

# RECURSIVE POLYNOMIAL MINIMUM MEAN SQUARE ERROR ESTIMATION WITH APPLICATIONS TO ORBIT DETERMINATION

Simone Servadio\* and Renato Zanetti†

This paper presents a systematic generalization of the linear update structure associated with the extended Kalman filtering for high order polynomial estimation of nonlinear dynamical systems. A minimum mean-square error criterion is used as a cost function to determine the optimal gains required for the estimation process. The high order series representation is implemented effectively using Differential Algebra techniques.

## INTRODUCTION

The optimal solution to the sequential stochastic Minimum Mean Square Error (MMSE) estimation problem is well known: the optimal estimate is the conditional mean. The optimal solution is obtained from the conditional probability density function which is calculated recursively with Baye's rule. For the linear/Gaussian case, the conditional distribution remains Gaussian at all times and hence fully described by its mean and covariance matrix that can be calculated using Kalman's mechanization.<sup>1,2</sup> For practical nonlinear/non-Gaussian problems, such as orbit determination,<sup>3</sup> the analytical solution is usually not available in closed form.

The most widely used nonlinear estimator is the extended Kalman filter (EKF).<sup>4</sup> The EKF linearizes the estimation error around the most current estimate and applies the Kalman filter equations to this linearized system. It has been shown, however, that nonlinearities of the orbit determination problem can make the linearization assumption insufficient to represent the actual uncertainty.<sup>5</sup> An alternative to linearization around the mean is stochastic linearization in the so-called unscented Kalman filter (UKF).<sup>6,7</sup> The UKF is a linear estimator, i.e. the estimate is a linear function of the measurement.

In some case, the uncertainty associated with orbital mechanics can be propagated analytically;<sup>8</sup> however these analytical solutions usually do not include perturbations other than  $J_2$ , nor process noise or measurement updates. Park and Scheeres<sup>9</sup> use state transition tensors (STT) to propagate mean and higher order central moments through arbitrary nonlinear dynamics. They subsequently expand their work to create higher order Kalman filters able to handle process noise and measurement updates,<sup>10</sup> their filter is also a linear estimator. In Ref.<sup>10</sup> Park and Scheeres only update the mean and covariance when a measurement becomes available, neglecting the contribution the measurement update has on the higher order central moments. Majji, Turner, and Junkins,<sup>11</sup> on the other hand, introduce a tensorial mechanization that expands the work by Park and Scheeres to include measurement updates in all the higher order central moments. Valli *et al.*<sup>12</sup> efficiently replicate these results using differential algebra (DA) techniques.

The EKF, UKF, and the higher order filter proposed by Majji *et al.* utilize a linear update, while it is known that a nonlinear update provides better performance in the nonlinear/non-Gaussian case. The optimal update is the conditional mean, which is given by some nonlinear function of the measurement whose calculation is usually not tractable. One approach to approximate the optimal nonlinear update is breaking the estimation error distribution in many smaller Gaussian components such that each is small enough to satisfy the linearization assumption of an EKF; this is the basis of the Gaussian sum filter.<sup>13</sup> Another approach is to

\*Ph.D. Student, Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin

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

approximate the nonlinear function with a Taylor series.<sup>4</sup> Truncating this series to first order produces the EKF. Generally, the higher the order of the Taylor series the better the performance of the filter; however this approach has two drawbacks. First is that truncating the Taylor series to order  $N$  requires knowledge of the estimation error's central moments up to order  $2N$ . For example the EKF truncates to first order and requires knowledge of the covariance. A second order filter requires knowledge of central moments up to order four. To avoid carrying third and fourth order central moments the Gaussian second order filter (GSOF)<sup>14</sup> approximates them assuming the distribution is Gaussian and constructs them from the covariance matrix.

The second potential drawback of this approach is that calculating the higher order gains is no trivial matter, although it is an operation that is required only once. The GSOF avoids this operation all together by producing a linear update based on a second order approximation of the posterior estimation error. To avoid these calculations, De Santis *et al.*<sup>15</sup> use an augmented state to obtain a nonlinear update but preserving the linear update structure. The work focus on linear but non-Gaussian systems, and on approximations of the optimal non-linear update as either quadratic<sup>15</sup> or polynomial.<sup>16</sup> While maintaining the exact same structure as the traditional Kalman filter, this approach takes the nonlinearity of the measurement update into consideration by implicitly accounting for high order moments in the estimation process. The augmented state is comprised of the high order raw moments. In the presence of non-linear dynamics and measurements, these nonlinear functions are approximated with an arbitrary order Taylor series expansion.

This work introduces a novel nonlinear filter that performs a polynomial update of arbitrary order  $N$  and carries an arbitrary number of central moments  $M \geq 2N$ . The mean and central moments are propagated following the work of Park and Scheeres.<sup>9</sup> The polynomial update is obtained using an augmented measurement approach; however, unlike De Santis *et al.*, the state vector is not augmented and all the central moments are updated independently.

## MATHEMATICAL FORMULATION

Let upper case bold indicate a matrix while non-bold quantities are scalars, for example, matrix  $M$  has entries  $M_{ij}$ . The kronecker product is used and denoted by  $\otimes$

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} A_{11}\mathbf{B} & A_{12}\mathbf{B} & \dots & A_{1m}\mathbf{B} \\ A_{21}\mathbf{B} & A_{22}\mathbf{B} & \dots & A_{2m}\mathbf{B} \\ \dots & \dots & \dots & \dots \\ A_{n1}\mathbf{B} & A_{n2}\mathbf{B} & \dots & A_{nm}\mathbf{B} \end{bmatrix}$$

The following properties are useful

$$\begin{aligned} (\mathbf{A} + \mathbf{B}) \otimes \mathbf{C} &= \mathbf{A} \otimes \mathbf{C} + \mathbf{B} \otimes \mathbf{C} \\ \mathbf{A} \otimes (\mathbf{B} + \mathbf{C}) &= \mathbf{A} \otimes \mathbf{B} + \mathbf{A} \otimes \mathbf{C} \\ (\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C} &= \mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C}) \\ (\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) &= \mathbf{AC} \otimes \mathbf{BD} \end{aligned}$$

where the matrix product is always executed first, the kronecker product second and addition third. Let  $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$  be vectors

$$\begin{aligned} \mathbf{a} \otimes \mathbf{b} &= \text{vect}(\mathbf{ba}^T) \\ \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c} &= \text{vect}(\mathbf{c}(\mathbf{a} \otimes \mathbf{b})^T) \\ \mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c} \otimes \mathbf{d} &= \text{vect}(\mathbf{d}(\mathbf{c} \otimes \mathbf{a} \otimes \mathbf{b})^T) \\ (\mathbf{a} \otimes \mathbf{b})\mathbf{c}^T &= \mathbf{a} \otimes \mathbf{bc}^T = \mathbf{ac}^T \otimes \mathbf{b} \end{aligned}$$

where  $\text{vect}(\mathbf{M})$  is the operator that vectorizes matrix  $\mathbf{M}$  by placing all its columns on top of each other.

## Minimum Mean Square Error Estimation

Given two random vectors  $\mathbf{x}$  and a  $\mathbf{y}$ , an estimator  $\hat{\mathbf{x}}$  of  $\mathbf{x}$  from  $\mathbf{y}$  is simply a function of  $\mathbf{y}$  that tries to approximate  $\mathbf{x}$ :

$$\hat{\mathbf{x}} = \mathbf{g}(\mathbf{y}). \quad (1)$$

The estimation error  $\mathbf{e}$  is a random vector defined as

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}} = \mathbf{x} - \mathbf{g}(\mathbf{y}) \quad (2)$$

The Minimum Mean Square Error (MMSE) estimate minimizes (on average) the mean-square error

$$\min_{\mathbf{g}(\cdot)} \mathbb{E}_{\mathbf{x}\mathbf{y}} \{ (\mathbf{x} - \mathbf{g}(\mathbf{y}))^T (\mathbf{x} - \mathbf{g}(\mathbf{y})) \} = \int_{\mathcal{S}(\mathbf{x}\mathbf{y})} (\mathbf{x} - \mathbf{g}(\mathbf{y}))^T (\mathbf{x} - \mathbf{g}(\mathbf{y})) p_{\mathbf{x}\mathbf{y}}(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} \quad (3)$$

where  $\mathcal{S}(\mathbf{x}\mathbf{y})$  indicates the support of  $\mathbf{x}$  and  $\mathbf{y}$ . The optimal solution is well known and given by the conditional mean

$$\hat{\mathbf{x}} = \mathbb{E}_{\mathbf{x}|\mathbf{y}} \{ \mathbf{x} \} \quad (4)$$

Notice that some authors prefer to define the performance index using the conditional expectation

$$\min_{\mathbf{g}(\cdot)} \mathbb{E}_{\mathbf{x}|\mathbf{y}} \{ (\mathbf{x} - \mathbf{g}(\mathbf{y}))^T (\mathbf{x} - \mathbf{g}(\mathbf{y})) \} \quad (5)$$

this is due to the definition of Bayesian estimator, which is an estimator that optimizes some function of the posterior distribution. The two formulations are equivalent and provide the same result, but the former is preferred here because is more easily extended to Linear Minimum Mean Square Error (LMMSE) estimation.

Suppose the measurement is partitioned in two parts  $\mathbf{y}_1$  and  $\mathbf{y}_2$

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_2 \\ \mathbf{y}_1 \end{bmatrix} \quad (6)$$

the MMSE estimate and the Bayes update can be calculated recursively when  $p_{\mathbf{y}_2|\mathbf{x}}$  and  $p_{\mathbf{y}_1|\mathbf{x}}$  are independent, i.e. when

$$p_{\mathbf{y}_2, \mathbf{y}_1|\mathbf{x}} = p_{\mathbf{y}_2|\mathbf{x}} p_{\mathbf{y}_1|\mathbf{x}} \quad (7)$$

This condition is satisfied when, for example, the measurement noise  $\boldsymbol{\eta}_k$  is white:

$$\mathbf{y}_2 = \mathbf{h}_2(\mathbf{x}, \boldsymbol{\eta}_2) \quad (8)$$

$$\mathbf{y}_1 = \mathbf{h}_1(\mathbf{x}, \boldsymbol{\eta}_1) \quad (9)$$

$$p_{\boldsymbol{\eta}_1 \boldsymbol{\eta}_2}(\boldsymbol{\eta}_1, \boldsymbol{\eta}_2) = p_{\boldsymbol{\eta}_1}(\boldsymbol{\eta}_1) p_{\boldsymbol{\eta}_2}(\boldsymbol{\eta}_2) \quad (10)$$

where  $\mathbf{h}_2$  and  $\mathbf{h}_1$  are nonlinear functions.

Under this assumption the posterior is given by

$$p_{\mathbf{x}|\mathbf{y}_2, \mathbf{y}_1} = \frac{p_{\mathbf{y}_2|\mathbf{x}, \mathbf{y}_1} p_{\mathbf{x}|\mathbf{y}_1}}{\int p_{\mathbf{y}_2|\mathbf{x}, \mathbf{y}_1} p_{\mathbf{x}|\mathbf{y}_1} d\mathbf{y}_2} \quad (11)$$

but

$$p_{\mathbf{y}_2|\mathbf{x}, \mathbf{y}_1} = \frac{p_{\mathbf{y}_2, \mathbf{y}_1|\mathbf{x}}}{p_{\mathbf{y}_1|\mathbf{x}}} = \frac{p_{\mathbf{y}_2|\mathbf{x}} p_{\mathbf{y}_1|\mathbf{x}}}{p_{\mathbf{y}_1|\mathbf{x}}} = p_{\mathbf{y}_2|\mathbf{x}} \quad (12)$$

hence

$$p_{\mathbf{x}|\mathbf{y}_2, \mathbf{y}_1} = \frac{p_{\mathbf{y}_2|\mathbf{x}} p_{\mathbf{x}|\mathbf{y}_1}}{\int p_{\mathbf{y}_2|\mathbf{x}} p_{\mathbf{x}|\mathbf{y}_1} d\mathbf{y}_2} \quad (13)$$

Therefore the measurement incorporation can be done recursively, first  $\mathbf{y}_1$

$$p_{\mathbf{x}|\mathbf{y}_1} = \frac{p_{\mathbf{y}_1|\mathbf{x}} p_{\mathbf{x}}}{\int p_{\mathbf{y}_1|\mathbf{x}} p_{\mathbf{x}} d\mathbf{y}_1} \quad (14)$$

after this step the posterior  $p_{\mathbf{x}|\mathbf{y}_1}$  becomes the new prior and the starting point to incorporate  $\mathbf{y}_2$  with Eq. (13). We have obtained a recursive algorithm. It is important to emphasize that the algorithm carries and recursively updates the conditional distribution, whose mean is the MMSE estimate. The total covariance of the estimation error (not conditioned on the measurement) is neither needed nor calculated, although it is equal to the minimized performance index.

