1}, \mathbf{R}_{k+1})$. Since the fictitious measurements add no information to the system, but they are purely a tool to perform the update, the remaining $(n - m)$ components of $\mathbf{y}_{aug}^{[i]}$ are drawn from a Gaussian with infinite covariance; $\mathcal{N}(\tilde{\mathbf{y}}_{F,k+1}, \infty)$. There is no *a priori* knowledge of the distribution of the fictitious measurements, therefore, each particle has the same weight regardless the magnitude of its deviation from the center of the series. In practice, the components of $\mathbf{y}_{aug}^{[i]}$ connected to the fictitious measurements are drawn from a uniform distribution, with boundaries chosen at five times the standard deviation of the selected variable. In the case of a nonlinear selection of $\mathbf{y}_F(\delta\mathbf{x}_{k+1})$, the relative covariance can be evaluated with Equation 12, substituting the measurement polynomial, and mean, to the state variables. The filter is now able to continue the remaining part of the algorithm as if it was working in normal conditions.

NUMERICAL APPLICATIONS - LORENZ63

The performance of the presented filtering technique has been tested on a Lorenz63 application [21, 22], where the state of the system undergoes the following dynamics

$$\frac{dx_1(t)}{dt} = \alpha(x_2(t) - x_1(t)) + \nu_1(t) \quad (26)$$

$$\frac{dx_2(t)}{dt} = x_1(t)(\rho - x_3(t)) - x_2(t) + \nu_2(t) \quad (27)$$

$$\frac{dx_3(t)}{dt} = x_1(t)x_2(t) - \beta x_3(t) + \nu_3(t) \quad (28)$$

with $\alpha = 10$, $\beta = 8/3$, and $\rho = 28$. This particular selection of parameter produces a chaotic behavior of the system, where almost all initial points will tend to the invariant set: the Lorenz attractor. In the presented application, the initial condition is assumed Gaussian, with parameters

$$\hat{\mathbf{x}}_0 = [4 \quad 5 \quad 3]^T \quad (29)$$

$$\mathbf{P}_0 = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.3 & 0 \\ 0 & 0 & 0.2 \end{bmatrix} \quad (30)$$

The state is integrated at 30 Hz, with the addition of progress noise $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}_3, 10^{-4}\mathbf{I}_{3 \times 3})$. Observations are obtained at each time step, and they consist in the range and bearing angles measurements, according to the following equations:

$$y_1(t_k) = \arctan\left(\frac{x_2(t_k)}{x_1(t_k)}\right) + \mu_1(t_k) \quad (31)$$

$$y_2(t_k) = \arcsin\left(\frac{x_3(t_k)}{\sqrt{x_1(t_k)^2 + x_2(t_k)^2 + x_3(t_k)^2}}\right) + \mu_2(t_k) \quad (32)$$

$$y_3(t_k) = \sqrt{x_1(t_k)^2 + x_2(t_k)^2 + x_3(t_k)^2} + \mu_3(t_k) \quad (33)$$

Observations are affected by Gaussian noise $\boldsymbol{\mu} \sim \mathcal{N}(\mathbf{0}_3, \mathbf{R})$, with

$$\mathbf{R} = 10^{-6} \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 400 \end{bmatrix} \quad (34)$$

A Monte Carlo analysis with 1000 runs has been performed with TBN2. Figure 2 reports the results of the performance analysis showing the estimation errors ϵ_i , for each of the three components of the state. The gray lines represent each single Monte Carlo run, while the black lines are the mean among all of them. Hence, the TBN2 filter is unbiased because the mean of the errors is null for all time steps. Figure 2 reports also the error standard deviation levels in blue. The continuous blue lines are the predicted standard deviation levels, that represents the filter's own uncertainty estimate, calculated directly from the updated covariance matrix (Equation (21)). The dashed blue lines are the effective standard deviation levels, that are calculated directly from all the Monte Carlo runs for each time step. The matching between continuous and dashed lines proves that the filter is consistent and that its own prediction of the spread of the error is correct.

