Spatially Localized Kalman Filtering for Data Assimilation

Oscar Barrero*, Dennis S. Bernstein**, and Bart L. R. De Moor*

Abstract— **In data assimilation applications involving large scale systems, it is often of interest to estimate a subset of the system states. For example, for systems arising from discretized partial differential equations, the chosen subset of states can represent the desire to estimate state variables associated with a subregion of the spatial domain. The use of a spatially localized Kalman filter is motivated by computing constraints arising from a multi-processor implementation of the Kalman filter as well as a lack of global observability in a nonlinear system with an extended Kalman filter implementation. In this paper we derive an extension of the classical output injection Kalman filter in which data is locally injected into a specified subset of the system states.**

I. INTRODUCTION

The classical Kalman filter provides optimal least-squares estimates of all of the states of a linear time-varying system under process and measurement noise. In many applications, however, optimal estimates are desired for a specified subset of the system states, rather than all of the system states. For example, for systems arising from discretized partial differential equations, the chosen subset of states can represent the desire to estimate state variables associated with a subregion of the spatial domain. However, it is well known that the optimal state estimator for a subset of system states coincides with the classical Kalman filter.

For applications involving high-order systems, it is often difficult to implement the classical Kalman filter, and thus it is of interest to consider computationally simpler filters that yield suboptimal estimates of a specified subset of states. One approach to this problem is to consider reduced-order Kalman filters. Such reduced-complexity controllers provide estimates of the desired states that are suboptimal relative to the classical Kalman filter [1–3, 6, 7]. Alternative variants of the classical Kalman filter have been developed for computationally demanding data assimilation applications such as weather forecasting [8–10], where the classical Kalman filter gain and covariance are modified so as to reduce the computational requirements.

The present paper is motivated by computationally demanding applications such as those discussed in [8–10]. For such applications, a high-order simulation model is assumed to be available, and the derivation of a reduced-order filter in the sense of [1–3, 6, 7] is not feasible due to the lack of a tractable analytic model. Instead, we consider the use of a full-order state estimator based directly on the simulation model. However, rather than implementing the classical output injection Kalman filter, we derive a suboptimal *spatially localized Kalman filter* in which the filter gain is constrained a priori to reflect the desire to estimate a specified subset of states. Our development is also more general than the classical treatment since the state dimension can be time varying. This extension is useful for variable-resolution discretizations of partial differential equations.

The use of a spatially localized Kalman filter in place of the classical Kalman filter is motivated by the use of the ensemble Kalman filter for nonlinear systems. For systems with sparse measurements, observability may not hold for the entire system. In this case, the spatially localized Kalman filter can be used to obtain state estimates for the observable portion of the system.

II. SPATIALLY LOCALIZED KALMAN FILTER (SLKF)

We begin by considering the discrete-time dynamical system

$$x_k = A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1}, \quad k \geq 0, \tag{1}$$

with output

$$y_k = C_k x_k + v_k, \tag{2}$$

where $x_k \in \mathbb{R}^{n_k}$, $x_{k-1} \in \mathbb{R}^{n_{k-1}}$, $u_{k-1} \in \mathbb{R}^{m_{k-1}}$, $y_k \in \mathbb{R}^{l_k}$, and A_{k-1}, B_{k-1}, C_k are known real matrices of appropriate size. The input u_{k-1} and output y_k are assumed to be measured, and $w_{k-1} \in \mathbb{R}^{n_{k-1}}$ and $v_k \in \mathbb{R}^{l_k}$ are zero-mean noise processes with known variances and correlation given by Q_{k-1}, R_k, and S_k, respectively. We assume that Q_{k-1}, R_k, and S_k are positive definite. Note that the dimension n_k of the state x_k can be time varying, and thus $A_{k-1} \in \mathbb{R}^{n_k \times n_{k-1}}$ is not necessarily square.

The problem of estimating a subset of states of (1) from measurements of the output (2) is discussed in this section.

A. Estimation Problem

Consider the discrete-time dynamical system described by (1) and (2). For this system, we take a state estimator of the form

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \Gamma_k K_k (y_k - \hat{y}_{k|k-1}), \quad k \geq 0, \tag{3}$$

with output

$$\hat{y}_{k|k-1} = C_k \hat{x}_{k|k-1}. \tag{4}$$

where $\hat{x}_{k|k} \in \mathbb{R}^{n_k}$ is the estimation of x_k using the measurements y_i for $0 \leq i \leq k$, $\hat{y}_{k|k-1} \in \mathbb{R}^{l_k}$, $\Gamma_k \in \mathbb{R}^{n_k \times p_k}$, and $K_k \in \mathbb{R}^{p_k \times l_k}$. The nontraditional feature of (3) is the presence

This research was supported in part by the National Science Foundation under grant.

