(Short Paper)

State Estimation in Linear Dynamical Systems By Partial Update Kalman Filtering

Vahid Vahidpour, Amir Rastegarnia and Saeid Sanei

Abstract

In this letter, we develop a partial update Kalman filtering (PUKF) algorithm to solve the state of a discrete-time linear stochastic dynamical system. In the proposed algorithm, only a subset of the state vector is updated at every iteration, which reduces its computational complexity, compared to the original KF algorithm. The required conditions for the stability of the algorithm are discussed. A closed-form expression for steady-state mean-square deviation (MSD) is also derived. Numerical examples are used to validate the correctness of the provided analysis. They also reveal the PUKF algorithm exhibits a trade-off between the estimation accuracy and the computational load which is extremely profitable in practical applications.

Index Terms

Partial Update, Kalman Filtering, Convergence analysis.

I. Introduction

Kalman filter (KF) is a fundamental tool in applications where the state variables of a dynamic system are required to be estimated based on a series of noisy observed information. A key factor in the implementation of KF algorithm is the dimensionality of state variables, since the computational complexity increases (even exponentially) as the dimensionality increases. Although this may not be a problem in some applications (for example, in some 2-D positioning applications), in some applications (such as positioning and numerical weather prediction) the state vector may contain variables for the locations of thousands of landmarks. So, reducing the computational complexity is a major issue for the practical use of KF such applications.

In the case of dynamic systems with a non-linear measurement model (where extended KF (EKF) is used), different methods of computational complexity reduction have been reported in the literature. In [1] it is suggested to divide the non-linear system model into linear and non-linear parts. A computationally efficient EKF for non-linear with *continuous* dynamic systems has been reported in [2]. In [3] different code optimization methods to reduce the computational complexity of EKF have been introduced. In the case of high dimentional non-linear system model with non-Gaussian data, the ensemble KF can be applied [4]. None of the mentioned methods is similar to our proposed algorithm in the current letter, which relies on the Partial updating (PU) method.

Partial updating is known as an effective method to reduce the computational complexity of adaptive filters [5], [6]. The main idea in PU is to update only a fraction of the filter coefficient vector at each iteration, instead of reducing filter order. The resulting algorithms reduce the computational complexity while keeping the estimation performance close to their counterpart

V. Vahidpour and A. Rastegarniaare with the Department of Electrical Engineering, Malayer University, Malayer 65719-95863, Iran (email: v.vahidpour@stu.malayeru.ac.ir; rastegarnia@malayeru.ac.ir).

full-update algorithms. The method was originally reported for LMS adaptive filter [7], but its extended to other adaptive filters [8]–[12]. Some distributed versions of PU adaptive filters such as [13]–[19] have been developed in the literature.

In certain applications, like network and acoustic echo cancellation, the complexity reduction accessible by partial coefficient updates is of practical importance. The main reason for employing such scheme is the finite availability of hardware multipliers frequently driven by cost, space and power consumption considerations. This problem is especially critical in Digital Signal Processing (DSP) and Field Programmable Gate Array (FPGA) implementation of adaptive signal processing algorithms. Even though the concept of partial updates is very effective in terms of addressing scarce hardware resources, it can lead to deterioration in convergence performance. After all, only a subset of all coefficients is updated, so one would expect some performance deterioration as a result [20].

In this letter, we apply the PU method to KF framework and develop the partial update Kalman filter (PUKF) to perform state estimation in linear dynamic systems. In the proposed algorithm, only a subset of the state vector is updated at every iteration. It is shown that the PUKF algorithm is able to provide an acceptable estimation performance, while the computational complexity is kept low. It is noteworthy to mention that PUKF algorithm has not been addressed in any of references [11] and [16]. Indeed, algorithm in [16] uses the Kalman filter for adaptation and learning in the agents, where nodes are allowed to receive (diffuse) only a subset of their intermediate state estimate entries. Algorithm in [11] consider widely linear model, which in this paper we have assumed a linear state-space model. Moreover, in [11] the Augmented Complex LMS algorithm is employed as the learning rule; whereas in our algorithm we have used kalman filter algorithm for adaptation and learning.

In summary, the following are the contributions of this letter:

- A new algorithm, called PU-ACLMS is proposed to control the computational complexity;
- The computational complexity for full-update KF and its partial-update (PUKF) implementations are examined thoroughly;
- The stability of PUKF algorithm in both mean and mean-square senses under certain statistical conditions is derived;
- The steady-state performance of the PUKF algorithm is evaluated theoretically in terms of the MSD metric;
- By numerical example, it is shown that the PUKF algorithm provides a trade-off between estimation accuracy and the computational load.

Throughout the letter, we adopt normal lowercase letters for scalars, bold lowercase letters for column vectors and bold uppercase letters for matrices.

The rest of the letter is organized as follows: In Section II, we briefly introduce the Kalman filtering and derive the PUKF algorithm algorithm. The convergence and steady-state analysis are provided in Section III. Performance evaluations are illustrated in Section IV. The letter is finally concluded in Section V.