The filtering technique can be now compared with the most common estimators. As such, a Monte Carlo analysis of the error state standard deviation has been performed by analyzing the performance of the Iterated Extended Kalman Filter (IKF), the UKF, the Bootstrap Particle Filter (BPF), and of the Differential Algebra High-Order filter (DAHO-*c*) from [23]. The predicted covariance of each filter is represented in sense of the standard deviation of the whole state

$$\sigma_{pred}^2 = \sum_{i=1}^n \mathbf{P}_{k+1,ii}^+ \quad (35)$$

where, indeed, σ_{pred}^2 is the trace of the updated covariance matrix. This value is compared to the effective distribution of the error of the state, whose covariance is evaluated, at each time step, among all the Monte Carlo runs

$$\bar{\epsilon}_i = \sum_{j=1}^{MC} \epsilon_{i,j} \quad (36)$$

$$\sigma_{eff}^2 = \sum_{i=1}^n \left(\frac{1}{MC} \sum_{j=1}^{MC} (\epsilon_{i,j} - \bar{\epsilon}_i)^2 \right) \quad (37)$$

with $i = 1, \dots, n$, and where MC is the number of Monte Carlo runs, selected as 1000 in the proposed analysis. Therefore, σ_{pred}^2 represents the filter's own prediction of its uncertainties, while σ_{eff}^2 shows the actual performance in terms of the spread of the error. A consistent filtering technique has a close match between σ_{pred}^2 and σ_{eff}^2 , meaning that the algorithm can correctly predict the covariance of the posterior distribution. Figure 3 reports, with a logarithmic scale, the results of the covariance analysis for the 5 different filters: each dashed line is σ_{eff}^2 , while the continuous lines represent σ_{pred}^2 . Notice how all the filters have comparable levels of the predicted uncertainties: the continuous lines settle to the same level of accuracy. The zoomed part of the figure highlights the difference, at steady state, among the continuous lines. The first filter to diverge is the IKF, blue lines in the figure. The dashed blue line separates from the continuous line right after the first iteration. The reason behind this behavior was found analyzing the single runs of the Monte Carlo