*Department of Electrical Engineering, ESAT-SCD/SISTA, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium, obarrero@esat.kuleuven.ac.be, bart.demoor@esat.kuleuven.ac.be

**Department of Aerospace Engineering, The University of Michigan, Ann Arbor, MI 48109-2140, USA, dsbaero@umich.edu

of the term Γ_k, which, in the classical case of output injection, is the identity matrix. Here, Γ_k constrains the state estimator so that only states in the range of Γ_k are directly affected by the gain K_k. We assume that Γ_k has full column rank for all $k \geq 0$.

In order to find the optimal gain K_k, the first step is to project $x_{k-1|k-1}$ ahead via (1) using

$$\hat{x}_{k|k-1} = A_{k-1}\hat{x}_{k-1|k-1} + B_{k-1}u_{k-1}. \tag{5}$$

Then, define the prior state estimation error by

$$e_{k|k-1} \stackrel{\triangle}{=} x_k - \hat{x}_{k|k-1}. \tag{6}$$

Substituting (5) and (1) into (6) we obtain

$$e_{k|k-1} = A_{k-1}e_{k-1|k-1} + w_{k-1}. \tag{7}$$

Now, define the prior error covariance matrix by

$$P_{k|k-1} \stackrel{\triangle}{=} \mathcal{E}[e_{k|k-1}e_{k|k-1}^{\mathrm{T}}], \tag{8}$$

where $\mathcal{E}$ denotes expected value. Hence,

$$P_{k|k-1} = A_{k-1}P_{k-1|k-1}A_{k-1}^{\mathrm{T}} + Q_{k-1}. \tag{9}$$

Next, define the state estimation error

$$e_{k|k} \stackrel{\triangle}{=} x_k - \hat{x}_{k|k}, \tag{10}$$

and the weighted estimation error covariance matrix

$$J_k(K_k) \stackrel{\triangle}{=} \mathcal{E}[(L_k e_{k|k})^T (L_k e_{k|k})], \tag{11}$$

where $L_k \in \mathbb{R}^{q_k \times n_k}$ determines the weighted error components. Then, the weighted estimation error can be obtained as

$$J_k(K_k) = \mathrm{tr}\ (P_{k|k}M_k), \tag{12}$$

where the error covariance matrix $P_{k|k} \in \mathbb{R}^{n_k \times n_k}$ is defined by

$$P_{k|k} \stackrel{\triangle}{=} \mathcal{E}[e_{k|k}e_{k|k}^{\mathrm{T}}], \tag{13}$$

and $M_k \in \mathbb{R}^{n_k \times n_k}$ by

$$M_k \stackrel{\triangle}{=} L_k^{\mathrm{T}} L_k. \tag{14}$$

Now, substituting (3) into (10) yields

$$e_{k|k} = x_k - \hat{x}_{k|k-1} - \Gamma_k K_k (y_k - C_k\hat{x}_{k|k-1}), \tag{15}$$

and using (15) with (13) implies

$$P_{k|k} = \hat{A}_k P_{k|k-1} \hat{A}_k^{\mathrm{T}} + \hat{Q}_k, \tag{16}$$

where

$$\begin{aligned}
P_{k|k-1} &\stackrel{\triangle}{=} \mathcal{E}[e_{k|k-1}e_{k|k-1}^{\mathrm{T}}], && (17)\\
e_{k|k-1} &\stackrel{\triangle}{=} x_k - \hat{x}_{k|k-1}, && (18)\\
\hat{A}_k &\stackrel{\triangle}{=} I_{n_k} - \Gamma_k K_k C_k, && (19)\\
\hat{Q}_k &\stackrel{\triangle}{=} \Gamma_k K_k \tilde{R}_k K_k^{\mathrm{T}} \Gamma_k^{\mathrm{T}} - S_k K_k^{\mathrm{T}} \Gamma_k^{\mathrm{T}} - \Gamma_k K_k S_k^{\mathrm{T}}, && (20)\\
\tilde{R}_k &\stackrel{\triangle}{=} C_k S_k + S_k^{\mathrm{T}} C_k^{\mathrm{T}} + R_k, && (21)\\
S_k &\stackrel{\triangle}{=} \mathcal{E}[w_{k-1}v_k^{\mathrm{T}}]. && (22)
\end{aligned}$$

Hence (12) becomes

$$J_k(K_k) = \mathrm{tr}\ [(\hat{A}_k P_{k|k-1} \hat{A}_k^{\mathrm{T}} + \tilde{Q}_k)M_k]. \tag{23}$$

