Remote State Estimation with Stochastic Event-triggered Sensor Schedule and Packet Drops

Liang Xu, Yilin Mo and Lihua Xie

***Abstract*—This paper studies the remote state estimation problem of linear time-invariant systems with stochastic event-triggered sensor schedules in the presence of packet drops between the sensor and the estimator. Due to the existence of packet drops, the Gaussianity at the estimator side no longer holds. It is proved that the system state conditioned on the available information at the estimator side is Gaussian mixture distributed. The minimum mean square error (MMSE) estimator can be obtained from a bank of Kalman filters. Since the optimal estimators require exponentially increasing computation and memory with time, sub-optimal estimators to reduce computational complexities by limiting the length and numbers of hypotheses are further provided. In the end, simulations are conducted to illustrate the performance of the optimal and sub-optimal estimators.**

I. Introduction

Sensor networks have wide applications in environment and habitat monitoring, industrial automation, smart buildings, etc. In many applications, sensors are battery powered and are required to reduce the energy consumption to prolong their service life. Sensor scheduling algorithms are therefore proposed as an efficient method by scheduled transmissions to reduce the communication frequency to prolong the service time of sensor devices.

Sensor scheduling algorithms can be roughly categorized as off-line schedules and event-triggered schedules. The off-line schedules are designed based on the communication frequency requirement and the statistics of the systems [1–3]. Compared with off-line schedules, event-triggered schedules depend on both the statistics and the realization of the system, which are expected to achieve better performance than off-line ones. Many triggering rules have been proposed in the literature based on the condition that, the estimation error [4], error in predicated output [5], functions of the estimation error [6, 7], or the error covariance [8], exceeds a given threshold. For example, a measurement innovation based event-triggered sensor scheduling scheme is proposed to reduce the communication rate in the remote state estimation problem in [6]. The novelty is to use the hold of transmission event to deliver information about the sensor measurement. However, since the innovation is not Gaussian, only sub-optimal estimators can be obtained. Stochastic event-triggered sensor scheduling algorithms are further proposed in [7] to handle the non-Gaussian problem. Both open-loop and closed-loop schedules are proposed, and it is shown that the conditional distributions of the system state are Gaussian. As a result, closed-form minimum mean square error (MMSE) estimators are obtained. Similar non-Gaussianity phenomenon could appear when transmit power control is used in sensor networks [9]. To overcome the problem, a transmit power controller based on a specific quadratic form of measurement innovations is carefully designed in [9] to preserve the Gaussianity of a posterior state distribution, which facilities the MMSE estimator design and performance analysis.

Wireless communications are mostly utilized in sensor networks, and packet drops are inevitable in wireless communications. Therefore, it is necessary to study how packet drops affect sensor scheduling algorithms [3, 10]. It should be noted that, for off-line schedulers and estimation error covariance based event-triggered schedulers, there is no need to distinguish between the channel loss event and the hold of transmission event when designing estimators. As long as the estimator receives the packet, it can conduct the measurement update to improve the estimate and vice versa. However, the case is different for the stochastic event-triggered sensor scheduling algorithms in [7] where the sensor measurement is used as the trigger criterion and the hold of transmission event contains information about the sensor measurement. In the presence of possible channel losses, the estimator cannot decide whether the non-reception of the packet can be attributed to the sensor measurement or the channel loss. If it is due to that the sensor measurement lies below the given threshold, then this information can be leveraged to improve the estimate. However, if it is caused by the channel loss, the estimator will have no information about the sensor measurement and no update will be carried out. This fact complicates the optimal estimator design. Furthermore, it is proved that, in the presence of channel losses, the Gaussian properties with the stochastic event-triggered sensor scheduling algorithms in [7] no longer hold [11].

This paper considers the same problem setting as in [7] with the additional consideration of the presence of packet drops between the sensor and the estimator. We try to derive the MMSE estimator in the case that the estimator has no knowledge about the channel loss events and only knows the channel loss rate. The main contributions are as follows. Firstly, we show that the conditional distributions of the system state at the estimator side are mixture Gaussian. Secondly, recursive MMSE estimators are derived. Thirdly, sub-optimal estimation algorithms to reduce computational complexities are provided.