Consider the following linear dynamics and linear measurement

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t) \mathbf{x}(t) + \mathbf{G}(t) \boldsymbol{\nu}(t) \quad (15)$$

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}(t_k) + \boldsymbol{\eta}_k \quad (16)$$

where  $\mathbf{x}(t_0) \sim \mathcal{N}(\hat{\mathbf{x}}_{0|0}, \mathbf{P}_{0|0})$  and  $\boldsymbol{\nu}(t)$  is zero mean Gaussian white noise with power spectral density  $\mathbf{Q}$ . The process noise is uncorrelated from  $\mathbf{x}(t_0)$  and  $\boldsymbol{\eta}_k$  is zero mean Gaussian white sequence with covariance matrix  $\mathbf{R}_k$ . The measurement noise is uncorrelated from the process noise and from  $\mathbf{x}(t_0)$ . Under this linear/Gaussian case, the conditional distribution of  $\mathbf{x}|\mathbf{y}$  remains Gaussian at all times, therefore the conditional mean and the conditional covariance matrix fully describe the conditional distribution. The conditional mean (i.e. the MMSE estimate) and conditional covariance matrix can be found recursively using the Kalman filter equations. Under these very special linear/Gaussian conditions, the covariance matrix obtained from the Kalman filter equations is both the conditional covariance of the state given the measurements and the total covariance of the estimation error, i.e.

$$\mathbb{E}_{\mathbf{x}\mathbf{y}} \{ (\mathbf{x} - \mathbb{E} \{ \mathbf{x} | \mathbf{y} \}) (\mathbf{x} - \mathbb{E} \{ \mathbf{x} | \mathbf{y} \})^T \} = \mathbb{E}_{\mathbf{x}|\mathbf{y}} \{ (\mathbf{x} - \mathbb{E} \{ \mathbf{x} | \mathbf{y} \}) (\mathbf{x} - \mathbb{E} \{ \mathbf{x} | \mathbf{y} \})^T \} \quad (17)$$

This unusual property is unique to jointly Gaussian  $\mathbf{x}$  and  $\mathbf{y}$  and simply states that the covariance of  $\mathbf{x}$  conditioned on  $\mathbf{y}$  is not a function of  $\mathbf{y}$ , that is to say, it's the same regardless of the value of  $\mathbf{y}$ .

### Linear Minimum Mean Square Error Estimation

Rather than working on the entire space of any possible function, one can limit the search to affine functions. The family  $\mathbf{g}$  of estimators is then written as

$$\mathbf{g}(\mathbf{y}) = \mathbf{a} + \mathbf{B}\mathbf{y} \quad (18)$$

where  $\mathbf{B}$  and  $\mathbf{a}$  are deterministic matrices and vectors, respectively, of appropriate dimensions. The optimal values are well known and given by

$$\mathbf{a} = \mathbb{E}_{\mathbf{x}} \{ \mathbf{x} \} - \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{y}} \boldsymbol{\Sigma}_{\mathbf{y}\mathbf{y}}^{-1} \mathbb{E}_{\mathbf{y}} \{ \mathbf{y} \}. \quad (19)$$

$$\mathbf{B} = \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{y}} \boldsymbol{\Sigma}_{\mathbf{y}\mathbf{y}}^{-1}. \quad (20)$$

where

$$\boldsymbol{\Sigma}_{\mathbf{x}\mathbf{y}} = \mathbb{E}_{\mathbf{x}\mathbf{y}} \{ (\mathbf{x} - \mathbb{E}_{\mathbf{x}} \{ \mathbf{x} \}) (\mathbf{y} - \mathbb{E}_{\mathbf{y}} \{ \mathbf{y} \})^T \} \quad (21)$$

$$\boldsymbol{\Sigma}_{\mathbf{y}\mathbf{y}} = \mathbb{E}_{\mathbf{y}} \{ (\mathbf{y} - \mathbb{E}_{\mathbf{y}} \{ \mathbf{y} \}) (\mathbf{y} - \mathbb{E}_{\mathbf{y}} \{ \mathbf{y} \})^T \} \quad (22)$$

Generally speaking, the LMMSE estimate cannot be calculated recursively. However, for linear systems (either Gaussian or not) the LMMSE is the Kalman filter and it is recursive. Once again, the key assumption is that all noises are white and independent from all other distributions.

Assume  $\boldsymbol{\eta}_1$  is independent from  $\boldsymbol{\eta}_2$

$$\mathbf{y}_1 = \mathbf{H}_1 \mathbf{x} + \boldsymbol{\eta}_1 \quad (23)$$

$$\mathbf{y}_2 = \mathbf{H}_2 \mathbf{x} + \boldsymbol{\eta}_2 \quad (24)$$

denote with  $\hat{\boldsymbol{x}}_0$  the mean of  $\boldsymbol{x}$  and with  $\boldsymbol{P}_0$  its covariance matrix. The LMMSE estimate is given by

$$\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_0 + \boldsymbol{P}_0 \begin{bmatrix} \boldsymbol{H}_1^\top & \boldsymbol{H}_2^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{H}_1 \boldsymbol{P}_0 \boldsymbol{H}_1^\top + \boldsymbol{R}_1 & \boldsymbol{H}_1 \boldsymbol{P}_0 \boldsymbol{H}_2^\top \\ \boldsymbol{H}_2 \boldsymbol{P}_0 \boldsymbol{H}_1^\top & \boldsymbol{H}_2 \boldsymbol{P}_0 \boldsymbol{H}_2^\top + \boldsymbol{R}_2 \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{y}_1 - \boldsymbol{H}_1 \hat{\boldsymbol{x}}_0 \\ \boldsymbol{y}_2 - \boldsymbol{H}_2 \hat{\boldsymbol{x}}_0 \end{bmatrix} \quad (25)$$

the matrix to be inverted is

$$\begin{bmatrix} \boldsymbol{H}_1 \boldsymbol{P}_0 \boldsymbol{H}_1^\top + \boldsymbol{R}_1 & \boldsymbol{H}_1 \boldsymbol{P}_0 \boldsymbol{H}_2^\top \\ \boldsymbol{H}_2 \boldsymbol{P}_0 \boldsymbol{H}_1^\top & \boldsymbol{H}_2 \boldsymbol{P}_0 \boldsymbol{H}_2^\top + \boldsymbol{R}_2 \end{bmatrix}^{-1} \quad (26)$$

$$= \begin{bmatrix} \boldsymbol{W}_1 & \boldsymbol{B} \\ \boldsymbol{B}^\top & \boldsymbol{D} \end{bmatrix}^{-1} \quad (27)$$

$$= \begin{bmatrix} \boldsymbol{W}_1^{-1} + \boldsymbol{W}_1^{-1} \boldsymbol{B} (\boldsymbol{D} - \boldsymbol{B}^\top \boldsymbol{W}_1^{-1} \boldsymbol{B})^{-1} \boldsymbol{B}^\top \boldsymbol{W}_1^{-1} & -\boldsymbol{W}_1^{-1} \boldsymbol{B} (\boldsymbol{D} - \boldsymbol{B}^\top \boldsymbol{W}_1^{-1} \boldsymbol{B})^{-1} \\ -(\boldsymbol{D} - \boldsymbol{B}^\top \boldsymbol{W}_1^{-1} \boldsymbol{B})^{-1} \boldsymbol{B}^\top \boldsymbol{W}_1^{-1} & (\boldsymbol{D} - \boldsymbol{B}^\top \boldsymbol{W}_1^{-1} \boldsymbol{B})^{-1} \end{bmatrix} \quad (28)$$

$$= \begin{bmatrix} \boldsymbol{W}_1^{-1} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} \end{bmatrix} + \begin{bmatrix} \boldsymbol{W}_1^{-1} \boldsymbol{B} \\ -\boldsymbol{I} \end{bmatrix} (\boldsymbol{D} - \boldsymbol{B}^\top \boldsymbol{W}_1^{-1} \boldsymbol{B})^{-1} \begin{bmatrix} \boldsymbol{W}_1^{-1} \boldsymbol{B} \\ -\boldsymbol{I} \end{bmatrix}^\top \quad (29)$$

$$\begin{aligned} \hat{\boldsymbol{x}} &= \hat{\boldsymbol{x}}_0 + \boldsymbol{P}_0 \boldsymbol{H}_1^\top \boldsymbol{W}_1^{-1} (\boldsymbol{y}_1 - \boldsymbol{H}_1 \hat{\boldsymbol{x}}_0) + \\ &+ \boldsymbol{P}_0 (\boldsymbol{H}_1^\top \boldsymbol{W}_1^{-1} \boldsymbol{B} - \boldsymbol{H}_2^\top) (\boldsymbol{D} - \boldsymbol{B}^\top \boldsymbol{W}_1^{-1} \boldsymbol{B})^{-1} (\boldsymbol{B}^\top \boldsymbol{W}_1^{-1} (\boldsymbol{y}_1 - \boldsymbol{H}_1 \hat{\boldsymbol{x}}_0) - \boldsymbol{y}_2 + \boldsymbol{H}_2 \hat{\boldsymbol{x}}_0) \end{aligned} \quad (30)$$

Let's define

$$\hat{\boldsymbol{x}}_1 = \hat{\boldsymbol{x}}_0 + \boldsymbol{P}_0 \boldsymbol{H}_1^\top \boldsymbol{W}_1^{-1} (\boldsymbol{y}_1 - \boldsymbol{H}_1 \hat{\boldsymbol{x}}_0) \quad (31)$$

and notice that

$$\boldsymbol{B}^\top \boldsymbol{W}_1^{-1} (\boldsymbol{y}_1 - \boldsymbol{H}_1 \hat{\boldsymbol{x}}_0) - \boldsymbol{y}_2 + \boldsymbol{H}_2 \hat{\boldsymbol{x}}_0 = \boldsymbol{H}_2 \boldsymbol{P}_0 \boldsymbol{H}_1^\top \boldsymbol{W}_1^{-1} (\boldsymbol{y}_1 - \boldsymbol{H}_1 \hat{\boldsymbol{x}}_0) - \boldsymbol{y}_2 + \boldsymbol{H}_2 \hat{\boldsymbol{x}}_0 \quad (32)$$

$$= -(\boldsymbol{y}_2 - \boldsymbol{H}_2 \hat{\boldsymbol{x}}_1) \quad (33)$$

so that

$$\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_1 + \boldsymbol{P} \left( -\boldsymbol{H}_1^\top \boldsymbol{W}_1^{-1} \boldsymbol{B} + \boldsymbol{H}_2^\top \right) (\boldsymbol{D} - \boldsymbol{B}^\top \boldsymbol{W}_1^{-1} \boldsymbol{B})^{-1} (\boldsymbol{y}_2 - \boldsymbol{H}_2 \hat{\boldsymbol{x}}_1) \quad (34)$$

and

$$\boldsymbol{P}_0 \left( -\boldsymbol{H}_1^\top \boldsymbol{W}_1^{-1} \boldsymbol{B} + \boldsymbol{H}_2^\top \right) = \boldsymbol{P}_0 \boldsymbol{H}_2^\top - \boldsymbol{P}_0 \boldsymbol{H}_1^\top \boldsymbol{W}_1^{-1} \boldsymbol{H}_1 \boldsymbol{P}_0 \boldsymbol{H}_2^\top \quad (35)$$

$$= \left( \boldsymbol{P}_0 - \boldsymbol{P}_0 \boldsymbol{H}_1^\top \boldsymbol{W}_1^{-1} \boldsymbol{H}_1 \boldsymbol{P}_0 \right) \boldsymbol{H}_2^\top \quad (36)$$

$$\boldsymbol{D} - \boldsymbol{B}^\top \boldsymbol{W}_1^{-1} \boldsymbol{B} = \boldsymbol{H}_2 \boldsymbol{P}_0 \boldsymbol{H}_2^\top + \boldsymbol{R}_2 - \boldsymbol{H}_2 \boldsymbol{P}_0 \boldsymbol{H}_1^\top \boldsymbol{W}_1^{-1} \boldsymbol{H}_1 \boldsymbol{P}_0 \boldsymbol{H}_2^\top \quad (37)$$

$$= \boldsymbol{H}_2 \left( \boldsymbol{P}_0 - \boldsymbol{P}_0 \boldsymbol{H}_1^\top \boldsymbol{W}_1^{-1} \boldsymbol{H}_1 \boldsymbol{P}_0 \right) \boldsymbol{H}_2^\top + \boldsymbol{R}_2 \quad (38)$$

we can define

$$\boldsymbol{P}_1 = \boldsymbol{P}_0 - \boldsymbol{P}_0 \boldsymbol{H}_1^\top \boldsymbol{W}_1^{-1} \boldsymbol{H}_1 \boldsymbol{P}_0 \quad (39)$$

Therefore the total LMMSE estimate can be written recursively can be obtained for linear systems, regardless of the errors distribution.