II. Kalman Filtering with Partial Updates

A. *Overview of Kalman Filtering*

Consider the following stochastic dynamic model and the sequence of noisy observations

$$\mathbf{x}_{i+1} = \mathbf{F}\mathbf{x}_i + \mathbf{w}_i \tag{1}$$

$$\mathbf{y}_i = \mathbf{H}\mathbf{x}_i + \mathbf{v}_i \tag{2}$$

where $\mathbf{x}_i \in \mathbb{R}^N$ is a state vector to be estimated, $\mathbf{F} \in \mathbb{R}^{N\times N}$, $\mathbf{H} \in \mathbb{R}^{P\times N}$, $\mathbf{w}_i \in \mathbb{R}^N$, and $\mathbf{v}_i \in \mathbb{R}^P$ denote the model matrix, observation matrix, state noise vector, and the observation noise vector, respectively. For the dynamics described by (1) and (2), the following assumptions are considered:

Assumption 1:

(i) The measurement noise, $\mathbf{v}_i$, the state noise, $\mathbf{w}_i$, and the initial state of the system, $\mathbf{x}_0$, are Gaussian distributed sequences with $\mathbf{v}_i = \mathcal{N}(0, \mathbf{R})$, $\mathbf{n}_i = \mathcal{N}(0, \mathbf{Q})$, and $\mathbf{x}_0 = \mathcal{N}(0, \mathbf{P}_0)$. Note that the covariance matrices $\mathbf{R}$ and $\mathbf{Q}$ are both positive definite.

(ii) The measurement noise, the state noise, and the initial state, $\{\mathbf{v}_i, \mathbf{n}_i, \mathbf{x}_0\}$, are uncorrelated random vector sequences.

For future reference, the define the following quantities are defined

Definition 1: we denote by $\hat{\mathbf{x}}_{i|i-1}$ and $\hat{\mathbf{x}}_{i|i}$ the *priori* and *posteriori* minimum-variance linear estimate of $\mathbf{x}_i$, respectively. We also use $\mathbf{P}_{i|i}$ to show the estimation error covariance matrix after processing the measurement $\mathbf{y}_i$, and $\mathbf{P}_{i|i-1}$ as the extrapolated error covariance matrix.

To estimate $\mathbf{x}_i$, the standard form of Kalman filter can be applied, which is described by the following recursive equations:

$$\begin{aligned}
\hat{\mathbf{x}}_{i|i-1} &= \mathbf{F}\hat{\mathbf{x}}_{i-1|i-1} \\
\mathbf{P}_{i|i-1} &= \mathbf{F}\mathbf{P}_{i-1|i-1}\mathbf{F}^{\intercal} + \mathbf{Q} \\
\mathbf{K}_{\mathrm{p},i} &= \mathbf{P}_{i|i-1}\mathbf{H}^{\intercal}\left(\mathbf{H}\mathbf{P}_{i|i-1}\mathbf{H}^{\intercal} + \mathbf{R}\right)^{-1} \\
\hat{\mathbf{x}}_{i|i} &= \hat{\mathbf{x}}_{i|i-1} + \mathbf{K}_{\mathrm{p},i}\left(\mathbf{y}_i - \mathbf{H}\hat{\mathbf{x}}_{i|i-1}\right) \\
\mathbf{P}_{i|i} &= \mathbf{P}_{i|i-1} + \mathbf{K}_{\mathrm{p},i}\mathbf{H}\mathbf{P}_{i|i-1}
\end{aligned} \tag{3}$$

Note that the update equation for $\hat{\mathbf{x}}_{i|i}$ can be expressed als as

$$\hat{\mathbf{x}}_{i|i} = \left(\mathbf{I} - \mathbf{K}_{\mathrm{p},i}\mathbf{H}\right)\mathbf{F}\hat{\mathbf{x}}_{i-1|i-1} + \mathbf{K}_{\mathrm{p},i}\mathbf{y}_i \tag{4}$$

Remark 1: As it is discussed in [21], the estimator (4) is *uniform asymptotic* stable, provided that $(\mathbf{F}, \mathbf{H})$ is *observable* and $\left(\mathbf{F}, \mathbf{Q}^{\frac{1}{2}}\right)$ is *controllable*. Uniform asymptotic stability means that for $\theta_1 > 0$ and $\theta_2 > 0$, such that for all

$$\forall\ i \geq j \geq 0,\ \left\|\mathbf{\Phi}_{i|j}\right\| \theta_2 e^{-\theta_1 (i-j)} \tag{5}$$

where $\mathbf{\Phi}_{i|j} \triangleq \left(\mathbf{I} - \mathbf{K}_{\mathrm{p},i}\mathbf{H}\right)\mathbf{F}$ such that

$$\mathbf{\Phi}_{i|j} \triangleq \begin{cases} \mathbf{I}, & \text{if } i = j \\ \mathbf{\Phi}_{i|i-1}, \ldots, \mathbf{\Phi}_{i|j-1}, & \text{if } i > j \\ \left(\mathbf{\Phi}_{i|j}\right)^{-1}, & \text{otherwise} \end{cases} \tag{6}$$

B. Partial Update KF Algorithm (PUKF)

Our main objective is to decrease computational complexity of the KF algorithm associated with adaptation process by updating only M $(M < N)$ elements of the state estimate vectors at each iteration. Clearly, partial updating can be applied to each of the equations in (3), which in turn, result in different forms of KF with partial updating. Here, we apply the partial updating on the update equation of the state vector (i.e. equation (3)). Then, we modify (3) in the proposed algorithm as

$$\hat{\mathbf{x}}_{i|i} = \hat{\mathbf{x}}_{i|i-1} + \mathbf{\Pi}_i\mathbf{K}_{\mathrm{p},i}\left(\mathbf{y}_i - \mathbf{H}\hat{\mathbf{x}}_{i|i-1}\right) \tag{7}$$

where $\mathbf{\Pi}_i$ coefficient selection matrix which is a diagonal matrix defined by:

$$\mathbf{\Pi}_i \triangleq \begin{bmatrix} \tau_1(i) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \tau_N(i) \end{bmatrix},$$