The optimal estimator design has been presented in [12]. This paper contains new contents about sub-optimal estimator designs with fixed computation requirements. This paper is organized as follows. The problem formulation is given in Section II. The optimal estimator is studied in Section III. Tow sub-optimal estimators with fixed computation requirements are proposed in Section IV, respectively. Simulation comparisons are provided in Section V. This paper ends with some concluding remarks in Section VI.

Notation: All matrices and vectors are assumed to be of appropriate dimensions that are clear from the context. $\mathbb{R}, \mathbb{R}^n, \mathbb{R}^{m\times n}$ represent the sets of real scalars, n-dimensional real column vectors and $m\times n$-dimensional real matrices, respectively. $\mathbb{N}$ denotes the set of natural numbers. $\mathcal{N}_x(\bar{x}, \Sigma)$ denotes the Gaussian probability density function of the random variable x with the mean $\bar{x}$ and the covariance matrix Σ. $f(x)$ ($\Pr(x)$) denotes the probability density function (probability) of the random variable x. $f(x|y)$ ($\Pr(x|y)$) denotes the probability density function (probability) of the random variable x conditioned on the event that $Y = y$. $\mathbb{E}\{\cdot\}$ denotes the expectation operator. A',

Liang Xu is with the Institute of Mechanical Engineering, École polytechnique fédérale de Lausanne (EPFL), Switzerland (e-mail: liang.xu@epfl.ch)

Lihua Xie is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore (e-mail: elhxie@ntu.edu.sg)

Yilin Mo is with the Department of Automation, BNRist, Tsinghua University, Beijing, China (e-mail: ylmo@tsinghua.edu.cn)

This work is supported in part by the National Key Research and Development Program of China under Grant 2018AAA0101601, and the A*STAR Industrial Internet of Things Research Program of Singapore under the RIE2020 IAF-PP Grant A1788a0023.

A^{-1} and $|A|$ are the transpose, the inverse and the determinant of matrix A, respectively. $A > 0$ means that the matrix A is positive-definite. The term $x'Ax$ for symmetric matrix A and vector x is abbreviated as $x'A(*)$.

II. Problem Formulation

In this paper, we are interested in the remote state estimation problem with stochastic event-triggered sensor schedule in the presence of packet drops depicted in Fig. 1. The linear process is

$$\begin{aligned} x_{k+1} &= Ax_k + w_k, \\ y_k &= Cx_k + v_k, \end{aligned}$$

where $x_k \in \mathbb{R}^n$ is the process state; $y_k \in \mathbb{R}^m$ is the measurement output; w_k and v_k are the process and measurement noises. We assume that $\{w_k\}_{k\geq 0}$ and $\{v_k\}_{k\geq 0}$ are white Gaussian processes with covariance matrices Q and R, respectively. Moreover, the initial system state satisfies $x_0 \sim \mathcal{N}_{x_0}(0, \Sigma_0)$ and is independent with w_k and v_k.

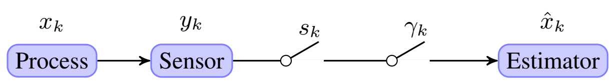

Fig. 1. Remote state estimation with stochastic event-triggered sensor schedule in the presence of packet drops

After receiving y_k, the sensor follows the stochastic event-triggered schedule [7] to decide whether to transmit y_k to the estimator or not. Let s_k denote the decision variable by the sensor. When $s_k = 1$, the sensor transmits y_k to the estimator and $s_k = 0$, otherwise. The stochastic event-triggered sensor schedule [7] operates as follows. At time k, the sensor randomly generates a variable ζ_k from the uniform distribution on $[0, 1]$. Then ζ_k is compared with $e^{-y_k' Y y_k}$, where $Y > 0$. The sensor schedules transmissions based on the following rule

$$s_k = \begin{cases} 0, & \text{if } \zeta_k \leq e^{-y_k' Y y_k}, \\ 1, & \text{if } \zeta_k > e^{-y_k' Y y_k}. \end{cases} \tag{1}$$

It is shown in [7] that the design (1) together with the random variable ζ_k can avoid the nonlinearity introduced by the truncated Gaussian prior conditional distribution of the system state in deterministic event-triggered schedule in [6].