To obtain the optimal gain K_k we set $\partial J_k(K_k)/\partial K_k = 0$, which gives

$$K_k = (\Gamma_k^{\mathrm{T}} M_k \Gamma_k)^{-1} \Gamma_k^{\mathrm{T}} M_k \hat{S}_k \hat{R}_k^{-1}, \tag{24}$$

with

$$\begin{aligned}
\hat{S}_k &\stackrel{\triangle}{=} S_k + P_{k|k-1}C_k^{\mathrm{T}}, \in \mathbb{R}^{n_k \times l_k}, && (25)\\
\hat{R}_k &\stackrel{\triangle}{=} \tilde{R}_k + C_k P_{k|k-1} C_k^{\mathrm{T}}, \in \mathbb{R}^{l_k \times l_k}. && (26)
\end{aligned}$$

To update the error covariance matrix, we first note that

$$\Gamma_k K_k = \pi_k \hat{S}_k \hat{R}_k^{-1}, \tag{27}$$

where $\pi_k \in \mathbb{R}^{n_k \times n_k}$ is defined by

$$\pi_k \stackrel{\triangle}{=} \Gamma_k (\Gamma_k^{\mathrm{T}} M_k \Gamma_k)^{-1} \Gamma_k^{\mathrm{T}} M_k. \tag{28}$$

Note that π_k is an oblique projector, that is, $\pi_k^2 = \pi_k$. Now using (27) with (16) yields the error covariance matrix update equation

$$P_{k|k} = P_{k|k-1} + \pi_{k\perp} \hat{S}_k \hat{R}_k^{-1} \hat{S}_k^{\mathrm{T}} \pi_{k\perp}^{\mathrm{T}} - \hat{S}_k \hat{R}_k^{-1} \hat{S}_k^{\mathrm{T}}, \tag{29}$$

where the complementary projector $\pi_{k\perp}$ is defined by

$$\pi_{k\perp} \stackrel{\triangle}{=} I_{n_k} - \pi_k. \tag{30}$$

If either $M_k = I_{n_k}$ or $L_k = \Gamma_k^{\mathrm{T}}$, then π_k is the orthogonal projector

$$\pi_k = \Gamma_k (\Gamma_k^{\mathrm{T}} \Gamma_k)^{-1} \Gamma_k^{\mathrm{T}}. \tag{31}$$

On the other hand, if $p_k = q_k$, so that $L_k \in \mathbb{R}^{p_k \times p_k}$, then, it can be shown that

$$\pi_k = \Gamma_k (L_k \Gamma_k)^{-1} L_k. \tag{32}$$

Specializing to the case $S_k = 0$, $\Gamma_k = I_{n_k}$, and $L_k = I_{n_k}$, so that $\pi_{k\perp} = 0$, yields the familiar Riccati update equation

$$P_{k|k} = P_{k|k-1} - P_k C_k^{\mathrm{T}} \hat{R}_k^{-1} C_k P_k. \tag{33}$$

In the classical case, $n_k = n$ for all $k \geq 0$. Summarizing the algorithm we have, for $k = 1, 2, \ldots$, the following steps:

1. Project ahead the error covariance matrix and the estimated states using (9) and (5).
2. Compute the SLFK gain using (27).
3. Update the estimated states using (3)
4. Update the error covariance matrix using (29)

III. Square Root Formulation of the SLKF

To avoid numerical problems when computing the SLKF, a square root formulation, for the case $S_k = 0$ is presented is this section. We can rewrite (29) as

$$P_{k|k} = P_{1_{k|k}} + P_{2_{k|k}}, \tag{34}$$

where $P_{1_{k|k}}$ and $P_{2_{k|k}} \in \mathbb{R}^{n_k \times n_k}$ are defined by

$$P_{1_{k|k}} \triangleq P_{k|k-1} - \hat{S}_k \hat{R}_k^{-1} \hat{S}_k^{\mathrm{T}}, \tag{35}$$

$$P_{2_{k|k}} \triangleq \pi_{k\perp} \hat{S}_k \hat{R}_k^{-1} \hat{S}_k^{\mathrm{T}} \pi_{k\perp}^{\mathrm{T}}. \tag{36}$$

Hence, the square root form of 34 can be written as

$$P_{k|k} = F_{k|k} F_{k|k}^{\mathrm{T}} \tag{37}$$

with $F_{k|k} \in \mathbb{R}^{n_k \times (n_k + l_k)}$ defined as

$$F_{k|k} \triangleq \left[F_{1_{k|k}} \; F_{2_{k|k}}\right], \tag{38}$$

where $F_{1_{k|k}} \in \mathbb{R}^{n_k \times n_k}$ and $F_{2_{k|k}} \in \mathbb{R}^{n_k \times l_k}$ satisfy

$$P_{1_{k|k}} = F_{1_{k|k}} F_{1_{k|k}}^{\mathrm{T}}, \tag{39}$$

$$P_{2_{k|k}} = F_{2_{k|k}} F_{2_{k|k}}^{\mathrm{T}}. \tag{40}$$

To compute $F_{1_{k|k}}$, first notice that the Schur complement of $\hat{R}_k$ in M_k is $P_{1_{k|k}}$,

$$M_k = \begin{bmatrix} \hat{R}_k & \hat{S}_k^{\mathrm{T}} \\ \hat{S}_k & P_{1_{k|k-1}} \end{bmatrix} \tag{41}$$