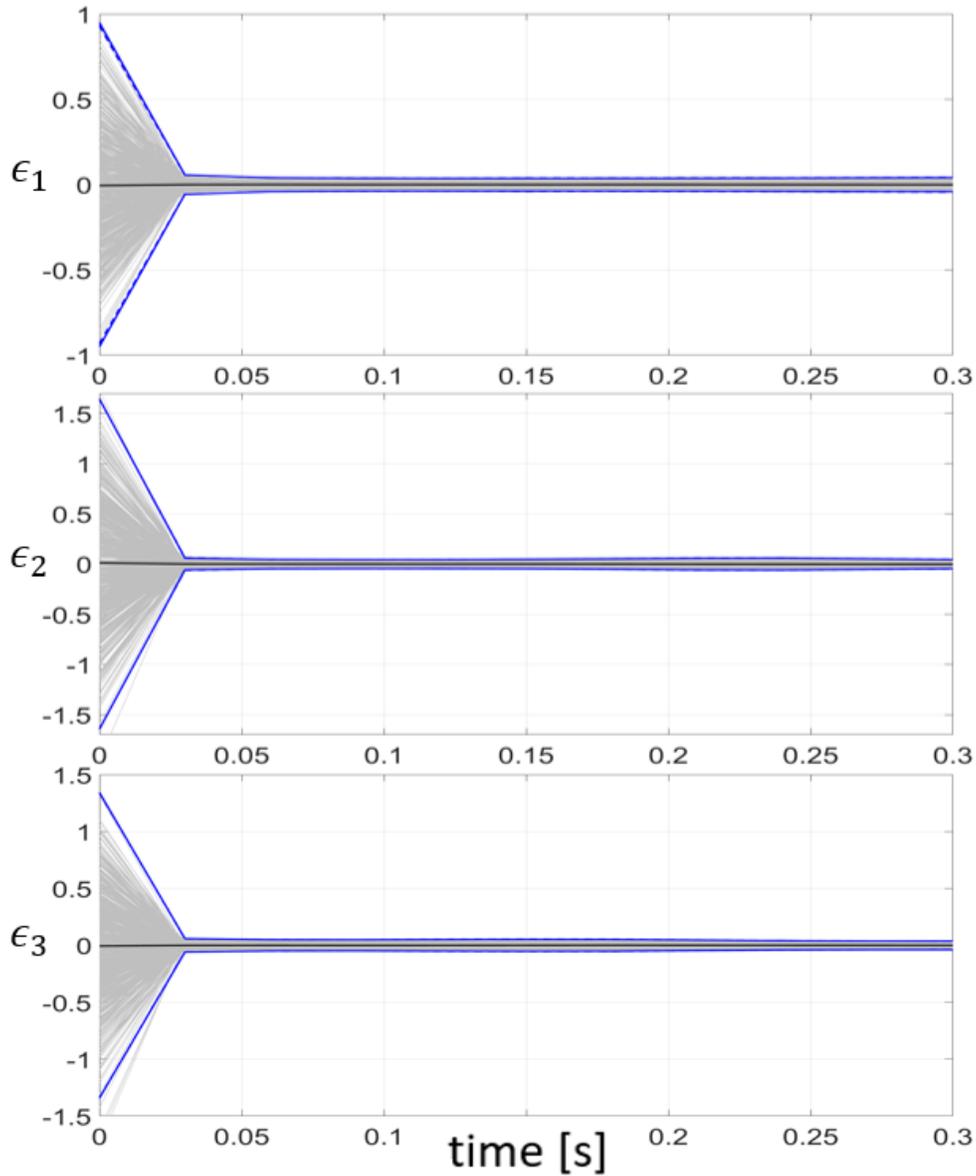


Figure 2. Results of the Monte Carlo analysis performed on the Lorenz63 system with the TBN2 filter for 1000 runs.

of the IKF. The initial covariance matrix is excessively large and the runs that start on the edge of the distribution are not able to reach converge and steady state behavior, diverging. The effective performance of the DAHO-2, in orange, and of the UKF, in green, is similar. Both filters reach steady state level and start diverging around the same time step. The particular high nonlinearities of the system, associated with the influence of the process noise, make the filters fail on their task to keep track of the state of the system, and they diverge. DAHO-2 and the UKF have been proven to have similar performances [17, 16], however, while DAHO-2 has a consistent prediction until the singularity, the predicted covariance of the UKF, continuous green line, decreases monotonically. The first consistent filter is the BPF, red lines in the figure. At steady state, the continuous and dashed red lines overlap, indicating a correct estimation of the covariance. However, the dashed red

line reaches steady state only after a third of the duration of the simulation. During the transient, the BPF has an incoherent prediction of the uncertainties and the filter believes it is performing more accurately than the actual performance. Lastly, in black, there is the TBN2 filter. TBN2 is the only filter that reaches convergence and has a consistent estimation of the error covariance along the whole simulation. Moreover, looking at the enhanced part of Figure 3, the continuous and dashed black lines perfectly overlap, while the only other consistent filter at steady state, the BPF in red, is less accurate. TBN2 shows to be the most accurate filtering technique, for the presented application, since its dashed line is the lowest among all others.