$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_k + \boldsymbol{P}_k \boldsymbol{H}_k^\top \left( \boldsymbol{H}_{k+1} \boldsymbol{P}_k \boldsymbol{H}_{k+1}^\top + \boldsymbol{R}_{k+1} \right)^{-1} (\boldsymbol{y}_{k+1} - \boldsymbol{H}_{k+1} \hat{\boldsymbol{x}}_k) \quad (40)$$

$$\boldsymbol{P}_{k+1} = \boldsymbol{P}_k - \boldsymbol{P}_k \boldsymbol{H}_k^\top \left( \boldsymbol{H}_{k+1} \boldsymbol{P}_k \boldsymbol{H}_{k+1}^\top + \boldsymbol{R}_{k+1} \right)^{-1} \boldsymbol{H}_{k+1} \boldsymbol{P}_k \quad (41)$$

These are the Kalman filter update equations. In the linear measurement case, when distributions are not Gaussian, the Kalman filter equations, in their recursive form, provide the LMMSE estimate. The estimate  $\hat{x}_k$  is not the conditional mean, it is just the LMMSE estimate. The covariance matrix  $P_k$  is not the conditional covariance matrix anymore, but it still is the covariance of the estimation error  $e = x - \hat{x}_k$ , that is to say

$$P = \mathbb{E}_{xy} \{ (x - \hat{x}) (x - \hat{x})^T \} \neq \mathbb{E}_{x|y} \{ (x - \mathbb{E}\{x|y\}) (x - \mathbb{E}\{x|y\})^T \} \quad (42)$$

The LMMSE has a nice, recursive formulation for linear systems with independent noise but generally it cannot be calculated recursively for nonlinear systems.

### Polynomial MMSE Estimation

It is possible to expand the concept of Linear MMSE to higher order polynomials.<sup>16</sup> Any polynomial function satisfies the conditions of the orthogonality principle, namely closure under addition and scalar multiplication, hence we can calculate an optimal polynomial update using the orthogonality principle.<sup>17</sup>

For LMMSE we needed knowledge of the first two moments (mean and covariance). In general, for a polynomial update of order  $p$  we need knowledge of the first  $2p$  moments. It is possible to use the Kronecker product  $\otimes$  to rewrite polynomial updates as linear ones, e.g. for a quadratic update

$$g(y) = a + B \begin{bmatrix} y \\ y \otimes y \end{bmatrix} = a + BY \quad (43)$$

where  $Y$  is an augmented measurement vector that includes both  $y$  and its square, this approach reduces the quadratic update to a Kalman-filter-like update

$$\hat{x} = \mathbb{E}_x \{x\} + \Sigma_{xY} \Sigma_{YY}^{-1} (Y - \mathbb{E}_y \{Y\}) \quad (44)$$

unfortunately, for non-scalar measurements, the elements of the augmented measurement  $Y$  are not unique, hence  $\Sigma_{YY}$  is not invertible. This is easily alleviated by removing the duplicate elements from  $Y$ . Even when the dynamics and the measurement  $y$  are linear, the system ceases to be linear ( $y \otimes y$  is quadratic in  $x$ ), therefore it also ceases to be recursive. Applying the equations recursively nevertheless, the result is a suboptimal estimate that is not the true quadratic MMSE estimate.

Define  $v^{[2]} = v \otimes v$ . Ref. 15 proposes to reinstate the linearity by also augmenting the state vector

$$X = \begin{bmatrix} x \\ x \otimes x \end{bmatrix} = \begin{bmatrix} x \\ x^{[2]} \end{bmatrix} \quad (45)$$

hence

$$Y = \begin{bmatrix} H & O \\ O & H \otimes H \end{bmatrix} X + \begin{bmatrix} \eta \\ (Hx) \otimes \eta + \eta \otimes (Hx) + \eta^{[2]} \end{bmatrix} = H^{aug} X + \eta^{aug} \quad (46)$$

and

$$\mathbb{E}_y \{Y\} = H^{aug} \mathbb{E}_x \{X\} + \begin{bmatrix} \mathbf{0} \\ \text{vect}(R) \end{bmatrix} \quad (47)$$

$$\Sigma_{XY} = P^{aug} (H^{aug})^T \quad (48)$$

$$\Sigma_{YY} = H^{aug} P^{aug} (H^{aug})^T + R^{aug} \quad (49)$$

$$P^{aug} = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xx^{[2]}} \\ \Sigma_{x^{[2]}x} & \Sigma_{x^{[2]}x^{[2]}} \end{bmatrix} \quad (50)$$

where

$$\mathbf{R}^{aug} = \begin{bmatrix} \mathbf{R} & \mathbf{R}_{12}^{aug} \\ \mathbf{R}_{21}^{aug} & \mathbf{R}_{22}^{aug} \end{bmatrix} \quad (51)$$

$$\mathbf{R}_{21}^{aug} = (\mathbf{H} \otimes \mathbf{I}) (\mathbb{E} \{\mathbf{x}\} \otimes \mathbf{R}) + (\mathbf{I} \otimes \mathbf{H}) (\mathbf{R} \otimes \mathbb{E} \{\mathbf{x}\}) + \mathbb{E} \left\{ \boldsymbol{\eta}^{[2]} \boldsymbol{\eta}^T \right\} \quad (52)$$

$$\mathbf{R}_{12}^{aug} = (\mathbf{R}_{21}^{aug})^T \quad (53)$$

$$\begin{aligned} \mathbf{R}_{22}^{aug} &= (\mathbf{H} \otimes \mathbf{I}) (\mathbb{E} \{\mathbf{x}\mathbf{x}^T\} \otimes \mathbf{R})(\mathbf{H} \otimes \mathbf{I})^T + (\mathbf{I} \otimes \mathbf{H}) (\mathbf{R} \otimes \mathbb{E} \{\mathbf{x}\mathbf{x}^T\}) (\mathbf{I} \otimes \mathbf{H})^T \\ &\quad + \mathbb{E} \left\{ \boldsymbol{\eta}^{[2]} (\boldsymbol{\eta}^{[2]})^T \right\} - \text{vect}(\mathbf{R}) \text{vect}(\mathbf{R})^T \end{aligned} \quad (54)$$

To simplify the above equations, Ref. 15 assumes  $\mathbf{x}$  to be zero mean. Therefore the algorithm carries a propagated mean of  $\mathbf{x}$ , together with an augmented estimate  $\hat{\mathbf{X}}$  of the deviation from the mean and its augmented covariance matrix. The following simplifications occur

$$\mathbf{R}_{21}^{aug} = \mathbb{E} \left\{ \boldsymbol{\eta}^{[2]} \boldsymbol{\eta}^T \right\} \quad (55)$$

$$\begin{aligned} \mathbf{R}_{22}^{aug} &= (\mathbf{H} \otimes \mathbf{I}) (\boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x}} \otimes \mathbf{R})(\mathbf{H} \otimes \mathbf{I})^T + (\mathbf{I} \otimes \mathbf{H}) (\mathbf{R} \otimes \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x}}) (\mathbf{I} \otimes \mathbf{H})^T + \boldsymbol{\Sigma}_{\boldsymbol{\eta}^{[2]} \boldsymbol{\eta}^{[2]}} \\ &= \mathbf{H} \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x}} \mathbf{H}^T \otimes \mathbf{R} + \mathbf{R} \otimes \mathbf{H} \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x}} \mathbf{H}^T + \boldsymbol{\Sigma}_{\boldsymbol{\eta}^{[2]} \boldsymbol{\eta}^{[2]}} \end{aligned} \quad (56)$$

Any attempt to use the math from the prior section to make this quadratic update recursive will run into issues because the “ $\mathbf{R}$ ” matrix is a function of the “ $\mathbf{P}$ ” matrix. Furthermore, in the linear case the recursive LMMSE uses  $\mathbb{E} \{\mathbf{x}\}$  only at initialization, after that the mean of  $\mathbf{x}$  is replaced by the LMMSE estimate, which is not the mean. It is therefore not clear that  $\mathbb{E} \{\mathbf{x}\} = \mathbf{0}$  should result in a simplification of Eq. (52) at any step other than the very first. In computing the measurement covariance with Eq. (54), Ref. 15 certainly replaces  $\mathbb{E} \{\mathbf{x}\mathbf{x}^T\}$  with the recursively updated covariance of the LMMSE; not the un-updated covariance of  $\mathbf{x}$ . It therefore seems that Eq. (52) and Eq. (54) are treated differently in that  $\mathbb{E} \{\mathbf{x}\}$  is always the mean of the true state, unchanged by the successive measurements being incorporated, while  $\mathbb{E} \{\mathbf{x}\mathbf{x}^T\}$  is the spread around the true state which is updated and reduced as more measurements become available. We propose a similar but different approach to this same problem.

## NONLINEAR UPDATES WITH POLYNOMIAL RESIDUALS

Let’s rewrite the quadratic update as

$$\mathbf{g}(\mathbf{y}) = \mathbf{a} + \mathbf{B}\mathbf{y} + \mathbf{C}\mathbf{y} \otimes \mathbf{y} \quad (57)$$

where it is understand that the redundant components of  $\mathbf{y} \otimes \mathbf{y}$  are eliminated. Without any loss of generality we can redefine the coefficients by adding and subtracting constants in order to obtain a different, but equivalent, family of quadratic estimators

$$\mathbf{g}(\mathbf{y}) = \mathbf{a} + \mathbb{E} \{\mathbf{x}\} + \mathbf{B}(\mathbf{y} - \mathbb{E} \{\mathbf{y}\}) + \mathbf{C} [(\mathbf{y} - \mathbb{E} \{\mathbf{y}\}) \otimes (\mathbf{y} - \mathbb{E} \{\mathbf{y}\})] \quad (58)$$

the quantity  $d\mathbf{y} = \mathbf{y} - \mathbb{E} \{\mathbf{y}\}$  is usually referred to as the measurement residual. Similarly, we define state deviation from the mean as  $d\mathbf{x} = \mathbf{x} - \mathbb{E} \{\mathbf{x}\}$ . We know that the optimal values of the estimator’s coefficients (denoted with an asterisk) satisfy the orthogonality principle

$$\mathbb{E} \left\{ \left( d\mathbf{x} - \mathbf{a}^* - \mathbf{B}^* d\mathbf{y} - \mathbf{C}^* (d\mathbf{y} \otimes d\mathbf{y}) \right) \left( \mathbf{a} + \mathbf{B}d\mathbf{y} + \mathbf{C} (d\mathbf{y} \otimes d\mathbf{y}) \right)^T \right\} = \mathbf{O}, \quad \forall \mathbf{a}, \mathbf{B}, \mathbf{C} \quad (59)$$

therefore the optimal coefficients are found solving the following linear system

$$\mathbf{a}^* + \mathbf{C}^* \mathbb{E} \{ (d\mathbf{y} \otimes d\mathbf{y}) \} = \mathbf{0} \quad (60)$$

$$\mathbf{B}^* \mathbf{P}_{d\mathbf{y}d\mathbf{y}} + \mathbf{C}^* \mathbf{P}_{d\mathbf{y}^{[2]}d\mathbf{y}} = \mathbf{P}_{d\mathbf{x}d\mathbf{y}} \quad (61)$$

$$\mathbf{a}^* \mathbf{v}^T + \mathbf{B}^* \mathbf{P}_{d\mathbf{y}d\mathbf{y}^{[2]}} + \mathbf{C}^* \mathbf{P}_{d\mathbf{y}^{[2]}d\mathbf{y}^{[2]}} = \mathbf{P}_{d\mathbf{x}d\mathbf{y}^{[2]}} \quad (62)$$

where

$$\mathbf{v} = \text{vect}(\mathbf{P}_{d\mathbf{y}d\mathbf{y}}) \quad (63)$$

which results in

$$\mathbf{a}^* = -\mathbf{C}^* \mathbf{v} \quad (64)$$

$$[\mathbf{B}^* \quad \mathbf{C}^*] \begin{bmatrix} \mathbf{P}_{d\mathbf{y}d\mathbf{y}} & \mathbf{P}_{d\mathbf{y}^{[2]}d\mathbf{y}} \\ \mathbf{P}_{d\mathbf{y}d\mathbf{y}^{[2]}} & \mathbf{P}_{d\mathbf{y}^{[2]}d\mathbf{y}^{[2]}} - \mathbf{v}\mathbf{v}^T \end{bmatrix} = [\mathbf{P}_{d\mathbf{x}d\mathbf{y}} \quad \mathbf{P}_{d\mathbf{x}d\mathbf{y}^{[2]}}] \quad (65)$$

the posterior estimation error is

$$d\mathbf{x}^+ = \mathbf{x} - \hat{\mathbf{x}} = \mathbf{x} - \mathbf{g}(\mathbf{y}) = \mathbf{x} - \left( \mathbb{E}\{\mathbf{x}\} + \mathbf{B}^* d\mathbf{y} + \mathbf{C}^* d\mathbf{y}^{[2]} - \mathbf{C}^* \mathbf{v} \right) \quad (66)$$

$$= d\mathbf{x} - \mathbf{B}^* d\mathbf{y} - \mathbf{C}^* (d\mathbf{y}^{[2]} - \mathbf{v}) \quad (67)$$

while the optimal estimate is obtained as

$$\hat{\mathbf{x}} = \mathbb{E}\{\mathbf{x}\} + \mathbf{B}^* (\tilde{\mathbf{y}} - \mathbb{E}\{\mathbf{y}\}) + \mathbf{C}^* [(\tilde{\mathbf{y}} - \mathbb{E}\{\mathbf{y}\}) \otimes (\tilde{\mathbf{y}} - \mathbb{E}\{\mathbf{y}\})] - \mathbf{C}^* \mathbf{v}$$

where  $\tilde{\mathbf{y}}$  is the actual (numerical value of the) measurement.