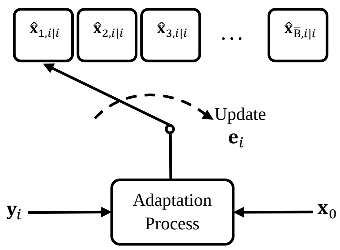

Fig. 1. Sequential partial updates using fractions of the state coefficient vector. $j = \bmod\left(i, \bar{B}\right)$ is associated with the partition to be updated at iteration i. Note that $\hat{\mathbf{x}}_{i|i} = \left[\hat{\mathbf{x}}_{1,i|i}, \hat{\mathbf{x}}_{2,i|i}, \ldots, \hat{\mathbf{x}}_{\bar{B},i|i}\right]$

where $\tau_j \in \{0, 1\}$, and $\tau_j = 1$ means that the jth element of $\hat{\mathbf{x}}_{i|i}$ is selected for update. In order to select M-subset of $\hat{\mathbf{x}}_{i|i}$, we consider two different schemes, including sequential partial update and stochastic partial update. In the sequential partial update (See Fig. 1), τ_j is given by

$$\tau_j\left(i\right) = \begin{cases} 1, & \text{if } j \in \mathcal{J}_{\bmod\left(i,\bar{B}\right)+1} \\ 0, & \text{otherwise} \end{cases}$$

with $\bar{B} = \lceil N/M \rceil$. The number of coefficients updated at each iteration is limited by M. The coefficient subsets $\mathcal{J}_i$ are not unique as long as they obey the following requirements:

- Cardinality of $\mathcal{J}_i$ is between 1 and M;
- $\bigcup_{k=1}^{\bar{B}} \mathcal{J}_k = \mathcal{S}$ where $\mathcal{S} = \{1, 2, \ldots, N\}$;
- $\mathcal{J}_k \cap \mathcal{J}_j = , \forall\ k, j \in \left\{1, \ldots, \bar{B}\right\}$ and $k \neq j$.

Note that if $\bar{B} = \lceil N/M \rceil$ (i.e. N/M is an integer) then the cardinality of each $\mathcal{J}_i$ must be M by necessity.

The following procedure yields a set of $\mathcal{J}_i$ compliant with the above requirements

$$\begin{aligned} \mathcal{J}_1 &= \{1, 2, \ldots, M\} \\ \mathcal{J}_2 &= \{M+1, M+2, \ldots, 2M\} \\ \mathcal{J}_{\bar{B}} &= \left\{\left(\bar{B}-1\right)M+1, \left(\bar{B}-1\right)M+2, \ldots, N\right\} \end{aligned}$$

where the cardinality of $\mathcal{J}_1, \mathcal{J}_2, \ldots, \mathcal{J}_{\bar{B}-1}$ is M and that of $\mathcal{J}_{\bar{B}}$ is $N - \left(\bar{B}-1\right)M \leq M$.

The second proposed algorithm, termed ~~the~~ stochastic partial update KF algorithm, is given by the same equation as described in (7). The only difference, however; between them is how to define the diagonal elements of the coefficient selection matrix $\mathbf{\Pi}_i$.

The diagonal elements of the $\mathbf{\Pi}_i$ are defined by the following scheme:

$$\tau_j\left(i\right) = \begin{cases} 1, & \text{if } j \in \mathcal{J}_{m(k)} \\ 0, & \text{otherwise} \end{cases}$$

Here the coefficient subsets $\mathcal{J}_i$ are characterized equitably to the sequential scheme, and $m\left(k\right)$ is an independent discrete random variable with uniform distribution:

$$\Pr\left\{m\left(k\right) = i\right\} = \frac{1}{\bar{B}},\ 1 \leq i \leq \bar{B}$$

Accordingly, the PUKF can be expressed as the Algorithm 1.

Algorithm 1 Partial Update Kalman Filter

Initialize $\hat{\mathbf{x}}_{0|0}$ **and** $\mathbf{P}_{0|0}$

for $i \geq 1$

Step 1: Time update (Predicted)

- Project the state ahead

$$\hat{\mathbf{x}}_{i|i-1} = \mathbf{F}\hat{\mathbf{x}}_{i-1|i-1}$$

- Project the error covariance ahead

$$\mathbf{P}_{i|i-1} = \mathbf{F}\mathbf{P}_{i-1|i-1}\mathbf{F}^{\mathsf{T}} + \mathbf{Q}$$

- Compute the Kalman gain

$$\mathbf{K}_{\mathrm{p},i} = \mathbf{P}_{i|i-1}\mathbf{H}^{\mathsf{T}}\left(\mathbf{H}\mathbf{P}_{i|i-1}\mathbf{H}^{\mathsf{T}} + \mathbf{R}\right)^{-1}$$

Step 2: Measurement update (Correct)

- Update estimate with measurement $\mathbf{y}_i$

$$\hat{\mathbf{x}}_{i|i} = \hat{\mathbf{x}}_{i|i-1} + \mathbf{\Pi}_i\mathbf{K}_{\mathrm{p},i}\left(\mathbf{y}_i - \mathbf{H}\hat{\mathbf{x}}_{i|i-1}\right)$$