Now specializing to the case $S_k = 0$ we have

$$M_k = \begin{bmatrix} R_k + C_k P_{k|k-1} C_k^{\mathrm{T}} & C_k P_{k|k-1} \\ P_{k|k-1} C_k^{\mathrm{T}} & P_{k|k-1} \end{bmatrix}. \tag{42}$$

Hence, the square root form of (42) is defined by

$$M_k \triangleq \beta_k \beta_k^{\mathrm{T}} \tag{43}$$

with $\beta_k \in \mathbb{R}^{(l_k + n_k) \times (l_k + N_{q_k})}$ given by

$$\beta_k = \begin{bmatrix} L_{R_k} & C_k F_{k|k-1} \\ 0 & F_{k|k-1} \end{bmatrix} \tag{44}$$

with N_{q_k} the order rank approximation of $F_{k|k-1}$ defined below in (55), where $R_k \triangleq L_{R_k} L_{R_k}^{\mathrm{T}}$ and $P_{k|k-1} \triangleq F_{k|k-1} F_{k|k-1}^{\mathrm{T}}$. Next, a lower triangular QR decomposition of β_k yields

$$\begin{bmatrix} L_{R_k} & C_k F_{k|k-1} \\ 0 & F_{k|k-1} \end{bmatrix} U_k = \begin{bmatrix} H_k & 0 \\ J_k & F_{1_{k|k}} \end{bmatrix}, \tag{45}$$

where $U_k \in \mathbb{R}^{(l_k + N_{q_k}) \times (l_k + N_{q_k})}$ is orthogonal. As a consequence, a square root factorization of M_k is given by

$$M_k = \begin{bmatrix} H_k & 0 \\ J_k & F_{1_{k|k}} \end{bmatrix} \begin{bmatrix} H_k^{\mathrm{T}} & J_k^{\mathrm{T}} \\ 0 & F_{1_{k|k}}^{\mathrm{T}} \end{bmatrix}, \tag{46}$$

from which, assuming H_k is nonsingular, it follows that

$$K_k = \pi_k J_k H_k^{-1}, \tag{47}$$

$$\hat{R}_k = H_k H_k^{\mathrm{T}}, \tag{48}$$

$$P_{k|k-1} C_k^{\mathrm{T}} = J_k H_k^{\mathrm{T}}. \tag{49}$$

Then, to find $F_{2_{k|k}}$ we substitute (48) and (49) into (36)

$$P_{2_{k|k}} = \pi_{k\perp} J_k H_k^{\mathrm{T}} (H_k H_k^{\mathrm{T}})^{-1} H_k J_k^{\mathrm{T}} \pi_{k\perp}^{T} \tag{50}$$

$$P_{2_{k|k}} = \pi_{k\perp} J_k J_k^{\mathrm{T}} \pi_{k\perp}^{\mathrm{T}}, \tag{51}$$

from where

$$F_{2_{k|k}} = \pi_{k\perp} J_k. \tag{52}$$

As a result, the recursive SLKF algorithm can be summarized as follows:

1) Compute the $F_{k|k-1}$ via

$$F_{k|k-1} = \left[A_{k-1} F_{k-1|k-1} \; L_{Q_{k-1}}\right], \tag{53}$$

where $Q_{k-1} \triangleq L_{Q_{k-1}} L_{Q_{k-1}}^{\mathrm{T}}$.

2) Compute a reduced rank approximation of $F_{k|k-1}$ to avoid the dimensions of (53) to increase in each iteration. A efficient way to do this is to apply the same trick as in [12]. First, compute the eigenvalue decomposition of $P_{k|k-1}^{\mathrm{T}}$

$$F_{k|k-1}^{\mathrm{T}} F_{k|k-1} = V_k D_k V_k^{\mathrm{T}}, \tag{54}$$

then, it turns out that the reduced rank approximation of (53) can be yielded by

$$F_{k|k-1}^{*} = F_{k|k-1} V_{k_{(1:n_k, 1:N_{q_k})}}, \tag{55}$$

where $N_{q_k} \le n_k$ is the order chosen to approximate (53), hence $F_{k|k-1}^{*} \in \mathbb{R}^{n_k \times N_{q_k}}$.

3) Then use (55) to compute the lower triangular QR decomposition of (44) to obtain $F_{1_{k|k}}$, (47), (48), and (49).