## DIFFERENTIAL ALGEBRA POLYNOMIAL UPDATE FILTER - HODAKF

Consider now the following system, where the state evolves according to a discrete-time nonlinear state transition equation and where the only information about the system is a set of measurements, related to the state vector, acquired at discrete times.

$$\mathbf{x}(k+1) = \mathbf{f}[\mathbf{x}(k)] + \boldsymbol{\nu}(k) \quad (68)$$

$$\mathbf{y}(k+1) = \mathbf{h}[\mathbf{x}(k+1)] + \boldsymbol{\eta}(k+1) \quad (69)$$

where  $\mathbf{f}$  is the process model,  $\mathbf{x}(k)$  is the  $n$ -dimensional state at time-step  $k$ ,  $\mathbf{y}$  is the  $m$ -dimensional vector of the actual measurement at time-step  $k+1$ , and  $\mathbf{h}$  is the measurement function. The process noise  $\boldsymbol{\nu}$  and the measurement noise  $\boldsymbol{\eta}$  are non-Gaussian zero-mean random sequences which satisfy the conditions  $\forall i, j > 0$ :

$$\mathbb{E}\{\boldsymbol{\nu}(i)\} = \mathbb{E}\{\boldsymbol{\eta}(i)\} = 0 \quad (70)$$

$$\mathbb{E}\{\boldsymbol{\nu}(i)\boldsymbol{\nu}^T(j)\} = \mathbf{Q}^{[2]}(i)\delta_{ij} \quad (71)$$

$$\mathbb{E}\{\boldsymbol{\eta}(i)\boldsymbol{\eta}^T(j)\} = \mathbf{R}^{[2]}(i)\delta_{ij} \quad (72)$$

$$\mathbb{E}\{\boldsymbol{\nu}(i)\boldsymbol{\eta}^T(j)\} = 0 \quad (73)$$

The Differential Algebra (DA) express quantities such as the state vector and the measurement vector in their Taylor series expansion up to a selected order. Therefore, the propagation function is applied directly on each polynomial (one polynomial for component) of the state. The same reasoning is valid for the measurement function.

$$x_i(k+1) = \mathbf{f}_i[\hat{\mathbf{x}}(k|k)] + \sum_{r=1}^c \frac{1}{r!} \sum \frac{\partial^r \mathbf{f}_i[\mathbf{x}(k|k)]}{\partial x_1^{\gamma_1} \dots \partial x_n^{\gamma_r}} \delta x_1^{\gamma_1}(k) \dots \delta x_n^{\gamma_r}(k) + \boldsymbol{\nu}_i(k) \quad (74)$$

$$z_p(k+1) = \mathbf{h}_p[\mathbf{f}[\hat{\mathbf{x}}(k+1|k)]] + \sum_{r=1}^c \frac{1}{r!} \sum \frac{\partial^r \mathbf{h}_p[\mathbf{x}(k+1)]}{\partial x_1^{\gamma_1} \dots \partial x_n^{\gamma_r}} \delta x_1^{\gamma_1}(k) \dots \delta x_n^{\gamma_r}(k) + \boldsymbol{\eta}_i(k) \quad (75)$$

Where  $c$  is the arbitrary order of the expansion and the second summation is over all permutations of  $\gamma_i \in \{1, \dots, n\}$  with  $i \in \{1, \dots, r\}$ . Hence, the summation of  $\mathbf{f}_i[\mathbf{x}(k|k)]$  includes the higher-order partials of the solution flow, which maps the deviations from time  $k$  to time  $k+1$  and, in an analogue way, the summation of  $\mathbf{h}_p[\mathbf{x}(k+1)]$  includes the higher-order partials of the measurement function. These two functions are both obtained by integration in the DA framework of the equations of motion and evaluating the measurement equation<sup>11,18</sup>.

## Prediction

Starting from the knowledge of the state estimate  $\hat{\mathbf{x}}(k)$  and its central moments up to a selected order  $c$ ,  $\mathbf{P}_{\mathbf{x}\mathbf{x}}^{[2]}$ ,  $\mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}}^{[3]}$ ,  $\mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x}}^{[4]}$ ,  $\dots$ ,  $\mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[c]}$ , the prediction part of the Kalman filter begins using Equation (74). It is now possible to predict the state mean:

$$\hat{\mathbf{x}}_i^- = \mathbb{E} \{ \mathbf{f}_i[\mathbf{x}(k)] + \boldsymbol{\nu}_i(k) \} = \quad (76)$$

$$= \mathbf{f}_i[\hat{\mathbf{x}}(k|k)] + \sum_{r=1}^c \frac{1}{r!} \sum \frac{\partial^r \mathbf{f}_i[\mathbf{x}(k|k)]}{\partial x_1^{\gamma_1} \dots \partial x_n^{\gamma_r}} \mathbb{E} \{ \delta x_1^{\gamma_1}(k) \dots \delta x_m^{\gamma_r}(k) \} \quad (77)$$

$$= \mathbf{f}_i[\hat{\mathbf{x}}(k|k)] + \sum_{\rho=1}^n \sum_{\sigma=1}^n \frac{1}{2!} \sum \frac{\partial^2 \mathbf{f}_i[\mathbf{x}(k|k)]}{\partial x_1^{\gamma_1} \dots \partial x_n^{\gamma_2}} \mathbf{P}_{\rho\sigma}^{[2]} + \sum_{\rho=1}^n \sum_{\sigma=1}^n \sum_{\tau=1}^n \frac{1}{3!} \sum \frac{\partial^3 \mathbf{f}_i[\mathbf{x}(k|k)]}{\partial x_1^{\gamma_1} \dots \partial x_n^{\gamma_3}} \mathbf{P}_{\rho\sigma\tau}^{[3]} + \quad (78)$$

$$+ \sum_{\rho=1}^n \sum_{\sigma=1}^n \sum_{\tau=1}^n \sum_{\nu=1}^n \frac{1}{4!} \sum \frac{\partial^4 \mathbf{f}_i[\mathbf{x}(k|k)]}{\partial x_1^{\gamma_1} \dots \partial x_n^{\gamma_4}} \mathbf{P}_{\rho\sigma\tau\nu}^{[4]} + \dots + \sum_{\rho=1}^n \dots \sum_{\omega=1}^n \frac{1}{c!} \sum \frac{\partial^c \mathbf{f}_i[\mathbf{x}(k|k)]}{\partial x_1^{\gamma_1} \dots \partial x_n^{\gamma_c}} \mathbf{P}_{\rho\dots\omega}^{[c]}$$

$$= \text{eval}(\mathbf{f}_i[\mathbf{x}(k)] + \boldsymbol{\nu}_i(k), \mathbf{P}_{\mathbf{x}\mathbf{x}}^{[2]}(k), \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}}^{[3]}(k), \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x}}^{[4]}(k), \dots, \mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[c]}(k)) \quad (79)$$

Where from now on, for the sake of brevity, the evaluation of the expected value through polynomial evaluation will be expressed using function  $\text{eval}()$ . In an analogous way, the predicted state covariance is evaluated considering the deviation of the propagated state polynomials with respect to the estimated mean.

$$\mathbf{P}_{\mathbf{x}\mathbf{x},ij}^{[2]-} = \mathbb{E} \{ [\mathbf{f}_i[\mathbf{x}(k)] - \hat{\mathbf{x}}_i^- + \boldsymbol{\nu}_i(k)] [\mathbf{f}_j[\mathbf{x}(k)] - \hat{\mathbf{x}}_j^- + \boldsymbol{\nu}_j(k)] \} \quad (80)$$

$$= \sum_{r=1}^c \sum_{s=1}^c \frac{1}{r!s!} \sum \frac{\partial^r \mathbf{f}_i[\mathbf{x}(k|k)]}{\partial x_1^{\gamma_1} \dots \partial x_n^{\gamma_r}} \sum \frac{\partial^s \mathbf{f}_j[\mathbf{x}(k|k)]}{\partial x_1^{\gamma_1} \dots \partial x_n^{\gamma_s}} \cdot \mathbb{E} \left\{ \delta x_1^{\gamma_1}(k) \dots \delta x_m^{\gamma_r}(k) \delta x_1^{\xi_1}(k) \dots \delta x_m^{\xi_s}(k) \right\} - \delta \hat{x}_i^- \delta \hat{x}_j^- + \mathbf{Q}_{ij}^{[2]}(k) \quad (81)$$

Where  $\xi_i \in \{1, \dots, m\}$  and  $\delta \hat{x}_i^- = \mathbf{f}_i[\hat{\mathbf{x}}(k|k)] - \hat{\mathbf{x}}_i^-$ . Defining the state deviation vector such as

$$d\mathbf{x}_i^* = \mathbf{f}_i[\mathbf{x}(k)] - \hat{\mathbf{x}}_i^- + \boldsymbol{\nu}_i(k) \quad (82)$$

equation (81) can be then written using the  $\text{eval}()$  function

$$\mathbf{P}_{\mathbf{x}\mathbf{x},ij}^{[2]*} = \text{eval}(d\mathbf{x}_i^*, d\mathbf{x}_j^*, \mathbf{P}_{\mathbf{x}\mathbf{x}}^{[2]}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}}^{[3]}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x}}^{[4]}, \dots, \mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[2c]}) \quad (83)$$

where it must be noted that the maximum moment requested by the  $\text{eval}()$  function is of order  $2c$ . The evaluation of the covariance require a multiplication between two  $c$ -th order polynomials, therefore the expansion order of the series is doubled. The pursuit of the filter is to estimate correctly up to the second moment, thus the precise evaluation of moments up to order  $2c$  must be implemented. The prediction of those moments performed thanks to the  $\text{eval}()$  function;

$$\mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x},ijk}^{[3]*} = \text{eval}(d\mathbf{x}_i^*, d\mathbf{x}_j^*, d\mathbf{x}_k^*, \mathbf{P}_{\mathbf{x}\mathbf{x}}^{[2]}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}}^{[3]}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x}}^{[4]}, \dots, \mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[3c]}) \quad (84)$$

$$\mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x},ijkl}^{[4]*} = \text{eval}(d\mathbf{x}_i^*, d\mathbf{x}_j^*, d\mathbf{x}_k^*, d\mathbf{x}_l^*, \mathbf{P}_{\mathbf{x}\mathbf{x}}^{[2]}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}}^{[3]}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x}}^{[4]}, \dots, \mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[4c]}) \quad (85)$$

$\dots = \dots$

$$\mathbf{P}_{\mathbf{x}\dots\mathbf{x},i\dots q}^{[2c]*} = \text{eval}(d\mathbf{x}_i^*, \dots, d\mathbf{x}_q^*, \mathbf{P}_{\mathbf{x}\mathbf{x}}^{[2]}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}}^{[3]}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x}}^{[4]}, \dots, \mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[2c^2]}) \quad (86)$$

where the order of the required inputs moments increases quadratically since, each time, the polynomial  $d\mathbf{x}$  multiplies by itself, increasing the order of a factor  $c$ .