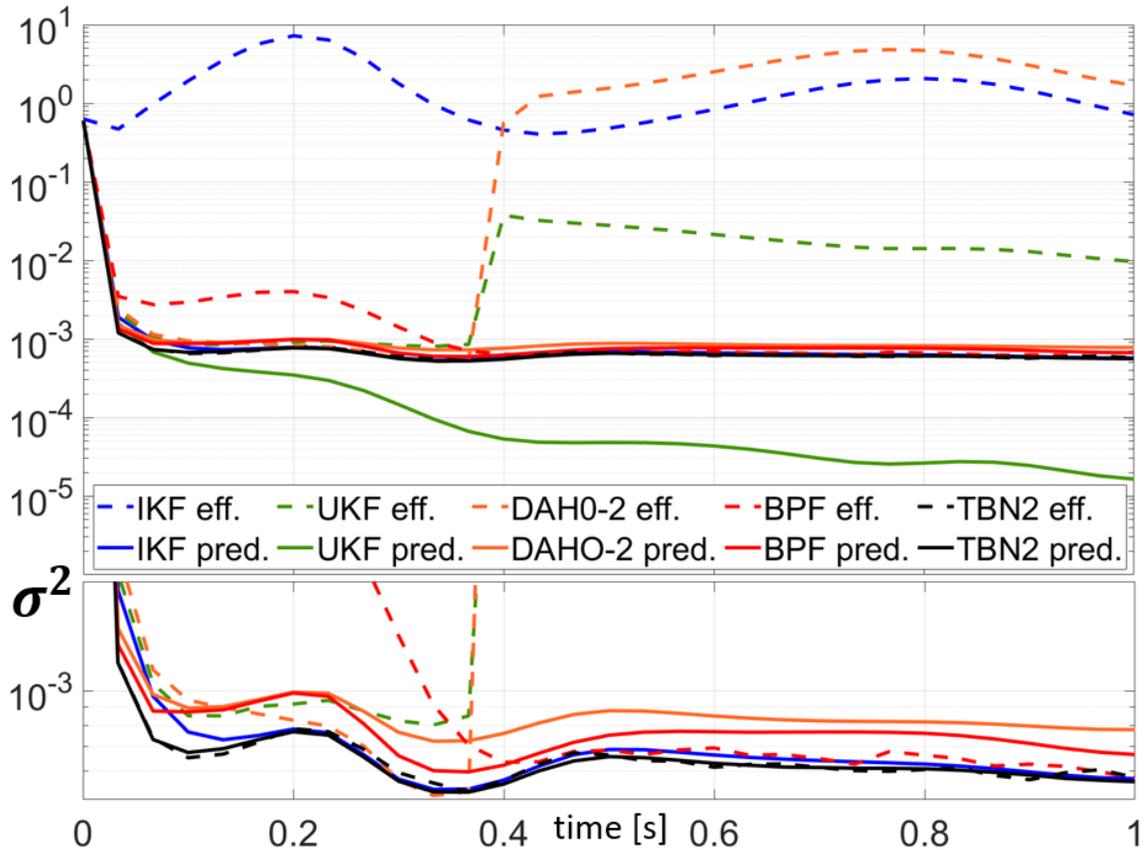


Figure 3. Error covariance comparison analysis of the square Lorenz63 problem among different filters. Monte Carlo performed with 1000 runs for 1s.

Non-Square Measurement Map

A more challenging application is now offered. In the previous example, the filter was receiving a number of measurements equal to the number of states. However, this particular condition limits the family of problems that the filters can solve. Therefore, a new analysis is here presented, where the TBN c receives only bearing angles measurements, without range. That is, the new application resemble the previous one in all the aspects, but the new measurement model is missing Equation (33). Consequently, by reducing the amount of information the filter receives about the state of the system, the performance is expected to decrease.

Figure 4 reports the Monte Carlo analysis results. Once again, the figure shows both the effective

covariance σ_{eff}^2 as dashed lines, and the filter's estimate of the state covariance σ_{pred}^2 as continuous lines. The result shows a similar pattern with the previous application, assessing the reliability of the TBNc filter for non square measurements maps. More in detail, the angles-only application has a longer transient with respect to the application with range measurements. This aspect is expected since the filter is receiving fewer information than before. By looking at each different filter separately, it can be noted that the IKF diverges from the beginning, having the σ_{eff}^2 (dashed blue) line increasing from the first iteration, out of the grid of the figure. The simple linearization in the IKF prediction step does not propagate central moments accurately enough for the filter to achieve a correct tracking. The UKF, green, and the DAHO-2, orange, reach steady state levels with an analogous behavior, but they both diverge around the same time step. This divergence is due those runs that start the farthest from the initial estimate mean: the filter is initially able to reduce the uncertainties, but then the nonlinearities of the system prevails and the tracking fails. Without range measurement, the filter has problems on selecting which lobe of the Lorenz's attractor to follow, due to the pathway symmetry with respect to the origin. Thus, in few Monte Carlo runs, the filter erroneously track the state of the system in the wrong lobe. The main difference with respect to the previous case comes from the BPF. The BPF achieves convergence at steady state, but it shows numerical problems during the transient. Hence, the dashed red line gets interrupted due to numerical issues during the most challenging runs which, in turns, breaks Equation (35) and (37). Therefore, even if the BPF is a consistent filter at steady state, knowledge of the state during the whole simulation is not achieved. Lastly, the TBN2, in black, is once again the only filter that achieves convergence and consistency correctly for the whole simulation. The black lines overlap during both the transient phase and at steady state, setting the filter accuracy level with the most accurate performance.