Remark 1. *There are several reasons for not implementing the Kalman filter at the sensor side. The first reason is that the sensor might be primitive [7], so it does not have a sufficient computation capability to run a local Kalman filter. Secondly, the system parameters might not be available to the sensor. Thirdly, in decentralized settings where there are multiple sensors measuring the same process, only the fusion center which has access to all the sensor measurements can run the Kalman filter. In the end, the state dimension might be larger than the output dimension. Therefore, it reduces the communication cost to transmit the sensor output and perform the Kalman filter at the estimator side.*

Remark 2. *In this paper, we only consider the open-loop stochastic event-triggered sensor schedule. The results for the closed-loop stochastic event-triggered sensor schedule [7] can be obtained in a similar way.*

The communication channel between the sensor and the estimator suffers from i.i.d. packet drops, which are described by the i.i.d. stochastic process $\{\gamma_k\}_{k\geq 0}$. When $\gamma_k = 1$, the transmission is successful, and $\gamma_k = 0$, otherwise. Moreover, we assume $\gamma_k \in \{0, 1\}$ has a Bernoulli distribution with $\Pr(\gamma_k = 0) = p$. Therefore, the following information is available to the estimator at time k

$$\mathcal{I}_k = \{s_0\gamma_0, \ldots, s_k\gamma_k, s_0\gamma_0 y_0, \ldots, s_k\gamma_k y_k\} \tag{2}$$

with $\mathcal{I}_{-1} = \emptyset$.

When there are no transmission packet drops, the posterior distribution $f(x_k|\mathcal{I}_k)$ is shown to be Gaussian in [7]. However, in the presence of packet drops, the Gaussian property no longer holds [11]. In subsequent sections, we will show that in the presence of packet drops, the posterior distribution $f(x_k|\mathcal{I}_k)$ is mixture Gaussian with exponentially increasing components with time. Moreover, the MMSE estimator is derived.

III. Optimal Estimator

In this section, we try to derive the MMSE estimator in the presence of packet drops between the sensor and the estimator. First of all, the following notions are defined. For any given $k \in \mathbb{N}$ and $i \in \mathbb{N}$ with $0 \leq i \leq 2^{k+1} - 1$, define the event

$$\Gamma_k^i = \{\gamma_k = b_k, \ldots, \gamma_0 = b_0\},$$

where b_k is the $(k+1)$-th element of the binary expansion of i, i.e., $i = b_k 2^k + b_{k-1}2^{k-1} + \ldots + b_0 2^0$. Therefore, Γ_k^i denotes a packet drop sequence $\{\gamma_k, \ldots, \gamma_0\}$ specified by the index i. For any given $i \in \mathbb{N}$ and $k \in \mathbb{N}$, let

$$i_k^- = \begin{cases} i, & \text{if } i < 2^k, \\ i - 2^k, & \text{if } i \geq 2^k. \end{cases}$$

Therefore, i_k^- is the index of the sub-sequence $\{\gamma_{k-1}, \ldots, \gamma_0\}$ extracted from the sequence $\{\gamma_k, \ldots, \gamma_0\}$ specified by the index i. From the law of total probability, we have that

$$f(x_k|\mathcal{I}_k) = \sum_{i=0}^{2^{k+1}-1} f(x_k|\Gamma_k^i, \mathcal{I}_k)\Pr(\Gamma_k^i|\mathcal{I}_k), \tag{3}$$

$$f(x_k|\mathcal{I}_{k-1}) = \sum_{i=0}^{2^k-1} f(x_k|\Gamma_{k-1}^i, \mathcal{I}_{k-1})\Pr(\Gamma_{k-1}^i|\mathcal{I}_{k-1}). \tag{4}$$

We call Γ_k^i the hypothesis, and $\Pr(\Gamma_k^i|\mathcal{I}_k)$ the hypothesis probability. The conditional distribution $f(x_k|\Gamma_k^i, \mathcal{I}_k)$ can be shown to be Gaussian. Therefore, $f(x_k|\mathcal{I}_k)$ is mixture Gaussian.