4) Compute $F_{2_{k|k}}$ using (52) and finally $F_{k|k}$ using (38)

IV. Mass-Spring System Example

To illustrate the performance of the SLKF a simple LTI mass-spring system is used. The state space representation in continuous time of this system is given by

$$\begin{bmatrix} \dot{z} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} A_z & A_x \\ I_N & 0_N \end{bmatrix} \begin{bmatrix} z \\ x \end{bmatrix} + \begin{bmatrix} I_N \\ 0_N \end{bmatrix} u, \tag{56}$$

with z_i and x_i the velocity and the position of the i-th mass, respectively, and N the number of masses. In this example the nodes at the extremes are assumed to be fixed, so, the number of analyzed nodes is equal to the number of masses. $A_x \in \mathbb{R}^{N \times N}$ is a tridiagonal matrix defined by

$$A_x \triangleq$$

$$\begin{bmatrix} -(k_1+k_2)/m_1 & k_2/m1 & 0 & \cdots \\ k_2/m_2 & -(k_2+k_3)/m_2 & k_3/m_2 & \vdots \\ 0 & \ddots & \ddots & \ddots \\ \vdots & \cdots & k_N/m_N & -(k_N+k_{N+1})/m_N \end{bmatrix}, \tag{57}$$

with k_i and m_i the spring constant and mass of the i-th node, respectively. $A_z \in \mathbb{R}^{N \times N}$ a diagonal matrix defined by

$$A_z \triangleq \begin{bmatrix} -c_1/m_1 & 0 & \cdots \\ 0 & \ddots & \vdots \\ \vdots & \cdots & -c_N/m_N \end{bmatrix}, \tag{58}$$

where c_i the friction coefficient of the i-th mass.

For simplicity of the analysis we take the parameters in each node to be equal to the others; hence, $m_i = 1Kg$, $k_i = 5Kg/s^2$, and $c_i = 5Kg/s$, with $1 \leq i \leq N$. We selected $N = 50$, $m = 1$ i.e., one input applied in the node 25 and defined by

$$u_{25}(t) = 30sin(t/2 + \pi/3), \tag{59}$$

Next, the system is discretized using the zero order hold method taking a sampling time $T_s = 0.1s$. Now, the discrete-system can be represented by

$$x_{k+1} = A_d x_k + B_d u_k + w_k, \tag{60}$$

where $A_d \in \mathbb{R}^{n\times n}$, with $n = 2N$, $B_d \in \mathbb{R}^n$, and $w_k \in \mathbb{R}^n$ is the process noise caused by the discretization. With output

$$y_k = C_d x_k + v_k \tag{61}$$

where $v_k \in \mathbb{R}^l$ is the measurement noise. The process and measurement noise are assumed to be uncorrelated white noises.

In order to apply the SLKF to this problem, first we define the region where we want the state estimation to be focused. Hence, we take the node 25, this node is taken as the measurement point for the input as well as the outputs; velocity and position. Therefore we specify Γ_k such that the SLKF concentrates around this region. Assuming $M_k = I_n$, the weighted matrix Γ_k is constructed such that the entries of each column are taken from a gaussian function with mean the position of the analyzed state, notice that for each analyzed state one column is needed, this makes the state estimation around the region of interest to be smooth, avoiding numerical problems in the model integration. For this example $p_k = l_k = 2$.

A. Results

Two experiments with different Signal-to-Noise ratio (SNR) were carried out, the first one with 6dB and the second with 1dB.

Figures 1 and 2 show a comparison of the performance of the SLKF to the classical KF at different locations, specifically, 2, 10, 15, and 25. General speaking, it can be seen that under high SNR conditions SLKF performs similar to KF for any node, whereas for low SNR conditions the performance of the SLKF is worst in the nodes far from the measurement point and similar in the nodes around it. This is due to the spatially localized strategy used in the SLKF.

Figure 2 compare the estimation of the velocity of the SLKF to the classical KF for the case SNR=1dB. Again, both filters work well at the node 25, but SLKF reduces its performance compared to the previous case in the other places, while the KF keeps its performance. As a matter of fact, the classical KF is expected to do better than SLKF in regions apart from the measurement point because it is not restricted to certain region, whereas the SLKF satisfies the localization constraint.

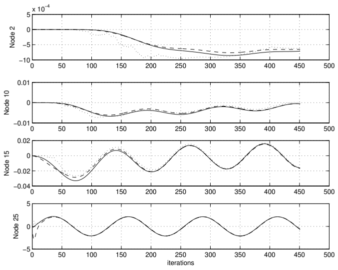

Fig. 1. Estimation of velocity with a SNR=6dB. Solid line, original state, dotted line classical KF estimation, and dashed line SLKF estimation.