A moment approximation has been used in order to stop this iteration process that, otherwise, would require to evaluate moments up to infinite order. Thanks to the monotonicity of the moments,<sup>19</sup> it is possible

to approximate their values using the following inequalities. The Lyapounov's inequality states that, if  $0 < r \leq s$  then

$$\mathbb{E}\{|\mathbf{X}|^r\}^{1/r} \leq \mathbb{E}\{|\mathbf{X}|^s\}^{1/s} \quad (87)$$

while Hölder inequality (used for multi-variable) states that if  $r > 1$  and  $\frac{1}{r} + \frac{1}{s} = 1$ , then

$$\mathbb{E}\{|\mathbf{XY}|\} \leq \mathbb{E}\{|\mathbf{X}|^r\}^{1/r} \mathbb{E}\{|\mathbf{Y}|^s\}^{1/s} \quad (88)$$

Therefore, moments from order  $(2c + 1)$ -th to order  $2c^2$ -th are evaluated using the lower-bound of these inequalities, as approximation, from the knowledge of the predicted moments up to order  $2c$ -th. It is also important to underline that, as long as working with the linear deviation  $d\mathbf{x}$ , the identities  $\mathbf{P}_{d\mathbf{x}d\mathbf{x}}^{[2]*} = \mathbf{P}_{\mathbf{x}\mathbf{x}}^{[2]*}$ ,  $\mathbf{P}_{d\mathbf{x}d\mathbf{x}d\mathbf{x}}^{[3]*} = \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}}^{[3]*}$  and  $\mathbf{P}_{d\mathbf{x}d\mathbf{x}d\mathbf{x}d\mathbf{x}}^{[4]*} = \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x}}^{[4]*}$  hold.

Adding the influence of the process noise:

$$\mathbf{P}_{\mathbf{x}\mathbf{x},ij}^{[2]-} = \mathbf{P}_{\mathbf{x}\mathbf{x},ij}^{[2]*} + \mathbf{Q}_{ij}^{[2]} \quad (89)$$

$$\mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x},ijk}^{[3]-} = \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x},ijk}^{[3]*} + \mathbf{Q}_{ijk}^{[3]} \quad (90)$$

$$\begin{aligned} \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x},ijkl}^{[4]-} &= \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x},ijkl}^{[4]*} + \mathbf{P}_{\mathbf{x}\mathbf{x},ij}^{[2]*} \mathbf{Q}_{kl}^{[2]} + \mathbf{P}_{\mathbf{x}\mathbf{x},ik}^{[2]*} \mathbf{Q}_{jl}^{[2]} + \mathbf{P}_{\mathbf{x}\mathbf{x},il}^{[2]*} \mathbf{Q}_{kj}^{[2]} + \\ &+ \mathbf{P}_{\mathbf{x}\mathbf{x},jk}^{[2]*} \mathbf{Q}_{il}^{[2]} + \mathbf{P}_{\mathbf{x}\mathbf{x},jl}^{[2]*} \mathbf{Q}_{ik}^{[2]} + \mathbf{P}_{\mathbf{x}\mathbf{x},kl}^{[2]*} \mathbf{Q}_{ij}^{[2]} + \mathbf{Q}_{ijkl}^{[4]} \end{aligned} \quad (91)$$

... = ...

$$\mathbf{P}_{\mathbf{x}\dots\mathbf{x},i\dots q}^{[2c^2]-} = \mathbf{P}_{\mathbf{x}\dots\mathbf{x},i\dots q}^{[2c^2]*} + \dots \quad (92)$$

The  $c$ -th moment of the process noise is evaluated with the integral definition

$$\mathbf{Q}_{ij\dots q}^{[c]} = \int \mathbf{w}_i \mathbf{w}_j \dots \mathbf{w}_q p(\mathbf{w}_i \mathbf{w}_j \dots \mathbf{w}_q) d\mathbf{x} \quad (93)$$

where  $p(\cdot)$  is the probability density function of the random process. The influence of the noise on high order moments can be easily calculated using combinatory theory. The  $c$ -th order state moment has a number of noise terms equal to the  $c$ -th row of the Pascal triangle, where each number on the row is found using the binomial coefficient. A quick example shows the process: considering the 6th order, the 6th row of the Pascal triangle is

$$\binom{6}{i} \quad i = 0, \dots, 6 \quad \longrightarrow \quad 1 \ 6 \ 15 \ 20 \ 15 \ 6 \ 1$$

The second column in Table 1 is the 6th row of the Pascal's triangle and it represents the number of terms with state moment of order  $i$ . Moreover, looking at Table 1, the rows with  $i = 1$  and  $i = 5$  have null contribution because  $\mathbb{E}\{d\mathbf{x}_i\} = \mathbb{E}\{\mathbf{w}_i\} = 0$ . Generalizing this example to an arbitrary order  $c$ , it reduces the problem of the influence of the process noise as listing all the different combinations between  $\mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[i]*}$  and  $\mathbf{Q}^{[j]}$ .

The state of the system has been estimated in all its moments up to the selected order. It is now possible to initialize a new polynomial in the DA framework, centering the Taylor expansion series around the predicted mean  $\hat{\mathbf{x}}^-$ . Thanks to this shifting, the predicted moments  $\mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[i]-}$  are the new central moments of the state. Therefore, from this point on, the evaluation of any expected value through the  $eval(\cdot)$  function will require  $\mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[i]-}$  as inputs instead of  $\mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[i]*}$ .

$$\mathbf{x} = \hat{\mathbf{x}}^- + d\mathbf{x}^- \quad \mathbf{P}_{\mathbf{x}\mathbf{x}}^{[2]-}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}}^{[3]-}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x}}^{[4]-}, \dots \quad (94)$$

The prediction step continues by evaluating the predicted measurement mean and its covariance.

$$\hat{\mathbf{y}}_i = eval(\mathbf{h}_i[\mathbf{x}(k+1)] + \boldsymbol{\eta}_i(k+1), \mathbf{P}_{\mathbf{x}\mathbf{x}}^{[2]-}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}}^{[3]-}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x}}^{[4]-}, \dots, \mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[c]-}) \quad (95)$$

$$d\mathbf{y}_i = \mathbf{h}_i[\mathbf{x}(k+1)] - \hat{\mathbf{y}}_i + \boldsymbol{\eta}_i(k+1) \quad (96)$$

$i$	# of terms	terms
0	1	$P_{\mathbf{xxxxxx},ijklpq}^{[6]*} = eval(dx_i^*, dx_j^*, dx_k^*, dx_l^*, dx_p^*, dx_q^*, P_{\mathbf{xx}}^{[2]}, P_{\mathbf{xxx}}^{[3]}, P_{\mathbf{xxxx}}^{[4]}, P_{\mathbf{xxxxx}}^{[5]}, P_{\mathbf{xxxxxx}}^{[6]})$
1	6	0
2	15	$P_{\mathbf{xx},ij}^{[2]*} Q_{klpq}^{[4]} + P_{\mathbf{xx},ik}^{[2]*} Q_{jlpq}^{[4]} + P_{\mathbf{xx},il}^{[2]*} Q_{kjpp}^{[4]} + P_{\mathbf{xx},ip}^{[2]*} Q_{kljq}^{[4]} + P_{\mathbf{xx},iq}^{[2]*} Q_{klpj}^{[4]} + P_{\mathbf{xx},jk}^{[2]*} Q_{ilpq}^{[4]} + P_{\mathbf{xx},jl}^{[2]*} Q_{kipq}^{[4]} + P_{\mathbf{xx},jp}^{[2]*} Q_{iklq}^{[4]} + P_{\mathbf{xx},jq}^{[2]*} Q_{iklp}^{[4]} + P_{\mathbf{xx},kl}^{[2]*} Q_{ijpq}^{[4]} + P_{\mathbf{xx},kp}^{[2]*} Q_{ijlq}^{[4]} + P_{\mathbf{xx},kq}^{[2]*} Q_{ijlp}^{[4]} + P_{\mathbf{xx},lp}^{[2]*} Q_{ijkq}^{[4]} + P_{\mathbf{xx},lq}^{[2]*} Q_{ijkp}^{[4]} + P_{\mathbf{xx},pq}^{[2]*} Q_{ijkl}^{[4]}$
3	20	$P_{\mathbf{xxx},ijk}^{[3]*} Q_{lpq}^{[3]} + P_{\mathbf{xxx},ijl}^{[3]*} Q_{kpq}^{[3]} + P_{\mathbf{xxx},ijp}^{[3]*} Q_{klq}^{[3]} + P_{\mathbf{xxx},ijq}^{[3]*} Q_{klp}^{[3]} + P_{\mathbf{xxx},ikl}^{[3]*} Q_{jpp}^{[3]} + P_{\mathbf{xxx},ikp}^{[3]*} Q_{jlp}^{[3]} + P_{\mathbf{xxx},ikq}^{[3]*} Q_{jlp}^{[3]} + P_{\mathbf{xxx},ilp}^{[3]*} Q_{jkq}^{[3]} + P_{\mathbf{xxx},ilq}^{[3]*} Q_{jkp}^{[3]} + P_{\mathbf{xxx},ipq}^{[3]*} Q_{jkl}^{[3]} + P_{\mathbf{xxx},jkl}^{[3]*} Q_{ipq}^{[3]} + P_{\mathbf{xxx},jqp}^{[3]*} Q_{ilk}^{[3]} + P_{\mathbf{xxx},jkq}^{[3]*} Q_{ilp}^{[3]} + P_{\mathbf{xxx},jlp}^{[3]*} Q_{ikq}^{[3]} + P_{\mathbf{xxx},jlp}^{[3]*} Q_{ikp}^{[3]} + P_{\mathbf{xxx},jlp}^{[3]*} Q_{ikl}^{[3]} + P_{\mathbf{xxx},klp}^{[3]*} Q_{ijq}^{[3]} + P_{\mathbf{xxx},hlq}^{[3]*} Q_{ijp}^{[3]} + P_{\mathbf{xxx},kpp}^{[3]*} Q_{ijl}^{[3]} + P_{\mathbf{xxx},lpp}^{[3]*} Q_{ijk}^{[3]}$
4	15	$P_{\mathbf{xxxx},klpq}^{[4]*} Q_{ij}^{[2]} + P_{\mathbf{xxxx},jlpq}^{[4]*} Q_{ik}^{[2]} + P_{\mathbf{xxxx},kjpp}^{[4]*} Q_{il}^{[2]} + P_{\mathbf{xxxx},kljq}^{[4]*} Q_{ip}^{[2]} + P_{\mathbf{xxxx},klpj}^{[4]*} Q_{iq}^{[2]} + P_{\mathbf{xxxx},ilpq}^{[4]*} Q_{jk}^{[2]} + P_{\mathbf{xxxx},kipq}^{[4]*} Q_{jl}^{[2]} + P_{\mathbf{xxxx},iklq}^{[4]*} Q_{jp}^{[2]} + P_{\mathbf{xxxx},iklp}^{[4]*} Q_{jq}^{[2]} + P_{\mathbf{xxxx},ijpp}^{[4]*} Q_{kl}^{[2]} + P_{\mathbf{xxxx},ijlq}^{[4]*} Q_{kp}^{[2]} + P_{\mathbf{xxxx},ijlp}^{[4]*} Q_{kq}^{[2]} + P_{\mathbf{xxxx},ijkq}^{[4]*} Q_{lp}^{[2]} + P_{\mathbf{xxxx},ijkp}^{[4]*} Q_{lq}^{[2]} + P_{\mathbf{xxxx},ijkl}^{[4]*} Q_{pq}^{[2]}$
5	6	0
6	1	$Q_{ijklpq}^{[6]}$

Table 1. Terms divided by moment noise order of  $P_{\mathbf{xxxxxx},ijklpq}^{[6]-} (k+1|k)$

As previously seen, the quadratic update requires to augment the measurements with their second power, using the Kronecker product. However, working with deviations only, it is possible to avoid the computation of the predicted mean of the measurements square: there is no need to evaluate the augmented measurement vector since the quadratic deviation vector is calculated in the following way.

$$d\mathbf{y}^{[2]} = d\mathbf{y} \otimes d\mathbf{y} \quad (97)$$

Therefore, the covariance of the augmented measurements is evaluated in its components.

$$\mathbf{P}_{d\mathbf{y}d\mathbf{y},ij}^{[2]*} = eval(d\mathbf{y}_i, d\mathbf{y}_j, \mathbf{P}_{\mathbf{xx}}^{[2]-}, \mathbf{P}_{\mathbf{xxx}}^{[3]-}, \mathbf{P}_{\mathbf{xxxx}}^{[4]-}, \dots, \mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[2c]-}) \quad (98)$$

$$\mathbf{P}_{d\mathbf{y}d\mathbf{y}^{[2]},i(jk)}^{[2]*} = eval(d\mathbf{y}_i, d\mathbf{y}_{jk}^{[2]}, \mathbf{P}_{\mathbf{xx}}^{[2]-}, \mathbf{P}_{\mathbf{xxx}}^{[3]-}, \mathbf{P}_{\mathbf{xxxx}}^{[4]-}, \dots, \mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[3c]-}) \quad (99)$$

$$\mathbf{P}_{d\mathbf{y}^{[2]}d\mathbf{y},(ij)k}^{[2]*} = \mathbf{P}_{\mathbf{yy}^{[2]},k(ij)}^{[2]*T} \quad (100)$$

$$\mathbf{P}_{d\mathbf{y}^{[2]}d\mathbf{y}^{[2]},(ij)(kl)}^{[2]*} = eval(d\mathbf{y}_{ij}^{[2]}, d\mathbf{y}_{kl}^{[2]}, \mathbf{P}_{\mathbf{xx}}^{[2]-}, \mathbf{P}_{\mathbf{xxx}}^{[3]-}, \mathbf{P}_{\mathbf{xxxx}}^{[4]-}, \dots, \mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[4c]-}) \quad (101)$$