Remark 3. *The Gaussian mixture phenomenon also appears in other control and estimation problems. For example, the estimation of Markov jump linear systems with unknown jump mode [13] and the estimation of linear systems with unknown control inputs [14]. In general, if there are hidden variables in the control and estimation problem, the resulting conditional distribution is a mixture distribution from the law of total probability.*

It should be noted that in (3), there are 2^{k+1} hypotheses at time k. However, certain hypotheses might be impossible. For example, when $\gamma_k s_k = 1$, we know that $\gamma_k = 1$. Therefore, the hypotheses with $\gamma_k = 0$ should be excluded. Nevertheless, for such case, we can show that $\Pr(\Gamma_k^i|\mathcal{I}_k) = 0$. In principle, the set of hypotheses at time k is determined by the received signals $\{\gamma_k s_k, \ldots, \gamma_0 s_0\}$, which however is stochastic and cannot be determined in a prior. Therefore, in the sequel, we always express $f(x_k|\mathcal{I}_k)$ as a summation of 2^{k+1} hypotheses as in (3) to simplify the description.

In the sequel, we will recursively derive $f(x_k|\Gamma_k^i, \mathcal{I}_k)$ and $\Pr(\Gamma_k^i|\mathcal{I}_k)$. Then in view of (3) and (4), the MMSE estimator can be obtained. The following results are required and presented first.

$$f(x_k|\Gamma_k^i, \mathcal{I}_{k-1}) = f(x_k|\Gamma_{k-1}^{i_k^-}, \mathcal{I}_{k-1}), \tag{5}$$

since $\{\gamma_k\}$ is i.i.d. and knowing γ_k only cannot help to improve the knowledge about x_k. We then have the following result.

Lemma 4.

$$f(x_k|\Gamma_k^i, \mathcal{I}_k) = \mathcal{N}_{x_k}(\hat{x}_{k|k}^i, P_{k|k}^i), \quad 0 \le i \le 2^{k+1}-1$$
$$f(x_k|\Gamma_{k-1}^i, \mathcal{I}_{k-1}) = \mathcal{N}_{x_k}(\hat{x}_{k|k-1}^i, P_{k|k-1}^i), \quad 0 \le i \le 2^{k}-1$$

where $\hat{x}_{k|k}^i, P_{k|k}^i, \hat{x}_{k|k-1}^i, P_{k|k-1}^i$ satisfy the following recursion.

Time update:

$$\hat{x}_{k|k-1}^i = A\hat{x}_{k-1|k-1}^i,$$
$$P_{k|k-1}^i = AP_{k-1|k-1}^i A' + Q.$$

Measurement update:

- *For $i < 2^k$,*

$$\hat{x}_{k|k}^i = \hat{x}_{k|k-1}^i, P_{k|k}^i = P_{k|k-1}^i. \tag{6}$$

- *For $i \ge 2^k$,*

$$\hat{x}_{k|k}^i = (I - K_k^{i_k^-} C)\hat{x}_{k|k-1}^{i_k^-} + K_k^{i_k^-} s_k\gamma_k y_k, \tag{7}$$

$$P_{k|k}^i = P_{k|k-1}^{i_k^-} - K_k^{i_k^-} CP_{k|k-1}^{i_k^-}, \tag{8}$$

$$K_k^{i_k^-} = P_{k|k-1}^{i_k^-} C'[CP_{k|k-1}^{i_k^-}C' + R + (1 - s_k\gamma_k)Y^{-1}]^{-1} \tag{9}$$

with initial conditions

$$\hat{x}_{0|-1}^0 = 0, \quad P_{0|-1}^0 = \Sigma_0.$$

Proof. The proofs of the initialization and the time update are straightforward. The measurement update (6) follows from the fact that for $i < 2^k$, we have $\gamma_k = 0$. Therefore, no new information is available and the measurement update is not needed.

The measurement update (7), (8) and (9) follow from the fact that for $i \ge 2^k$, we have $\gamma_k = 1$. Therefore, the measurement update is the same as [7]. □

Next we calculate the probabilities of $\alpha_{k|k-1}^i = \Pr(\Gamma_k^i|\mathcal{I}_{k-1})$, $\alpha_{k|k}^i = \Pr(\Gamma_k^i|\mathcal{I}_k)$ and have the following result.