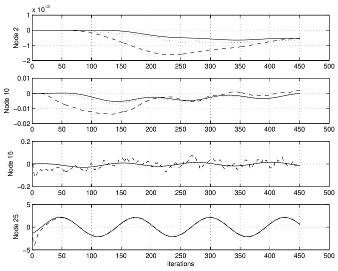

Fig. 2. Estimation of velocity with a SNR=1dB. Solid line, original state, dotted line classical KF estimation, and dashed line SLKF estimation.

Summarizing, we can say that the estimation of the SLKF and the classical KF are the same under high SNR conditions for all the states, but under low SNR conditions, the estimation of the SLKF is reliable just into the interested region.

V. The Ensemble SLKF (EnSLKF)

Based on the ensemble Kalman filter (EnKF) [13] the previous results can be extended to the nonlinear case. Therefore, a short introduction to the EnKF is first given before presenting the EnSLKF.

We write an uncertain nonlinear model, as the discrete-time stochastic differential equation

$$x_{k+1} = \mathcal{A}_k(x_k) + \eta_k, \tag{62}$$

with output

$$y_k = \mathcal{C}_k(x_k) + v_k, \tag{63}$$

and where η_k is the model error at time k defined by

$$\eta_k \triangleq G_k(x_k)\delta_k, \tag{64}$$

with $G_k(x_k) \in \mathbb{R}^{n\times l_{m_k}}$ is a state-dependent matrix, whose covariance $Q_k \in \mathbb{R}^{n\times n}$ is

$$Q_k \triangleq \mathcal{E}[G_k(x_k)G_k(x_k)^{\mathrm{T}}], \tag{65}$$

and $\delta_k \in \mathbb{R}^{l_{m_k}}$ is a process we want to approximate as white noise. Equation (62) implies that even if the initial state is known precisely, future model states cannot since unknown random model errors are continually added.

The main difference between the KF and the EnKF is that the error covariance matrices for the prior and the current estimate, $P_{k|k-1}$ and $P_{k|k}$, are in the Kalman filter defined in terms of the true state, while in the EnKF in terms of an ensemble of forecasted model states.

Thus, the EnKF algorithm can be summarized as follows. First, generate an initial ensemble $\mathbf{X}_{k-1|k-1} \in \mathbb{R}^{n_k \times N}$ which properly represent the error statistics of the initial guess for the model state.

1) Update the ensemble members of $\mathbf{X}_{k-1|k-1}$ for $i = 1,\ldots,N$ according to

$$x^i_{k|k-1} = \mathcal{A}_k(x^i_{k-1|k-1})$$

2) Compute the ensemble prior error covariance matrix,

$$E_{k|k-1} \triangleq \left[\hat{x}^1_{k|k-1} - \bar{\hat{x}}_{k|k-1}, \ldots, \hat{x}^N_{k|k-1} - \bar{\hat{x}}_{k|k-1}\right], \quad (66)$$

3) Compute the EnKF gain

$$L_{e_k} = \hat{P}_{k|k-1}\mathcal{C}_k^{\mathrm{T}}(\mathcal{C}_k\hat{P}_{k|k-1}\mathcal{C}_k^{\mathrm{T}} + R_{e_k})^{-1}, \quad (67)$$

4) Update the ensemble members

$$\hat{x}^i_{k|k} = \hat{x}^i_{k|k-1} + L_{e_k}\left(y^i_k - \mathcal{C}_k(\hat{x}^i_{k|k-1})\right), \quad (68)$$

5) Compute the state estimation taking the mean of the updated ensemble members

$$\hat{x}_{k|k} \triangleq \frac{1}{N}\sum_{i=1}^{N}\hat{x}^i_{k|k}. \quad (69)$$

A. The EnSLKF

Comparing the algorithms of the SLKF (section II) to the EnKF, we observe that the error covariance update steps are omitted in the EnKF due to the fact that these steps are implicit when the ensembles $\mathbf{X}_{k|k-1}$, and $\mathbf{X}_{k|k}$ are updated. As a result, by combining the ideas of the SLKF and EnKF we can easily obtain a nonlinear version of the SLKF.

The main difference between the two algorithms is the computation of the filter gain, while for the EnKF is given by (67) for the SLKF by (27). Hence, the EnSLKF gain can be defined as

$$\Gamma_k K_k \triangleq \pi_k \hat{S}_{e_k}\hat{R}^{-1}_{e_k}, \quad (70)$$

with $S_{e_k} \in \mathbb{R}^{n_k \times l_k}$ defined by

$$\hat{S}_{e_k} \triangleq S_{e_k} + P_{k|k-1}\mathcal{C}_k^{\mathrm{T}}. \quad (71)$$

Apart from this change, notice that the rest of the algorithm for the EnSLKF stays the same to the EnKF.