Indeed, working with deviation has the advantage to zero all the terms with the presence of the predicted mean: thus  $\mathbf{P}_{d\mathbf{y}d\mathbf{y}d\mathbf{y}}^{[3]*} = \mathbf{P}_{d\mathbf{y}d\mathbf{y}^{[2]}}^{[2]*}$ . The influence of the noise is then added having in mind that working around the mean resets mostly of the cross terms.

$$\mathbf{P}_{d\mathbf{y}d\mathbf{y},ij}^{[2]-} = \mathbf{P}_{d\mathbf{y}d\mathbf{y},ij}^{[2]*} + \mathbf{R}_{ij}^{[2]} \quad (102)$$

$$\mathbf{P}_{d\mathbf{y}d\mathbf{y}^{[2]},i(jk)}^{[2]-} = \mathbf{P}_{d\mathbf{y}d\mathbf{y}^{[2]},i(jk)}^{[2]*} + \mathbf{R}_{ijk}^{[3]} \quad (103)$$

$$\mathbf{P}_{d\mathbf{y}^{[2]}d\mathbf{y},(ij)k}^{[2]-} = \mathbf{P}_{d\mathbf{y}d\mathbf{y}^{[2]},k(ij)}^{[2]-T} \quad (104)$$

$$\begin{aligned} \mathbf{P}_{d\mathbf{y}^{[2]}d\mathbf{y}^{[2]},(ij)(kl)}^{[2]-} &= \mathbf{P}_{d\mathbf{y}^{[2]}d\mathbf{y}^{[2]},(ij)(kl)}^{[2]*} + \mathbf{P}_{d\mathbf{y}d\mathbf{y},ik}^{[2]*} \mathbf{R}_{jl}^{[2]} + \mathbf{P}_{d\mathbf{y}d\mathbf{y},il}^{[2]*} \mathbf{R}_{jk}^{[2]} + \\ &+ \mathbf{P}_{d\mathbf{y}d\mathbf{y},jk}^{[2]*} \mathbf{R}_{il}^{[2]} + \mathbf{P}_{d\mathbf{y}d\mathbf{y},ij}^{[2]*} \mathbf{R}_{kl}^{[2]} + \mathbf{P}_{d\mathbf{y}d\mathbf{y},kl}^{[2]*} \mathbf{R}_{ij}^{[2]} + \mathbf{P}_{d\mathbf{y}d\mathbf{y},jl}^{[2]*} \mathbf{R}_{ik}^{[2]} + \mathbf{R}_{ijkl}^{[4]} \end{aligned} \quad (105)$$

$$\mathbf{P}_{d\mathbf{y}^{[2]}d\mathbf{y}^{[2]}}^{[2]-} = \mathbf{P}_{d\mathbf{y}^{[2]}d\mathbf{y}^{[2]}}^{[2]*} - \text{vect} \left( \mathbf{P}_{d\mathbf{y}d\mathbf{y}}^{[2]-} \right) \otimes \text{vect} \left( \mathbf{P}_{d\mathbf{y}d\mathbf{y}}^{[2]-} \right)^T \quad (106)$$

$$\mathbf{P}_{\mathbf{Y}\mathbf{Y}} = \begin{bmatrix} \mathbf{P}_{d\mathbf{y}d\mathbf{y}}^{[2]-} & \mathbf{P}_{d\mathbf{y}d\mathbf{y}^{[2]}}^{[2]-} \\ \mathbf{P}_{d\mathbf{y}^{[2]}d\mathbf{y}}^{[2]-} & \mathbf{P}_{d\mathbf{y}^{[2]}d\mathbf{y}^{[2]}}^{[2]-} \end{bmatrix} \quad (107)$$

Where all the redundant components have been deleted. Moreover, Equation (106) is a necessity since it allows the passage from  $\mathbf{P}_{d\mathbf{y}d\mathbf{y}d\mathbf{y}d\mathbf{y}}^{[4]-}$  to  $\mathbf{P}_{d\mathbf{y}^{[2]}d\mathbf{y}^{[2]}}^{[2]-}$ .

The prediction step of the Kalman filter is complete after the evaluation of the state-measurement cross-covariance matrix.

$$\mathbf{P}_{\mathbf{x}d\mathbf{y},ij}^{[2]-} = eval(d\mathbf{x}_i^-, d\mathbf{y}_j, \mathbf{P}_{\mathbf{xx}}^{[2]-}, \mathbf{P}_{\mathbf{xxx}}^{[3]-}, \mathbf{P}_{\mathbf{xxxx}}^{[4]-}, \dots, \mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[2c]-}) \quad (108)$$

$$\mathbf{P}_{\mathbf{x}d\mathbf{y}^{[2]},i(jk)}^{[2]-} = eval(d\mathbf{x}_i^-, d\mathbf{y}_{jk}^{[2]}, \mathbf{P}_{\mathbf{xx}}^{[2]-}, \mathbf{P}_{\mathbf{xxx}}^{[3]-}, \mathbf{P}_{\mathbf{xxxx}}^{[4]-}, \dots, \mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[2c]-}) \quad (109)$$

$$\mathbf{P}_{\mathbf{x}\mathbf{Y}} = \begin{bmatrix} \mathbf{P}_{\mathbf{x}d\mathbf{y}}^{[2]-} & \mathbf{P}_{\mathbf{x}d\mathbf{y}^{[2]}}^{[2]-} \end{bmatrix} \quad (110)$$

Where, again,  $\mathbf{P}_{\mathbf{x}d\mathbf{y}}^{[2]-} = \mathbf{P}_{d\mathbf{x}d\mathbf{y}}^{[2]-}$  and  $\mathbf{P}_{\mathbf{x}d\mathbf{y}^{[2]}}^{[2]-} = \mathbf{P}_{d\mathbf{x}d\mathbf{y}^{[2]}}^{[2]-}$ .

## Update

The solution of system is the Kalman Gain using the augmented measurement  $\mathbf{Y}$ .

$$\mathbf{K} = \mathbf{P}_{\mathbf{x}\mathbf{Y}} \mathbf{P}_{\mathbf{Y}\mathbf{Y}}^{-1} \quad (111)$$

which, after the evaluation of the residual, leads to the updated estimation error and the optimal state estimate.

$$\mathbf{d} = \tilde{\mathbf{y}} - \hat{\mathbf{y}} \quad (112)$$

$$\mathbf{d}^{[2]} = \mathbf{d} \otimes \mathbf{d} - \text{vect}(\mathbf{P}_{d\mathbf{y}d\mathbf{y}}^{[2]*}) \quad (113)$$

$$\mathbf{n} = \begin{bmatrix} \mathbf{d} & \mathbf{d}^{[2]} \end{bmatrix}^T \quad (114)$$

$$d\mathbf{Y} = \begin{bmatrix} d\mathbf{y} & d\mathbf{y}^{[2]} - \text{vect}(\mathbf{P}_{d\mathbf{y}d\mathbf{y}}^{[2]-}) \end{bmatrix}^T \quad (115)$$

$$d\mathbf{x}^+ = d\mathbf{x}^- - \mathbf{K}d\mathbf{Y} \quad (116)$$

$$\hat{\mathbf{x}}^+ = \hat{\mathbf{x}}^- + \mathbf{K}\mathbf{n} \quad (117)$$

where  $\tilde{\mathbf{y}}$  is the actual numerical value of the measurement (the outcome of random vector  $\mathbf{y}$  read from sensors) and the measurement covariance consider the presence of the noise.

The predicted moments are corrected to their updated version by applying function  $eval()$  with the new state deviation vector.

$$\mathbf{P}_{\mathbf{x}\mathbf{x},ij}^{[2]+} = eval(d\mathbf{x}_i^+, d\mathbf{x}_j^+, \mathbf{P}_{\mathbf{x}\mathbf{x}}^{[2]-}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}}^{[3]-}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x}}^{[4]-}, \dots, \mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[2c]-}) + \mathbf{K}\mathbf{R}_{aug}\mathbf{K}^T \quad (118)$$

$$\mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x},ijk}^{[3]+} = eval(d\mathbf{x}_i^+, d\mathbf{x}_j^+, d\mathbf{x}_k^+, \mathbf{P}_{\mathbf{x}\mathbf{x}}^{[2]-}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}}^{[3]-}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x}}^{[4]-}, \dots, \mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[3c]-}) \quad (119)$$

$$\mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x},ijkl}^{[4]+} = eval(d\mathbf{x}_i^+, d\mathbf{x}_j^+, d\mathbf{x}_k^+, d\mathbf{x}_l^+, \mathbf{P}_{\mathbf{x}\mathbf{x}}^{[2]-}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}}^{[3]-}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x}}^{[4]-}, \dots, \mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[4c]-}) \quad (120)$$

$\dots = \dots$

$$\mathbf{P}_{\mathbf{x}\dots\mathbf{x},i\dots q}^{[2c]+} = eval(d\mathbf{x}_i^+, \dots, d\mathbf{x}_q^+, \mathbf{P}_{\mathbf{x}\mathbf{x}}^{[2]-}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}}^{[3]-}, \mathbf{P}_{\mathbf{x}\mathbf{x}\mathbf{x}\mathbf{x}}^{[4]-}, \dots, \mathbf{P}_{\mathbf{x}\dots\mathbf{x}}^{[2c^2]-}) \quad (121)$$

where

$$\mathbf{R}_{aug} = \begin{bmatrix} \mathbf{R}^{[2]} & \mathbf{R}^{[3]T} \\ \mathbf{R}^{[3]} & \mathbf{R}^{[4]} \end{bmatrix} \quad (122)$$

and considering the influence of the noise on the higher order moments using combinatorial as in the prediction step.

And, as last step, approximate the remaining updated moments up to order  $2c^2$ -th using Lyapounov's and Hölder's inequalities approximation.

### NON-GAUSSIAN NOISE EXAMPLE

The reliability of the filter has been assessed, at first, in a linear problem with non-Gaussian process and measurement noise. The chosen example resemble the one from De Santis and Germani,<sup>15</sup> with a simple linear discrete-time scalar system:

$$x_{k+1} = ax_k + f_k \quad \bar{x} = 0 \quad (123)$$

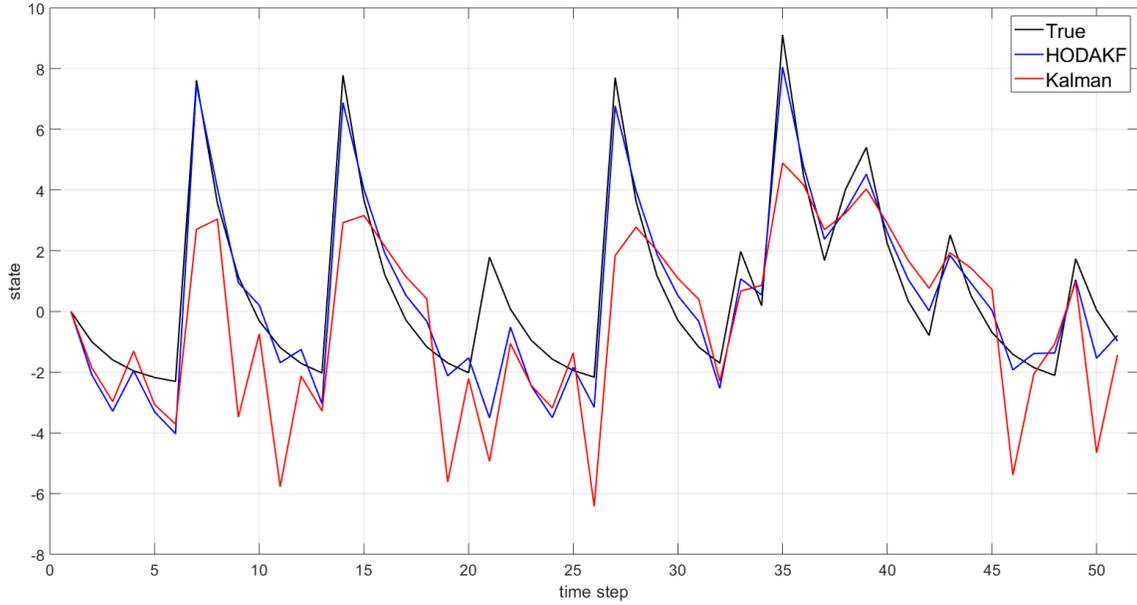
$$y_k = cx_k + g_k \quad (124)$$

with  $a = 0.6$ ,  $c = 0.8$ ,  $f_k$  and  $g_k$  independent zero-mean random sequences with distribution described in Table 2. The HODAKF has been compared with the classical Kalman filter in the example, both with perfect

$f_k$		-1		3		9		$g_k$		1		-3		-9
$P(f_k)$		15/18		2/18		1/18		$P(g_k)$		15/18		2/18		1/18

**Table 2. Random sequences distribution**

initial condition knowledge (null initial condition and null initial moments). A single run for the example is displayed in Fig. 1, with  $N = 50$  steps. It can be seen how the high order filter better follows the true



**Figure 1. HODAKF vs Kalman Filter**

value of the system under the influence of noise, while the Kalman filter struggles to match the true behavior especially when a high noise pick occurs. The HODAKF, indeed, has a much smaller error distribution, as proven by comparing the covariance values (HODAKF run with order  $n = 2$ ).

$$P_{HODAKF} = 1.2068 \quad P_{Kalman} = 2.0972 \quad (125)$$

and it is shown in Fig. 2 where it can be seen, through a Monte Carlo analysis, how the two filters predicted covariances match the effective one from the samples. Indeed, the covariance of the HODAKF is much smaller with respect to the one predicted by the classical Kalman filter. In both cases, the error has zero mean (unbiased filters).