Lemma 5. *$\alpha_{k|k-1}^i$ and $\alpha_{k|k}^i$ with $0 \le i \le 2^{k+1}-1$ satisfy the following recursion.*

Time update:

- *For $i < 2^k$,*

$$\alpha_{k|k-1}^i = p\alpha_{k-1|k-1}^i. \tag{10}$$

- *For $i \ge 2^k$,*

$$\alpha_{k|k-1}^i = (1-p)\alpha_{k-1|k-1}^{i_k^-}. \tag{11}$$

Measurement update:

$$\alpha_{k|k}^i = \frac{\Pr(s_k\gamma_k|\Gamma_k^i, \mathcal{I}_{k-1})\alpha_{k|k-1}^i}{\sum_{j=0}^{2^{k+1}-1}\Pr(s_k\gamma_k|\Gamma_k^j, \mathcal{I}_{k-1})\alpha_{k|k-1}^j},$$

where

- *for $j < 2^k$,*

$$\Pr(s_k\gamma_k|\Gamma_k^j, \mathcal{I}_{k-1}) = 1 - s_k\gamma_k.$$

- *for $j \ge 2^k$,*

$$\Pr(s_k\gamma_k|\Gamma_k^j, \mathcal{I}_{k-1}) = s_k\gamma_k + \frac{1 - 2s_k\gamma_k}{\sqrt{|(CP_{k|k-1}^{j_k^-}C' + R)Y + I|}}$$
$$\times e^{-\frac{1}{2}(C\hat{x}_{k|k-1}^{j_k^-})'[Y^{-1} + (CP_{k|k-1}^{j_k^-}C'+R)]^{-1}C\hat{x}_{k|k-1}^{j_k^-}}$$

with the initial condition

$$\alpha_{0|-1}^0 = p, \quad \alpha_{0|-1}^1 = 1-p.$$

Proof. Time update: (10) follows from the fact that for $i < 2^k$, we have $\gamma_k = 0$. Therefore

$$\begin{aligned}
\alpha_{k|k-1}^i &= \Pr(\Gamma_{k-1}^i, \gamma_k = 0|\mathcal{I}_{k-1})\\
&= \Pr(\gamma_k = 0)\Pr(\Gamma_{k-1}^i|\mathcal{I}_{k-1})\\
&= p\alpha_{k-1|k-1}^i.
\end{aligned}$$

On the other hand, (11) follows from the fact that for $i \ge 2^k$, we have $\gamma_k = 1$. Therefore

$$\begin{aligned}
\alpha_{k|k-1}^i &= \Pr(\Gamma_{k-1}^{i_k^-}, \gamma_k = 1|\mathcal{I}_{k-1})\\
&= \Pr(\gamma_k = 1)\Pr(\Gamma_{k-1}^{i_k^-}|\mathcal{I}_{k-1})\\
&= (1-p)\alpha_{k-1|k-1}^{i_k^-},
\end{aligned}$$

which is (11).

Measurement update: Since

$$\begin{aligned}
\alpha_{k|k}^i &= \Pr(\Gamma_k^i|\mathcal{I}_k)\\
&= \Pr(\Gamma_k^i|s_k\gamma_k, s_k\gamma_k y_k, \mathcal{I}_{k-1})\\
&= \Pr(\Gamma_k^i|s_k\gamma_k, \mathcal{I}_{k-1})\\
&= \frac{\Pr(s_k\gamma_k|\Gamma_k^i, \mathcal{I}_{k-1})\Pr(\Gamma_k^i|\mathcal{I}_{k-1})}{\Pr(s_k\gamma_k|\mathcal{I}_{k-1})}\\
&= \frac{\Pr(s_k\gamma_k|\Gamma_k^i, \mathcal{I}_{k-1})\Pr(\Gamma_k^i|\mathcal{I}_{k-1})}{\sum_{j=0}^{2^{k+1}-1}\Pr(s_k\gamma_k|\Gamma_k^j, \mathcal{I}_{k-1})\Pr(\Gamma_k^j|\mathcal{I}_{k-1})}\\
&= \frac{\Pr(s_k\gamma_k|\Gamma_k^i, \mathcal{I}_{k-1})\alpha_{k|k-1}^i}{\sum_{j=0}^{2^{k+1}-1}\Pr(s_k\gamma_k|\Gamma_k^j, \mathcal{I}_{k-1})\alpha_{k|k-1}^j},
\end{aligned}$$