On the other hand, note in (70) that for n_k large π_k is too big to store, so, we can rewrite $\Gamma_k K_k$ such that the computation of the EnSLKF gain can be done more efficiently. Hence,

$$\Gamma_k K_k \triangleq \Upsilon_k \Phi_k, \quad (72)$$

where $\Upsilon_k \in \mathbb{R}^{n_k \times p_k}$ is defined by

$$\Upsilon_k \triangleq \Gamma_k(\Gamma_k^{\mathrm{T}} M_k \Gamma_k)^{-1}, \quad (73)$$

and $\Phi_k \in \mathbb{R}^{p_k \times l_k}$ is defined by

$$\Phi_k \triangleq \Gamma^{\mathrm{T}}\hat{S}_{e_k}\hat{R}^{-1}_{e_k} \quad (74)$$

Consequently, the term Υ_k can be computed off-line and stored in memory, while Φ_k is computed once each iteration.

B. Space Weather Forecast Example

As an example a Magneto-Hydrodynamics (MHD) system is taken. MHD systems are used to simulate the plasma in the magnetosphere around the earth. Therefore, in this example the SLKF is used to predict the behaviour of the magnetosphere subjected to a magnetic solar storm in certain regions where satellites for measuring might be located.

This system was simulated with the VAC code [5] with the following parameters:

- Grid size: 34×54 with $0 \le x \le 0.2$ and $0 \le y \le 1$
- Initial conditions of the state space variables:
  - Mass density, $\rho = 1.0Kg/m^3$
  - Velocity in x- and y-directions, $V_x = 20m/s$, $V_y = 0m/s$
  - Pressure, $p = 1.0Kg/ms^2$
  - Magnetic field in x- and y-directions, $B_x = 0mT$, $B_y = 1.0mT$
- ratio of specific heats, $\gamma = 5/3$
- Simulation sample time, 1×10^{-4} seconds
- Data assimilation sample time, 4×10^{-4} seconds
- Spatial discretization method, Total Variation Diminishing Lax-Friedrich, TVDLF.

As a result the order of the system is $n = 11016$, 6 state space variables and 34×54 grid-points. To excite the system, a square sinusoidal wave for B_y and V_x was generated at the left-hand boundary, simulating a magnetic storm. The variation of B_y were from 1 to 1.5 mT while for V_x from 20 to 30 m/s. Notice that the earth is located on the right-hand boundary.

To apply the EnSLKF to this example, we took 6 measurement points randomly represented by black triangles in Figure 3. Then, we assumed $M = I_n$, in this example the subscript k is dropped because the system is time invariant, and defined the columns of Γ as gaussian functions with center the measurement points, $\sigma_x = 4$ grid-points, and $\sigma_y = 6$ grid-points. The number of members for the initial ensemble was 300.

Adopting this sort of structure for Γ we observed a good stability of the simulation code, different than the one reported in [4] where the VAC code crashed when less than 10 measurement points were taken. This is due to the fact that taking a smooth function as weighting function for Γ makes the changes of the SLKF estimations to be smooth around the localized region . Hence, the data assimilation process reduces the chances of obtaining unrealistic values that could make crash the model integration code. This is an important point, because one of the major issues when a nonlinear system is simulated is to give realistic initial conditions so that the integration method can converge to a solution.

Figure 3 show the residuals between the simulated data and the SLKF estimated data along of time. In general, the estimated data around the measurement points (black triangles) exhibit smaller residuals than the ones in the rest of the grid, as expected. At the beginning of the simulation, $k = 1$ iteration, the residuals are small because the initial conditions for the simulated and estimated systems are chosen to be closed, but while the time passes, $k = 40$ to 110 iterations, the residuals increases in areas far apart from the measurement points, because the estimation is done in localized areas. Moreover, it can be seen how the residuals are reduced downstream the measurement points locate far from the earth, left-hand side, as a result of the spatially localized assimilation of the data into the model. This is an interesting result for the forecast of solar storms, because it suggests that it is not necessary to place several satellites in order to have a good prediction of the system, but to place few strategically in the path between the sun and the earth.

Close to the earth, right-hand side, is observed that the residuals are bigger, even though there are three measurement points around it. The reason is that the main dynamic changes occur in there, so, the effective region for estimation of the localized assimilation is smaller, contrary to the case when the measurement points are located on the left-hand side.