CONCLUSIONS

A new type of filtering technique has been presented. The update step is performed by mapping the measurement noise PDF, centered at the given outcome from the sensors, back into the *a priori* distribution of the state. The estimate is therefore obtained by studying the fraction that the noise distribution highlights in the prior. When compared to previous techniques [10], the TBNc is able to perform the update each time a measurement becomes available, without the drawback of waiting for the number of measurements to be equal to the number of states. This feature is achieved thanks to the particle nature of the filter, where the TBNc creates fictitious measurements that square the measurement map, making it invertible.

Two numerical applications have been selected to test the performance of TBNc. In both cases, either square and non-square mapping, the TBN2 has shown high accuracy levels, with a coherent and consistent estimate of the state error covariance standard deviation. The overall performance is superior when compared to other nonlinear filtering techniques common in the current state of the art.

REFERENCES

- [1] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, Vol. 82, No. Series D, March 1960, pp. 35–45, doi:10.1115/1.3662552.
- [2] Kalman, R. E. and Bucy, R. S., "New Results in Linear Filtering and Prediction," *Journal of Basic Engineering*, Vol. 83, No. Series D, March 1961, pp. 95–108, doi:10.1115/1.3658902.
- [3] Gelb, A., editor, *Applied Optimal Estimation*, The MIT press, Cambridge, MA, 1974, ISBN:9780262200271.

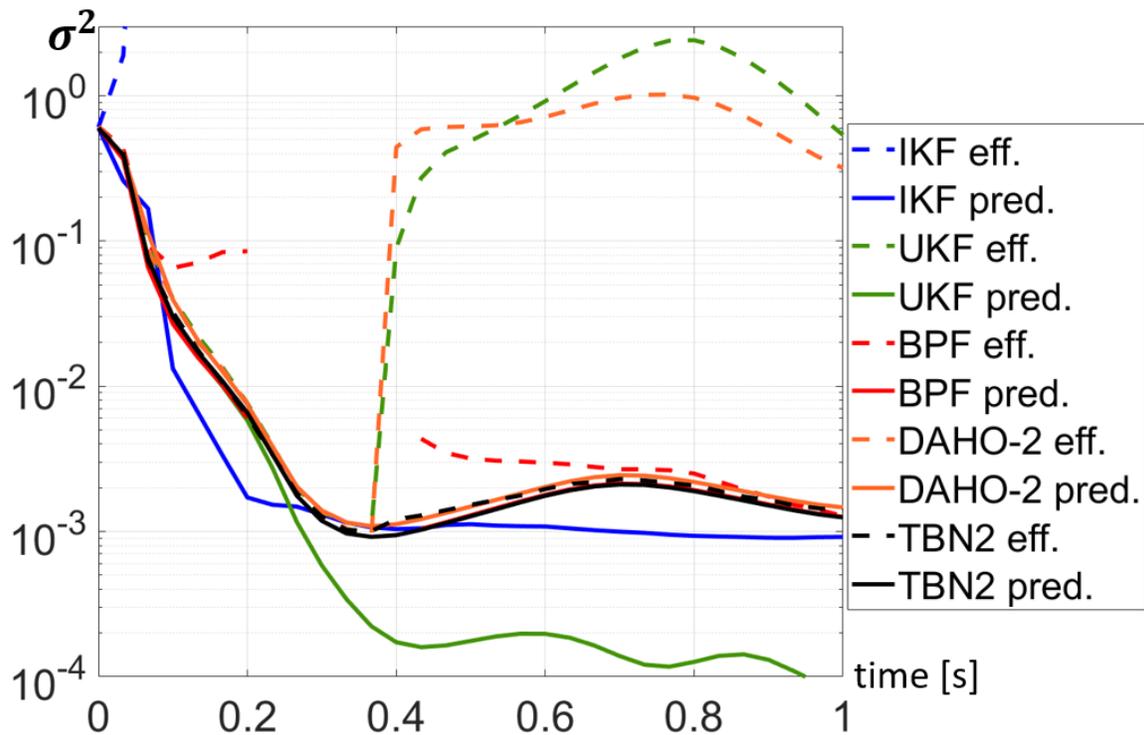


Figure 4. Error covariance comparison analysis of the non-square Lorenz63 problem among different filters. Monte Carlo performed with 1000 runs for 1s.