- Update the error covariance

$$\mathbf{P}_{i|i} = \mathbf{P}_{i|i-1} + \mathbf{K}_{\mathrm{p},i}\mathbf{H}\mathbf{P}_{i|i-1}$$

End for

Remark 2: The $\tau_j(i)$ are some Bernoulli random variables, that the randomness varies over each iteration i. Each $\tau_j(i)$ is either zero or one with probability

$$\Pr\{\tau_j(i) = 0\} = \tau, \; 0 \leq \tau < 1$$

Remark 3: Table I illustrates the complexity of update equation in the proposed PUKF algorithm compared to KF algorithm, in terms of scalar multiplications and additions per iteration. Clearly, in models where N and P are very large (See real-World applications in [22]), the complexity of the proposed algorithm is much lower than the KF algorithm.

III. Performance Analysis

A. Mean Stability

Define the intermediate state estimate-error vector as

$$\widetilde{\mathbf{x}}_{i|i} \triangleq \mathbf{x}_i - \hat{\mathbf{x}}_i \tag{8}$$

Subtracting both sides (7) from $\mathbf{x}_i$ and using (2) give

$$\widetilde{\mathbf{x}}_{i|i} = \widetilde{\mathbf{x}}_{i|i-1} - \mathbf{\Pi}_i\mathbf{K}_{\mathrm{p},i}\mathbf{H}\widetilde{\mathbf{x}}_{i|i-1} - \mathbf{\Pi}_i\mathbf{K}_{\mathrm{p},i}\mathbf{v}_i \tag{9}$$

or more compactly as

$$\widetilde{\mathbf{x}}_{i|i} = \left(\mathbf{I} - \mathbf{\Pi}_i\mathbf{K}_{\mathrm{p},i}\mathbf{H}\right)\widetilde{\mathbf{x}}_{i|i-1} - \mathbf{\Pi}_i\mathbf{K}_{\mathrm{p},i}\mathbf{v}_i \tag{10}$$

Moreover, we have that

$$\widetilde{\mathbf{x}}_{i|i-1} = \mathbf{F}\widetilde{\mathbf{x}}_{i-1|i-1} + \mathbf{w}_{i-1} \tag{11}$$

TABLE I

REQUIRED COMPLEXITY FOR IMPLEMENTATION OF UPDATE EQUATION FOR STATE ~~ESTIMATE FOR~~ BOTH PUKF ~~ALGORITHM~~ AND KF ALGORITHMS.

Algorithm	Multiplications	Additions
KF	$2NP$	$2NP$
PUKF	$M(2N+1)$	$M(N+1)$

Substituting (11) into (10) the following recursion update equation is achieved

$$\widetilde{\mathbf{x}}_{i|i} = (\mathbf{I} - \mathbf{\Pi}_i \mathbf{K}_{\mathrm{p},i}\mathbf{H})\,\mathbf{F}\widetilde{\mathbf{x}}_{i-1|i-1} - \mathbf{\Pi}_i \mathbf{K}_{\mathrm{p},i}\mathbf{v}_i - (\mathbf{I} - \mathbf{\Pi}_i \mathbf{K}_{\mathrm{p},i}\mathbf{H})\,\mathbf{w}_{i-1} \tag{12}$$

Taking expectation from both sides of (12) we have

$$\mathsf{E}\big[\widetilde{\mathbf{x}}_{i|i}\big] = (\mathbf{I} - (1-\tau)\,\mathbf{K}_{\mathrm{p},i}\mathbf{H})\,\mathbf{F}\mathsf{E}\big[\widetilde{\mathbf{x}}_{i-1|i-1}\big] \tag{13}$$

From (13), we observe that in order for the algorithm to be stable in the mean sense, the matrix $(\mathbf{I} - (1-\tau)\,\mathbf{K}_{\mathrm{p},i}\mathbf{H})\,\mathbf{F}$ should be stable.

Assumption 2: The pair of matrices $(\mathbf{F}, \mathbf{H})$ is detectable and $\left(\mathbf{F}, \mathbf{H}^{\frac{1}{2}}\right)$ is controllable on the unique circle. Moreover, $\left(\mathbf{F}, \mathbf{H}^{\frac{1}{2}}\right)$ is stabilizable.

In light of Remark 1 and Assumption 2, the recursion of type (13) is stable in mean sense.

B. Mean Square Performance

For the system described by (1) and (2), partial update easily leads to the convergence of PUKF. Hence, ~~in this section~~, we discuss the stability of PUKF for system (1) and (2) under further assumption.