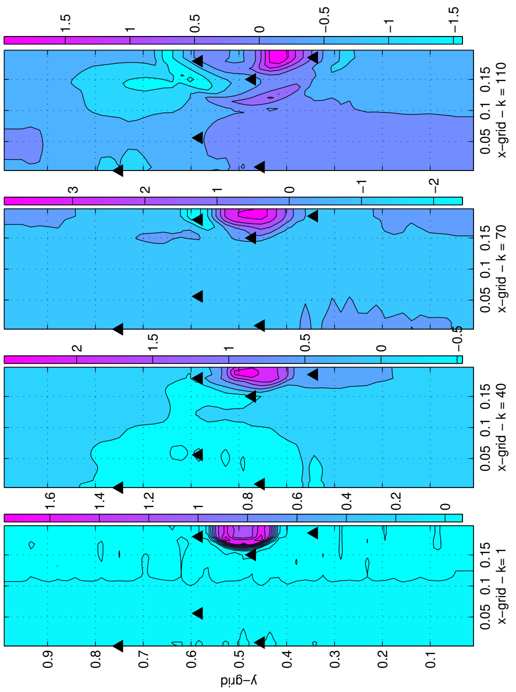

Fig. 3. Residuals of the magnetic field. The black triangles represent the measurement location.

VI. Conclusions

In this paper we presented an extension of the classical output injection Kalman filter we called Spatially Localized Kalman filter (SLKF). To avoid numerical problems in its computation a square root formulation was developed, and its performance assessed in a 1D linear mass-spring system. Despite the fact that the error covariance matrix is manipulated, the SLKF shown to work as good as the KF in the localized regions under low SNR conditions without presenting numerical problems.

The SLKF was extended to the nonlinear case using the classical EnKF formulation, it was called the EnSLKF. The EnSLKF was tested in a large scale MHD system simulating a solar storm reaching the earth's magnetosphere using the VAC code. The estimation of the filter was good around big regions of the measurement points far from the earth and small regions close to it. This happens because the dynamics changes in the vicinity of the earth are bigger than in the rest of the analyzed space. Moreover, there were no problems with the numerical stability of the data-assimilation process, even though we chose just 6 out of 1834 measurement points. However the computation complexity of the EnSLKF is still as big as the classical EnKF.

References

[1] D. S. Bernstein and D. C. Hyland, "The Optimal Projection Equations for Reduced-Order State Estimation," *IEEE Trans. Autom. Contr.*, Vol. AC-30, pp. 583-585, 1985.

[2] W. M. Haddad and D. S. Bernstein, "The Optimal Projection Equations for Discrete-Time Reduced-Order State Estimation for Linear Systems with Multiplicative White Noise," *Systems Contr. Lett.*, Vol. 8, pp. 381-388, 1987.

[3] W. M. Haddad and D. S. Bernstein, "Optimal Reduced-Order Observer-Estimators," *AIAA J. Guid. Dyn. Contr.,* Vol. 13, pp. 1126-1135, 1990.

[4] O. Barrero, B. De Moor, and D. S. Bernstein, Data Assimilation for Magneto-Hydrodynamics Systems, Internal report 04-74, ESAT-SISTA, Katholieke Universiteit Leuven, Belgium, 2004.

[5] G. Tóth and R. Keppens, Versatile Advection Code, Universiteit Utrecht, The Netherlands, 2003.

[6] W. M. Haddad, D. S. Bernstein, H.-H. Huang, and Y. Halevi, "Fixed-Order Sampled-Data Estimation," *Int. J. Contr.*, Vol. 55, pp. 129-139, 1992.

[7] C.-S. Hsieh, "The Unified Structure of Unbiased Minimum-Variance Reduced-Order Filters," *Proc. Contr. Dec. Conf.*, pp. 4871-4876, Maui, HI, December 2003.

[8] B. F. Farrell and P. J. Ioannou, "State Estimation Using a Reduced-Order Kalman Filter," *J. Atmos. Sci.*, Vol. 58, pp. 3666-3680, 2001.

[9] A. W. Heemink, M. Verlaan, and A. J. Segers, "Variance Reduced Ensemble Kalman Filtering," *Mon. Weather Rev.*, Vol. 129, pp. 1718-1728, 2001.

[10] P. Fieguth, D. Menemenlis, and I. Fukumori, "Mapping and Pseudo-Inverse Algorithms for Data Assimilation," *Proc. Int. Geoscience Remote Sensing Symp.*, pp. 3221-3223, 2002.

[11] M. Morf and T. Kailath, "Square-Root Algorithms for Least-Squares Estimation," *IEEE Trans. Autom. Contr.*, Vol. AC-20, pp. 487–497, 1975.

[12] M. Verlaan and A. W. Heemink, "Reduced Rank Square Root Filters for Large Scale Data Assimilation Problems," *2nd Int. Symp. on Assimilation of Obs. in Met. & Ocea, WMO*, pp. 247-252, 1995.

[13] G. Evensen, "The ensemble Kalman filter: theroretical formulation and practical implementation," *Ocean Dynamics*, 2003, vol. 53, pp. 343-367.