- [4] Junkins, J. and Singla, P., “How Nonlinear is it? A tutorial on Nonlinearity of Orbit and Attitude Dynamics,” *Journal of the Astronautical Sciences*, Vol. 52, No. 1-2, 2004, pp. 7–60.
- [5] Julier, S. J. and Uhlmann, J. K., “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, Vol. 92, No. 3, March 2004, pp. 401–422, doi: 10.1109/JPROC.2003.823141.
- [6] Julier, S. J., 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*, Vol. 45, No. 3, March 2000, pp. 477–482, doi: 10.1109/9.847726.
- [7] Sorenson, H. W. and Alspach, D. L., “Recursive Bayesian Estimation Using Gaussian Sums,” *Automatica*, Vol. 7, No. 4, July 1971, pp. 465–479, doi:10.1016/0005-1098(71)90097-5.
- [8] Alspach, D. and Sorenson, H., “Nonlinear Bayesian estimation using Gaussian sum approximations,” *IEEE Transactions on Automatic Control*, Vol. 17, No. 4, August 1972, pp. 439–448, doi: 10.1109/SAP.1970.270017.
- [9] Ristic, B., Arulampalam, S., and Gordon, N., *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Radar Library, Artech House, 2004.
- [10] Armellin, R., Di Lizia, P., and Zanetti, R., “Dealing with uncertainties in angles-only initial orbit determination,” *Celestial Mechanics and Dynamical Astronomy*, Vol. 125, No. 4, 2016, pp. 435–450.
- [11] Massari, M., Di Lizia, P., Cavenago, F., and Wittig, A., “Differential Algebra software library with automatic code generation for space embedded applications,” *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, 2018, p. 0398, doi: 10.2514/6.2018-0398.
- [12] Berz, M., *Differential Algebraic Techniques*, Entry in *Handbook of Accelerator Physics and Engineering*, World Scientific, New York, 1999.
- [13] Rasotto, M., Morselli, A., Wittig, A., Massari, M., Di Lizia, P., Armellin, R., Valles, C., and Ortega, G., “Differential algebra space toolbox for nonlinear uncertainty propagation in space dynamics,” 2016.
- [14] Servadio, S. et al., *New developments in nonlinear filtering using differential algebra*, Ph.D. thesis, 2021.
- [15] Servadio, S., Zanetti, R., and Jones, B. A., “Nonlinear Filtering with a Polynomial Series of Gaussian Random Variables,” *IEEE Transactions on Aerospace and Electronic Systems*, 2020, doi: 10.1109/TAES.2020.3028487.

- [16] 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*, 2018, p. 1964, doi: 10.2514/6.2018-1964.
- [17] Servadio, S. and Zanetti, R., “Recursive Polynomial Minimum Mean-Square Error Estimation with Applications to Orbit Determination,” *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 5, 2020, pp. 939–954, <https://doi.org/10.2514/1.G004544>.
- [18] Wittig, A., Di Lizia, P., Armellin, R., Makino, K., Bernelli-Zazzera, F., and Berz, M., “Propagation of large uncertainty sets in orbital dynamics by automatic domain splitting,” *Celestial Mechanics and Dynamical Astronomy*, Vol. 122, No. 3, 2015, pp. 239–261, <https://doi.org/10.1007/s10569-015-9618-3>.
- [19] 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, 1918, pp. 134–139.
- [20] Berz, M., *Modern Map Methods in Particle Beam Physics*, Academic Press, Sec 2.3.1, <https://bt.pa.msu.edu/pub/papers/AIEP108book/AIEP108book.pdf>.
- [21] Terejanu, G., Singla, P., Singh, T., and Scott, P. D., “Adaptive Gaussian Sum Filter for Nonlinear Bayesian Estimation,” *IEEE Transactions on Automatic Control*, Vol. 56, No. 9, 2011, pp. 2151–2156, doi: 10.1109/TAC.2011.2141550.
- [22] Raihan, D. and Chakravorty, S., “Particle Gaussian mixture filters-I,” *Automatica*, Vol. 98, 2018, pp. 331–340, <https://doi.org/10.1016/j.automatica.2018.07.023>.
- [23] Valli, M., Armellin, R., Di Lizia, P., and Lavagna, M. R., “Nonlinear filtering methods for spacecraft navigation based on differential algebra,” *Acta Astronautica*, Vol. 94, No. 1, 2014, pp. 363–374, doi: 10.1016/j.actaastro.2013.03.009.