Taking mean square of (10) and (11) on both sides, we can get that

$$\begin{aligned}\widetilde{\mathbf{P}}_{i|i-1} &\triangleq \mathsf{E}\Big[\widetilde{\mathbf{x}}_{i|i-1}\left(\widetilde{\mathbf{x}}_{i|i-1}\right)^{\mathsf{T}}\Big] \\ &= \mathsf{E}\Big[\left(\mathbf{x}_i - \hat{\mathbf{x}}_{i|i-1}\right)\left(\mathbf{x}_i - \hat{\mathbf{x}}_{i|i-1}\right)^{\mathsf{T}}\Big] \\ &= \mathbf{F}\widetilde{\mathbf{P}}_{i-1|i-1}\mathbf{F}^{\mathsf{T}} + \mathbf{Q}\end{aligned} \tag{14}$$

$$\begin{aligned}\widetilde{\mathbf{P}}_{i|i} &\triangleq \mathsf{E}\Big[\widetilde{\mathbf{x}}_{i|i}\left(\widetilde{\mathbf{x}}_{i|i}\right)^{\mathsf{T}}\Big] \\ &= \mathsf{E}\Big[\left(\mathbf{x}_i - \hat{\mathbf{x}}_{i|i}\right)\left(\mathbf{x}_i - \hat{\mathbf{x}}_{i|i}\right)^{\mathsf{T}}\Big] \\ &= (\mathbf{I} - \mathsf{E}[\mathbf{\Pi}_i]\,\mathbf{K}_{\mathrm{p},i}\mathbf{H})\,\widetilde{\mathbf{P}}_{i|i-1}\,(\mathbf{I} - \mathsf{E}[\mathbf{\Pi}_i]\,\mathbf{K}_{\mathrm{p},i}\mathbf{H})^{\mathsf{T}} \\ &\quad + \mathsf{E}[\mathbf{\Pi}_i]\,\mathbf{K}_{\mathrm{p},i}\mathsf{E}[\mathbf{v}_i\mathbf{v}_i^{\mathsf{T}}]\,\mathbf{K}_{\mathrm{p},i}^{\mathsf{T}}\mathsf{E}[\mathbf{\Pi}_i] \\ &\quad - (\mathbf{I} - \mathsf{E}[\mathbf{\Pi}_i]\,\mathbf{K}_{\mathrm{p},i}\mathbf{H})\,\mathsf{E}[\mathbf{v}_i^{\mathsf{T}}]\,\mathbf{K}_{\mathrm{p},i}^{\mathsf{T}}\mathsf{E}[\mathbf{\Pi}_i] \\ &\quad - \mathsf{E}[\mathbf{\Pi}_i]\,\mathbf{K}_{\mathrm{p},i}\mathsf{E}[\mathbf{v}_i]\,(\mathbf{I} - \mathsf{E}[\mathbf{\Pi}_i]\,\mathbf{K}_{\mathrm{p},i}\mathbf{H})^{\mathsf{T}} \\ &= \left(\mathbf{I} - \bar{\mathbf{K}}_{\mathrm{p},i}\mathbf{H}\right)\widetilde{\mathbf{P}}_{i|i-1}\left(\mathbf{I} - \bar{\mathbf{K}}_{\mathrm{p},i}\mathbf{H}\right)^{\mathsf{T}} + \bar{\mathbf{K}}_{\mathrm{p},i}\mathbf{R}\bar{\mathbf{K}}_{\mathrm{p},i}^{\mathsf{T}}\end{aligned} \tag{15}$$

where $\bar{\mathbf{K}}_{\mathrm{p},i} \triangleq (1-\tau)\,\mathbf{K}_{\mathrm{p},i}$. The two equations above can be written more compactly as follows:

$$\widetilde{\mathbf{P}}_{i|i} = \left(\mathbf{I} - \bar{\mathbf{K}}_{\mathrm{p},i}\mathbf{H}\right)\left(\mathbf{F}\widetilde{\mathbf{P}}_{i-1|i-1}\mathbf{F}^{\mathsf{T}} + \mathbf{Q}\right)\left(\mathbf{I} - \bar{\mathbf{K}}_{\mathrm{p},i}\mathbf{H}\right)^{\mathsf{T}} + \bar{\mathbf{K}}_{\mathrm{p},i}\mathbf{R}\bar{\mathbf{K}}_{\mathrm{p},i}^{\mathsf{T}} \tag{16}$$

This leads to the following equation

$$\widetilde{\mathbf{P}}_{i|i} = \left(\boldsymbol{\mathcal{F}}\widetilde{\mathbf{P}}_{i-1|i-1}\boldsymbol{\mathcal{F}}^{\mathsf{T}} + \boldsymbol{\mathcal{Q}}\right) + \bar{\mathbf{K}}_{\mathrm{p},i}\mathbf{R}\bar{\mathbf{K}}_{\mathrm{p},i}^{\mathsf{T}} \tag{17}$$

where

$$\boldsymbol{\mathcal{F}} = \left(\mathbf{I} - \bar{\mathbf{K}}_{\mathrm{p},i}\mathbf{H}\right)\mathbf{F}$$
$$\boldsymbol{\mathcal{Q}} = \left(\mathbf{I} - \bar{\mathbf{K}}_{\mathrm{p},i}\mathbf{H}\right)\mathbf{Q}\left(\mathbf{I} - \bar{\mathbf{K}}_{\mathrm{p},i}\mathbf{H}\right)^{\mathsf{T}}$$

The steady-state MSD can be expressed by

$$\mathrm{MSD} = \lim_{i\to\infty} \mathsf{E}\left[\left\|\mathbf{x}_i - \tilde{\mathbf{x}}_{M,i|i}\right\|^2\right] = \mathsf{tr}\left(\widetilde{\mathbf{P}}_{i|i}\right) \tag{18}$$

Assumption 3: The pair is $(\mathbf{F}, \mathbf{H})$ detectable for every, i.e., there exists a matrix $\mathbf{L}$ such that $\mathbf{F} - \mathbf{L}\mathbf{H}$ is stable (all of its eigenvalues lie inside the unit circle). Moreover, $\left(\mathbf{F}, \mathbf{H}^{\frac{1}{2}}\right)$ is stabilizable, i.e., there exists a matrix $\mathbf{K}$ such that $\mathbf{F} - \mathbf{G}\mathbf{Q}^{\frac{1}{2}}\mathbf{K}$ is stable as well.