### ORBIT ESTIMATION EXAMPLE

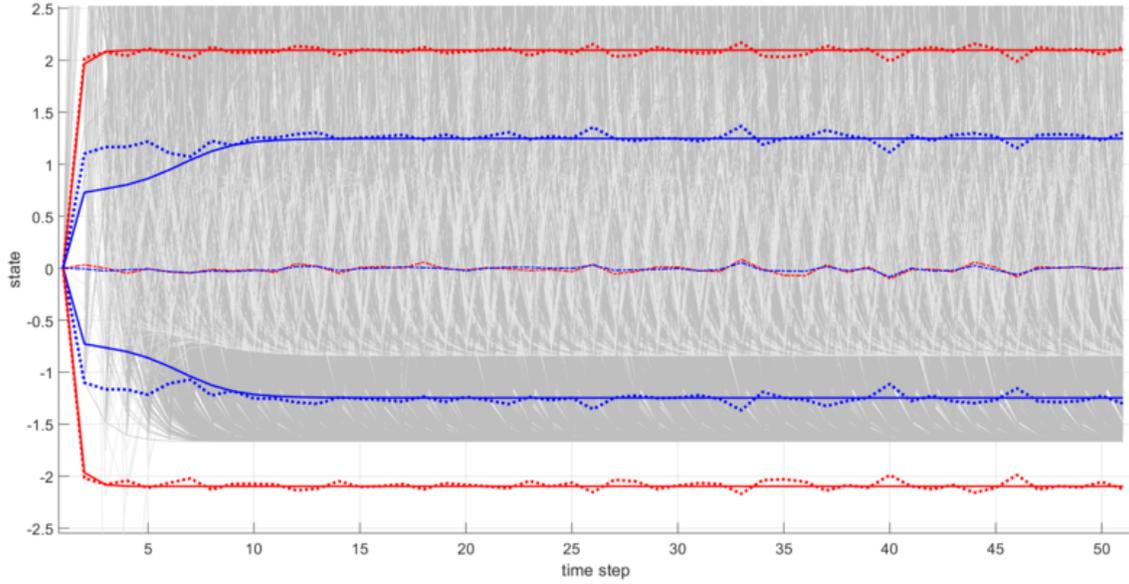
The performance of the filter has been assessed with a nonlinear problem common in celestial mechanics: the two body problem. The equations of motion governing the system are the ones associated to the Keplerian dynamics, where  $\mathbf{r}$  is the position vector of the spacecraft and  $\mu$  is the Earth gravitational parameter.

$$\frac{d\dot{\mathbf{r}}}{dt} = -\frac{\mu}{r^3}\mathbf{r} \quad (126)$$

The initial condition and uncertainties values resemble the one in<sup>12,18</sup>. Therefore, each length unit has been scaled by the orbit semi-major axis,  $a = 8788 \text{ km}$ , and by the time  $\sqrt{\frac{a^3}{\mu}}$ .

$$\mathbf{x}_0 = \begin{pmatrix} \mathbf{r}_0 \\ \mathbf{v}_0 \end{pmatrix} = \begin{pmatrix} -0.68787 \\ -0.39713 \\ 0.28448 \\ -0.51331 \\ 0.98266 \\ 0.37611 \end{pmatrix} \quad (127)$$

and the initial estimate of the system state has a 10% offset from the true initial state.



**Figure 2. HODAKF (blue) vs Kalman Filter (red), Monte Carlo close up.  $n = 2$**

The measurement model assumes the radial position of the spacecraft w.r.t. the Earth and the line of sight direction of the planet:

$$y_1 = r + \eta_1 \quad (128)$$

$$y_2 = \arctan\left(\frac{x_2}{x_1}\right) + \eta_2 \quad (129)$$

$$y_3 = \arcsin\left(\frac{x_3}{r}\right) + \eta_3 \quad (130)$$

where  $\eta_i$ , with  $i = 1, 2, 3$ , is the measurement noise, assumed to be Gaussian. The standard deviation of the error is assumed to be  $10^{-3}$  for the radial position and  $1.745 \cdot 10^{-6}$  for the angle error. The simulation provides a Gaussian process noise with zero mean and standard deviation  $10^{-5}$ .

As already stated, this example is characterized by the presence of Gaussian noise. It is then required to state the first  $2c$  noise moments, where  $c$  is the selected filter order.

$$\mathbf{R}_{ij}^{[2]} = \begin{cases} \sigma_{\nu,ij}^2 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (131)$$

$$\mathbf{R}_{ijk}^{[3]} = 0 \quad \forall i, j, k \quad (132)$$

$$\mathbf{R}_{ijkl}^{[4]} = \begin{cases} 3\sigma_{\nu,ijkl}^4 & \text{if } i = j = k \\ \sigma_{\nu,ij}^2 \sigma_{\nu,kl}^2 & \text{if } i = j \wedge k = l \\ 0 & \text{otherwise} \end{cases} \quad (133)$$

and so on.

The initial error covariance matrix has been assumed as diagonal, the value of the variance for the position vector components is 0.01, while the variance for the velocity vector components is  $10^{-4}$ . Thus,

$$\sigma_r = 10^{-2} a \quad (134)$$

$$\sigma_v = 10^{-4} \sqrt{\frac{\mu}{a}} \quad (135)$$

The state high order moments are evaluated in the same way expressed by equation (133), by associating  $\sigma_r$  to indexes 1, 2, 3 and  $\sigma_v$  to indexes 4, 5, 6. However, the DACE<sup>18,20,21</sup> (Differential Algebra Core Engine) limits matrices dimensions up to 2; therefore any high order moment has been expressed using their vectorization representation. The  $c$ -th order moment of the state should be expressed with a  $c$ -dimensions matrix; however, thanks to the technique shown in Table 3, it is possible to use only two dimensions. The table shows that,

term	row position	column position	order $c$
$P_{ijklp}^{[5]}$	$in^2 + kn + l$	$jn + p$	odd
$P_{ijklpq}^{[6]}$	$jn^2 + ln + p$	$in^2 + kn + q$	even
$P_{ijth\dots pq}^{[c]}$	$in^{(c-1)/2} + tn^{(c-1)/2-1} \dots + p$	$jn^{(c-1)/2-1} + hn^{(c-1)/2-2} + \dots + q$	odd
$P_{ijth\dots pq}^{[c]}$	$jn^{c/2} + hn^{c/2-1} + \dots n + p$	$in^{c/2} + tn^{c/2-1} \dots + q$	even

**Table 3. High order moments representation**

except for the first two indexes that always fill the row first and then the column, each additional dimension creates a column vector of the matrices if it is odd and a matrix of column vectors if it is even.

However, due to the high computational demand in the moments evaluation of the filter for a state of 6 variables (a  $2^{nd}$  order filter asks for the 8-th moment which is a 1296x1296 matrix), it has been decided to evaluate moments using the Isserlis' formula. Future work is focused on how to achieve faster computation and match the estimation for systems with a bigger state vector. The filter follows the same algorithm described above, the only difference is that now the  $eval()$  function uses the Isserlis' formula on the monomials of the Taylor polynomials.<sup>22,23</sup> Therefore, the covariance update equation can then be written in the canonical form

$$P_{xx}^{[2]+} = P_{xx}^{[2]-} - KP_{YY}K^T \quad (136)$$

leading to a faster evaluation, with  $K$  and  $P_{YY}$  matched for a non-linear (quadratic) update, as described above.

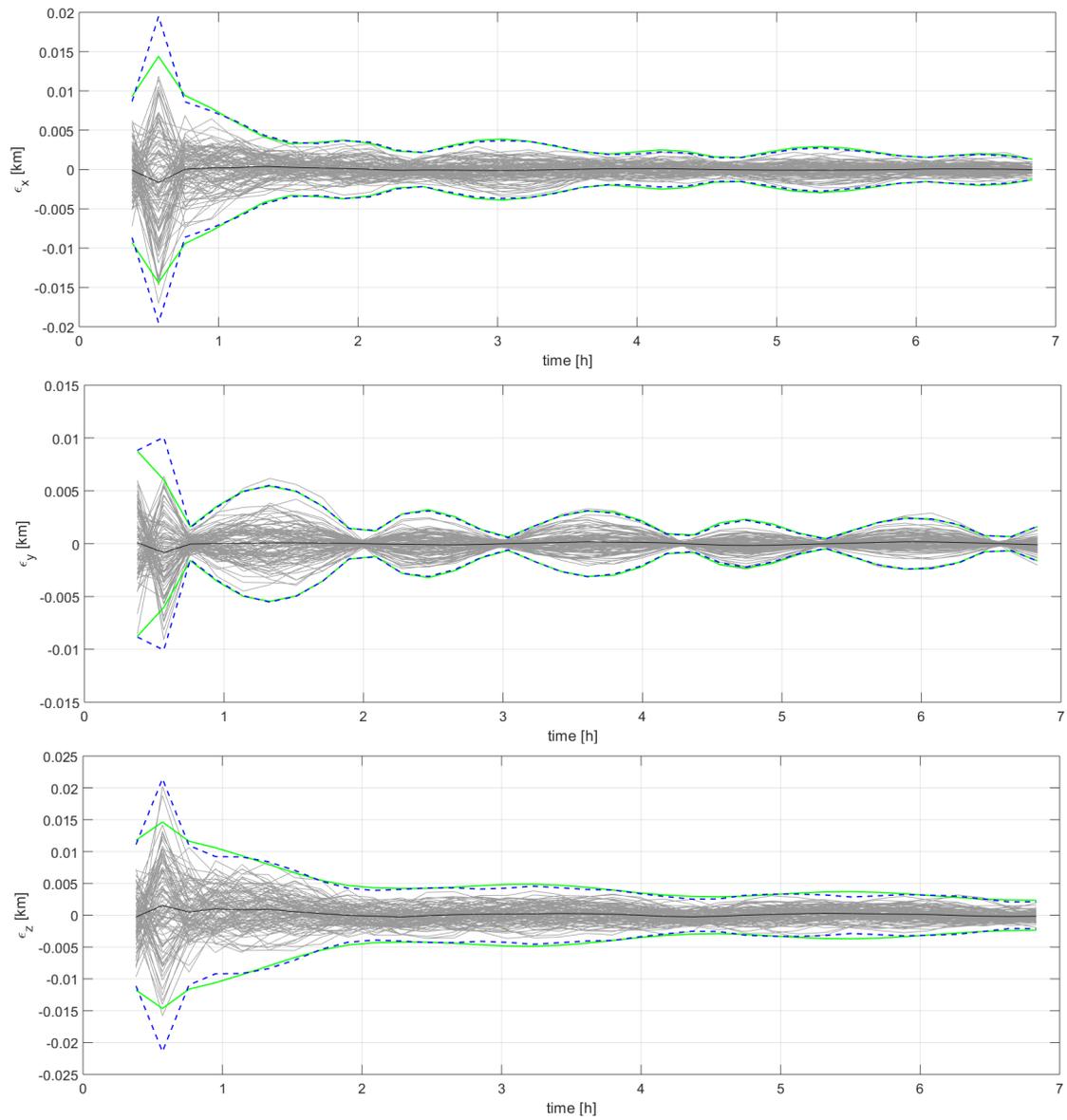
Fig. 3 shows the position error of the spacecraft in the three components while Fig. 4 shows the corresponding velocity. Order  $n = 2$  has been selected for the simulations. The error has been evaluated as the difference between the estimated position and velocity with the corresponding true values. The two figures represents the time duration of three orbits, with a total of 12 observations each orbit separated by the same time interval. The first steps have been left out in order to show the convergence of the error and the correct estimation of the covariance, shown with two lines in each graph as  $\pm 3\sigma$ . A Monte Carlo analysis with 100 samples has been performed and it can be noted how the filter covariance (green) is consistent with the one from the Monte Carlo samples (blue). Steady state is fast achieved and the figures shows the classical error behavior related to the orbital estimation in two body dynamics.