Remark 4: As it is discussed in [23], when Assumption 3 holds, matrix $\mathbf{F}$ is stable, which in turn guarantees the convergence of (17).

Lemma 1: Consider a recursion with the following for

$$\mathbf{X}_{i+1} = \mathbf{A}\mathbf{X}_i\mathbf{A}^{\mathsf{T}} + \mathbf{B}_i \tag{19}$$

where $\mathbf{A}$ is a stable matrix and $\mathbf{B}_i$ converges uniformly to $\mathbf{B}$ as $i \to \infty$. Then, $\mathbf{X}_i$ converges to $\mathbf{X}$ which is the solution of the following Lyapunov equation

$$\mathbf{X} = \mathbf{A}\mathbf{X}\mathbf{A}^{\mathsf{T}} + \mathbf{B} \tag{20}$$

Proof 1: For proof, see Theorem 5.1 in [23].

In light of Lemma 1, (17) converges to the unique solution of the following Lyapunov equation:

$$\widetilde{\mathbf{P}} = \boldsymbol{\mathcal{F}}\widetilde{\mathbf{P}}\boldsymbol{\mathcal{F}}^{\mathsf{T}} + \bar{\mathbf{K}}\mathbf{R}\bar{\mathbf{K}}^{\mathsf{T}} \tag{21}$$

IV. Numerical Simulation

~~In this section~~, the performance of the proposed algorithm and ~~the accuracy of~~ the theoretical stead-state analysis are evaluated. To this end, we use the proposed algorithm to perform the state-estimation task in a centralized sensor network with $J = 20$ sensors. In this simulation scenario, each sensor transmits its noisy observations of a high-dimensional state $\mathbf{x}_i$ with $N = 80$ to a fusion center (FC), where the estimation task is performed. The noisy measurement of each senor n (i.e. $\mathbf{y}_{n,i}$) is given by

$$\mathbf{y}_{n,i} = \mathbf{H}_n\mathbf{x}_i + \mathbf{v}_{n,i}$$

where

$$\mathbf{H}_n = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \tag{22}$$

TABLE II

STEADY-STATE PERFORMANCE OF DIFFERENT ALGORITHMS.

Proposed (stochastic) $M = 20$	Proposed (stochastic) $M = 57$	Full-update KF	Method [24]
-7.52	-9.1	-9.74	-8.8

with $\mathbf{v}_{n,i}$ is a zero mean Gaussian process with $\mathbf{R} = 0.001\mathbf{I}_M$. The initial state is $\mathbf{x}_{0|0} = 0.5 \times [1 \cdots 1]^\intercal$. The collected noisy data at the FC is given by (1) where

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_0 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & F_0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & F_0 \end{bmatrix}, \quad \mathbf{H}_i = \begin{bmatrix} \mathbf{H}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H}_J \end{bmatrix},$$

$$\mathbf{F}_0 = \begin{bmatrix} 1 & 0 & 0.61 & 0 \\ 0 & 1 & 0 & 0.61 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{v}_i = \begin{bmatrix} \mathbf{v}_{1,i} \\ \mathbf{v}_{2,i} \\ \vdots \\ \mathbf{v}_{J,i} \end{bmatrix}, \quad \mathbf{y}_i = \begin{bmatrix} \mathbf{y}_{1,i} \\ \mathbf{y}_{2,i} \\ \vdots \\ \mathbf{y}_{J,i} \end{bmatrix}$$

The state noise $\mathbf{w}_i$ is zero mean Gaussian process with $\mathbf{Q} = 0.001\mathbf{I}_N$.

Figures 2 and 3 show the evolution of MSD for PUKF algorithm for different values of M. Both stochastic and sequential partial updating schemes are considered. The simulated MSD are obtained by averaging over $r = 1000$ Monte Carlo runs.

$$\mathrm{MSD}_i^{\mathrm{Sim}} = \frac{1}{r} \sum_{\ell=1}^{r} \left(\mathbf{x}_i - \hat{\mathbf{x}}_{i|i}\right) \left(\mathbf{x}_i - \hat{\mathbf{x}}_{i|i}\right)^\intercal \tag{23}$$

Note that the theoretical MSD values are calculated using (18). These figures show that the PUKF algorithm with both sequential and stochastic schemes can deliver acceptable estimate accuracy. Moreover, the MSD curves tend to the theoretical steady-state value, which validates the provided steady-state analysis. Fig. 4 shows the evolution of first elements in $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$ for different values of M. We can observe that the PUKF algorithm provides a trade-off between the estimation accuracy and the computational complexity, i.e. as M increases, better estimates for state vector $\mathbf{x}_i$ are obtained. From this figure, we can also observe it is possible to select a small value for M and obtain an acceptable estimation accuracy. Table. II shows the steady-state performance of the proposed algorithm in comparison with the full-update KF and the given algorithm in [24]. As we can see, there is a performance deterioration in comparison to the full-update KF, as expected. In addition, it provides an acceptable performance, compared to the given method in [24] which relies on the combination of combinations of the recursive least squares (RLS) and least mean square (LMS) algorithms, which equire more computaions than our proposed algorithm.

V. CONCLUSION AND FUTURE WORK

In this letter we derived the partial update Kalman filtering algorithm for state estimation. We considered sequential and stochastic partial updating schemes in the proposed algorithm. The condition for mean and mean-square convergence discussed and provided steady-state analysis has been evaluated by simulation results. Theoretical analysis and numerical simulations provided valuable insights into the performance of the PUKF algorithm. As we have shown in the simulation results, although partial updating is an effective way to address scarce hardware resources (reduce algorithm complexity), however, it can lead to performance deterioration is expected, since only a subset of all coefficients is updated.

In our future work, we will consider the problem of developing reduced-complexity Kalman filtering algorithm for more realistic case such as state estimation of linear and nonlinear stochastic systems in presence of faults and nonGaussian noises [25]–[27].

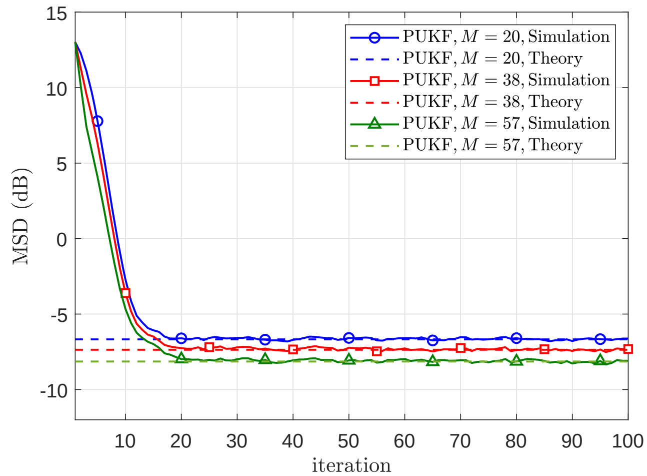

Fig. 2. The evolution of experimental MSD of sequential PUKF for different values of M.

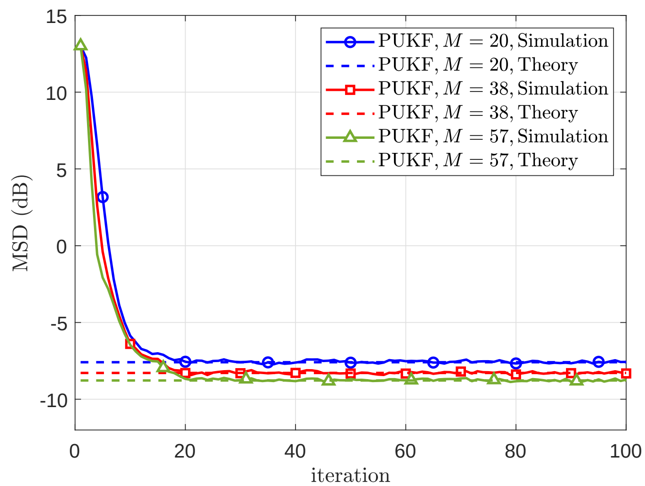

Fig. 3. The evolution of experimental MSD of stochastic PUKF for different values of M.

Data Availability Statement

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

References

[1] X. Sun and M. Medvedovic, "Model reduction and parameter estimation of non-linear dynamical biochemical reaction networks," *IET Systems Biology*, vol. 10, no. 1, pp. 10–16, 2016.

[2] J. B. Jorgensen, P. G. Thomsen, H. Madsen, and M. R. Kristensen, "A computationally efficient and robust implementation of the continuous-discrete extended kalman filter," in *2007 American Control Conference*, 2007, pp. 3706–3712.

[3] M. Raitoharju and R. Pich, "On computational complexity reduction methods for kalman filter extensions," *IEEE Aerospace and Electronic Systems Magazine*, vol. 34, no. 10, pp. 2–19, 2019.

[4] M. Roth, G. Hendeby, C. Fritsche, and F. Gustafsson, "The ensemble kalman filter: a signal processing perspective," *EURASIP J. Adv. Signal Process*, vol. 56, 2017.

[5] B. Xie and T. Bose, "Partial update least-square adaptive filtering," *Synthesis Lectures on Communications*, vol. 7, no. 1, pp. 1–115, 2014.

[6] S. C. Douglas, "Adaptive filters employing partial updates," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, no. 3, pp. 209–216, 1997.

[7] M. Godavarti and A. O. Hero, "Partial update lms algorithms," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2382–2399, 2005.