The benefits of the nonlinear update can be appreciated by studying the filter predicted covariance in the first steps. Fig. 5 represents the standard deviation profiles both for the spacecraft position and velocity. These quantities are calculated from the diagonal terms of the estimate position and velocity covariance matrices  $\sigma_{rx}^2, \sigma_{ry}^2, \sigma_{rz}^2, \sigma_{vx}^2, \sigma_{vy}^2, \sigma_{vz}^2$ .

$$\sigma_r = \sqrt{\sigma_{rx}^2 + \sigma_{ry}^2 + \sigma_{rz}^2} \quad (137)$$

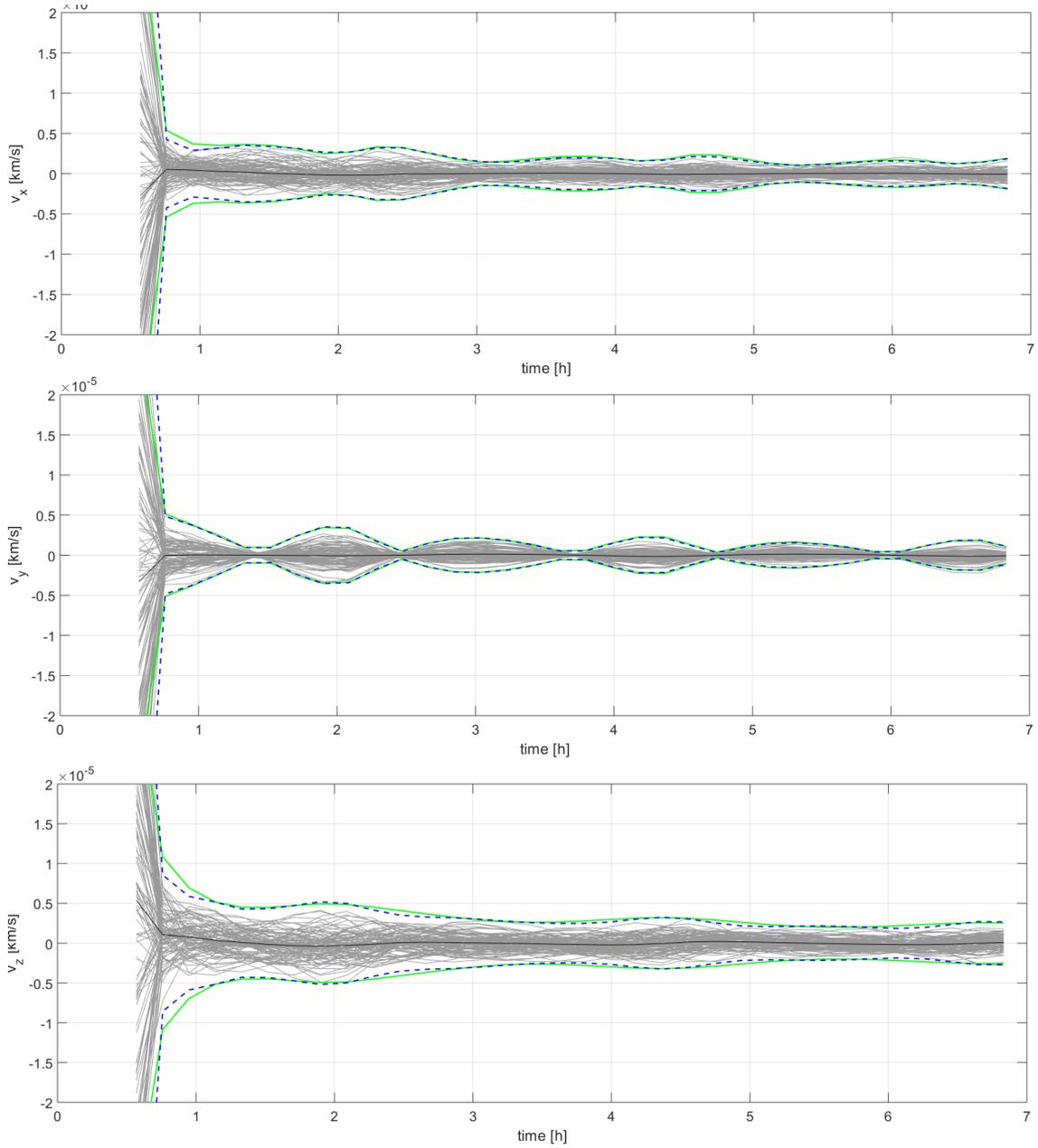
$$\sigma_v = \sqrt{\sigma_{vx}^2 + \sigma_{vy}^2 + \sigma_{vz}^2} \quad (138)$$

Fig. 5 shows the improvement of HODAKF with respect to a DA high-order propagation filter. The blue line is the high-order extended Kalman filter with a nonlinear propagation step, but a linear update. This line (associated to<sup>12</sup>) represents the real covariance behavior of the filter when the update is linear: the green line, instead, represents the covariance estimation of the classical EKF. The EKF relies too much on the measurement knowledge, indeed its Kalman gain will make the filter rely more than it should on the



**Figure 3. Position error. HODAKF order:  $n = 2$ .**

measurement correction. Thus, the EKF considers the measurement dispersion to be better than it actually is, leading to an overconfidence on the state covariance update. The green line, therefore, does not respect the actual true behavior of the covariance. The high-order EKF, referred in the Fig. 5 as EKFDA2,<sup>23</sup> is able to describe the true behavior of the state standard deviation since the measurement covariance has been predicted with an higher order. A well tuned UKF would also produce the same result because it has a liner update. Looking now at the red line, it is possible to see the improvement of the HODAKF: during the first steps, the filter is able to decrease the state uncertainty way faster than the other filters. Because of the nonlinear update, the Kalman gain better balance the state prediction and update correction thanks to the improved evaluation of the measurement covariance matrix.

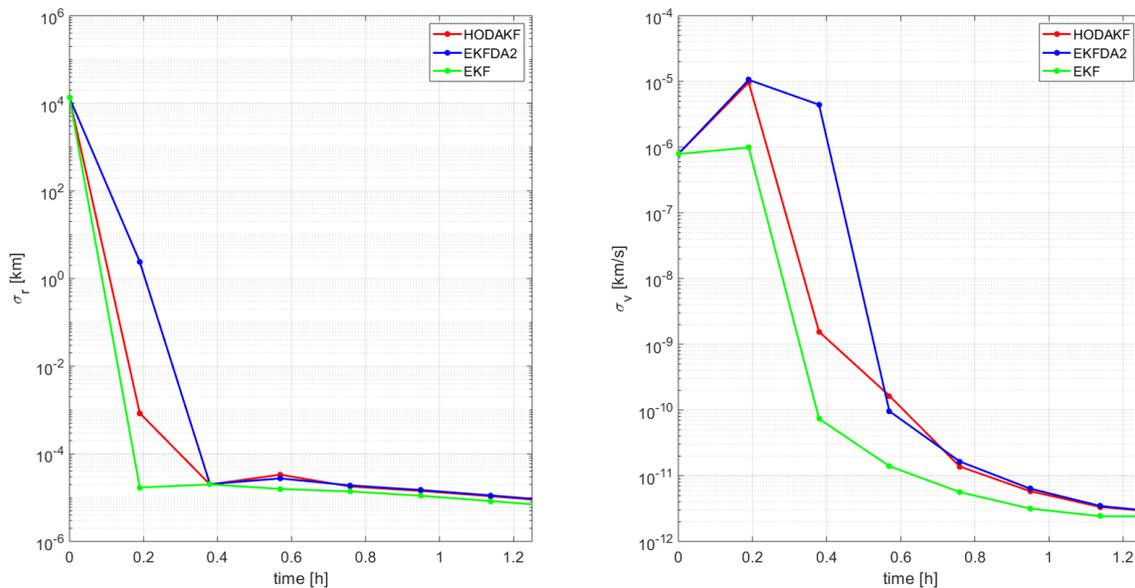


**Figure 4. Velocity error. HODAKF order:  $n = 2$ .**

## CONCLUSIONS

The second-order polynomial update has been proven to increase the accuracy of the filtering problem, especially in the estimation of the state covariance in the first iteration steps. The HODAKF is able to improve the estimation of the state of a system subjected to non Gaussian noise, by reducing, also, its uncertainties. The second order polynomial update achieves better results with respect to other polynomial filters, as well as for UKF, that just rely on a high-order propagation only, leaving the update linear.

Future developments will require a more accurate moments approximation for multi-variable systems,



**Figure 5. Standard deviation estimation, notice that the EKF is unrealistically over-optimistic.**

where the filter does not need to rely on boundaries set by the moments monotonicity. Moreover, future work requires the realization of a software that demands less computational work from the machine: indeed, the creation of new adaptive libraries can set the HODAKF suitable for big systems and drastically reduce the computational time.

## REFERENCES

- [1] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, Vol. 82, March 1960, pp. 35–45. doi:10.1115/1.3662552.
- [2] R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction," *Journal of Basic Engineering*, Vol. 83, March 1961, pp. 95–108. doi:10.1115/1.3658902.
- [3] B. D. Tapley, B. E. Schutz, and G. H. Born, *Statistical Orbit Determination*. Elsevier Academic Press, 2004.
- [4] A. Gelb, ed., *Applied Optimal Estimation*. Cambridge, MA: The MIT press, 1974.
- [5] J. Junkins and P. Singla, "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.
- [6] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, Vol. 92, March 2004, pp. 401–422. doi: 10.1109/JPROC.2003.823141.
- [7] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, Vol. 45, March 2000, pp. 477–482. doi: 10.1109/9.847726.
- [8] K. Fujimoto, D. J. Scheeres, and K. T. Alfriend, "Analytical Nonlinear Propagation of Uncertainty in the Two-Body Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 2, 2012, pp. 497–509. doi: 10.2514/1.54385.
- [9] R. Park and D. Scheeres, "Nonlinear mapping of Gaussian statistics: theory and applications to spacecraft trajectory design," *Journal of guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1367–1375.
- [10] R. S. Park and D. J. Scheeres, "Nonlinear Semi Analytic Methods for Trajectory Estimation," *Journal of Guidance Control and Dynamics*, Vol. 30, No. 6, 2007, pp. 1668–1676.
- [11] M. Majji, J. Junkins, and J. Turner, "A High Order Method for Estimation of Dynamic Systems," *The Journal of the Astronautical Sciences*, Vol. 56, No. 3, 2008, pp. 401–440.
- [12] M. Valli, R. Armellin, P. Di Lizia, and M. Lavagna, "Nonlinear mapping of uncertainties in celestial mechanics," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 1, 2012, pp. 48–63.

- [13] D. Alspach and H. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Transactions on Automatic Control*, Vol. 17, August 1972, pp. 439–448. doi:10.1109/SAP.1970.270017.
- [14] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*, Vol. 64 of *Mathematics in Sciences and Engineering*. New York, New York 10003: Academic Press, 1970.
- [15] A. De Santis, A. Germani, and M. Raimondi, "Optimal quadratic filtering of linear discrete-time non-Gaussian systems," *Automatic Control, IEEE Transactions on*, Vol. 40, No. 7, 1995, pp. 1274–1278, doi:10.1109/9.400478.
- [16] F. Carravetta, A. Germani, and N. Raimondi, "Polynomial filtering of discrete-time stochastic linear systems with multiplicative state noise," *Automatic Control, IEEE Transactions on*, Vol. 42, No. 8, 1997, pp. 1106–1126. doi:10.1109/9.618240.
- [17] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, first ed., 1965.
- [18] P. Di Lizia, M. Massari, and F. Cavenago, "Assessment of onboard DA state estimation for spacecraft relative navigation," 2017.
- [19] J.-M. Dufour, "Properties of moments of random variables," 2003.
- [20] M. Massari, P. Di Lizia, F. Cavenago, and A. Wittig, "Differential Algebra software library with automatic code generation for space embedded applications," *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, p. 0398, 2018.
- [21] M. Rasotto, A. Morselli, A. Wittig, M. Massari, P. Di Lizia, R. Armellin, C. Valles, and G. Ortega, "Differential algebra space toolbox for nonlinear uncertainty propagation in space dynamics," 2016.
- [22] 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, 1918, pp. 134–139.
- [23] F. Cavenago, P. Di Lizia, M. Massari, S. Servadio, and A. Wittig, "DA-based nonlinear filters for spacecraft relative state estimation," *2018 Space Flight Mechanics Meeting*, 2018, p. 1964.