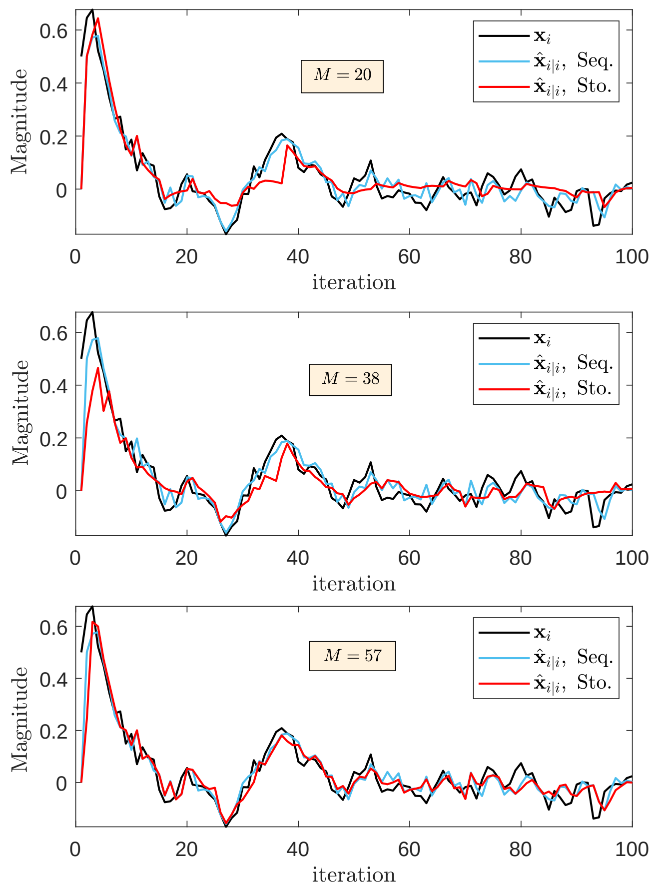

Fig. 4. The evolution of experimental MSD of stochastic PUKF for different values of M.

[8] C. Li, S. Chu, and S. Tsai, "Improved variable step-size partial update lms algorithm," in *2016 3rd International Conference on Green Technology and Sustainable Development (GTSD)*, 2016, pp. 17–19.

[9] Minglu Jin, "Partial updating rls algorithm," in *Proceedings 7th International Conference on Signal Processing, 2004. Proceedings. ICSP '04. 2004.*, vol. 1, 2004, pp. 392–395 vol.1.

[10] S. Werner, M. L. De Campos, and P. S. Diniz, "Partial-update NLMS algorithms with data-selective updating," *IEEE Transactions on Signal Processing*, vol. 52, no. 4, pp. 938–949, 2004.

[11] V. Vahidpour, A. Rastegarnia, A. Khalili, W. M. Bazzi, and S. Sanei, "Variants of partial update augmented clms algorithm and their performance analysis," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3146–3157, 2020.

[12] K. Mayyas and L. Afeef, "A variable step-size partial-update normalized least mean square algorithm for second-order adaptive volterra filters," *Circuits Syst Signal Processing*, vol. 39, pp. 6073–6097, 2020.

[13] R. Arablouei, S. Werner, and K. Dogancay, "Partial-diffusion recursive least-squares estimation over adaptive networks," in *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013 IEEE 5th International Workshop on*. IEEE, 2013, pp. 89–92.

[14] R. Arablouei, S. Werner, Y.-F. Huang, and K. Dogancay, "Distributed least mean-square estimation with partial diffusion," *Signal Processing, IEEE Transactions on*, vol. 62, no. 2, pp. 472–484, 2014.

[15] V. Vahidpour, A. Rastegarnia, A. Khalili, and S. Sanei, "Analysis of partial diffusion recursive least squares adaptation over noisy links," *IET Signal Processing*, vol. 11, no. 6, pp. 749–757, 2017.

[16] ——, "Partial diffusion Kalman filtering for distributed state estimation in multiagent networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[17] V. Vahidpour, A. Rastegarnia, A. Khalili, W. M. Bazzi, and S. Sanei, "Analysis of partial diffusion lms for adaptive estimation over networks with noisy links," *IEEE Transactions on Network Science and Engineering*, vol. 5, no. 2, pp. 101–112, 2017.

[18] V. Vahidpour, A. Rastegarnia, M. Latifi, A. Khalili, and S. Sanei, "Performance analysis of distributed Kalman filtering with partial diffusion over noisy network," *IEEE Transactions on Aerospace and Electronic Systems*, 2019.

[19] A. Khalili, V. Vahidpour, A. Rastegarnia, W. M. Bazzi, and S. Sanei, "Partial diffusion kalman filter with adaptive combiners," *IEEE Transactions on Aerospace and Electronic Systems*, 2020.

[20] K. Dogancay, *Partial-update adaptive signal processing: Design Analysis and Implementation*. Academic Press, 2008.

[21] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear estimation*. Prentice Hall, 2000, no. BOOK.

[22] M. Katzfuss, J. R. Stroud, and C. K. Wikle, “Ensemble kalman methods for high-dimensional hierarchical dynamic space-time models,” *Journal of the American Statistical Association*, vol. 115, no. 530, pp. 866–885, 2020.

[23] F. S. C. Dobarro, *Distributed collaborative processing over adaptive networks*. University of California, Los Angeles, 2010.

[24] R. Claser and V. H. Nascimento, “Low-complexity approximation to the kalman filter using convex combinations of adaptive filters from different families,” in *2017 25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 2630–2633.

[25] V. Stojanovic, S. He, and B. Zhang, “State and parameter joint estimation of linear stochastic systems in presence of faults and non-gaussian noises,” *International Journal of Robust and Nonlinear Control*, vol. 30, no. 16, pp. 6683–6700, 2020.

[26] V. Stojanovic and D. Prsic, “Robust identification for fault detection in the presence of non-gaussian noises: application to hydraulic servo drives,” *Nonlinear Dynamics*, vol. 100, p. 22992313, 2020.

[27] H. Fang, G. Zhu, V. Stojanovic, R. Nie, S. He, X. Luan, and F. Liu, “Adaptive optimization algorithm for nonlinear markov jump systems with partial unknown dynamics,” *International Journal of Robust and Nonlinear Control*, vol. 31, no. 6, pp. 2126–2140, 2021.