where the third equality follows from the fact that when $s_k\gamma_k = 0$, $s_k\gamma_k y_k = 0$, it is useless to know $s_k\gamma_k y_k$; when $s_k\gamma_k = 1$, knowing $s_k\gamma_k y_k$ is equivalent to know y_k, which is also useless in predicting Γ_k^i. Next we will show how to calculate $\Pr(s_k\gamma_k|\Gamma_k^i, \mathcal{I}_{k-1})$.

When $i < 2^k$, since $\gamma_k = 0$, we have that $s_k\gamma_k \equiv 0$. Therefore

$$\Pr(s_k\gamma_k|\Gamma_k^i, \mathcal{I}_{k-1}) = 1 - s_k\gamma_k. \tag{12}$$

When $i \ge 2^k$, we have $\gamma_k = 1$. Let $M_k^{i_k^-} = CP_{k|k-1}^{i_k^-}C' + R$, we then have

$$\begin{aligned}
&\Pr(s_k\gamma_k|\Gamma_k^i, \mathcal{I}_{k-1}) = \Pr(s_k|\Gamma_k^i, \mathcal{I}_{k-1})\\
&= \int_{\mathbb{R}^m}\Pr(s_k|y_k, \Gamma_k^i, \mathcal{I}_{k-1})f(y_k|\Gamma_k^i, \mathcal{I}_{k-1})\mathrm{d}y_k\\
&= \int_{\mathbb{R}^m}\Pr(s_k|y_k)f(Cx_k + v_k|\Gamma_k^i, \mathcal{I}_{k-1})\mathrm{d}y_k\\
&\overset{(a)}{=} \int_{\mathbb{R}^m}\left(s_k(1 - 2e^{-\frac{1}{2}y_k'Yy_k}) + e^{-\frac{1}{2}y_k'Yy_k}\right)\\
&\times f(Cx_k + v_k|\Gamma_{k-1}^{i_k^-}, \mathcal{I}_{k-1})\mathrm{d}y_k\\
&= \int_{\mathbb{R}^m}\left(s_k(1 - 2e^{-\frac{1}{2}y_k'Yy_k}) + e^{-\frac{1}{2}y_k'Yy_k}\right)\\
&\times \mathcal{N}_{y_k}(C\hat{x}_{k|k-1}^{i_k^-}, M_k^{i_k^-})\mathrm{d}y_k\\
&= s_k + (1 - 2s_k)\int_{\mathbb{R}^m}e^{-\frac{1}{2}y_k'Yy_k}\mathcal{N}_{y_k}(C\hat{x}_{k|k-1}^{i_k^-}, M_k^{i_k^-})\mathrm{d}y_k\\
&= s_k + \frac{1 - 2s_k}{\sqrt{(2\pi)^m|M_k^{i_k^-}|}}\\
&\times \int_{\mathbb{R}^m}e^{-\frac{1}{2}y_k'Yy_k - \frac{1}{2}(y_k - C\hat{x}_{k|k-1}^{i_k^-})'(M_k^{i_k^-})^{-1}(*)}\mathrm{d}y_k
\end{aligned}$$

$$
\begin{aligned}
&= s_k + \frac{1-2s_k}{\sqrt{(2\pi)^m |M_k^{i_k^-}|}} e^{-\frac{1}{2}(C\hat{x}_{k|k-1}^{i_k^-})'(M_k^{i_k^-})^{-1}(*)} \\
&\times \int_{\mathbb{R}^m} e^{-\frac{1}{2}y_k'[Y+(M_k^{i_k^-})^{-1}]y_k + (C\hat{x}_{k|k-1}^{i_k^-})'(M_k^{i_k^-})^{-1}y_k} \mathrm{d}y_k \\
&\overset{(b)}{=} s_k + \frac{1-2s_k}{\sqrt{(2\pi)^m |M_k^{i_k^-}|}} e^{-\frac{1}{2}(C\hat{x}_{k|k-1}^{i_k^-})'(M_k^{i_k^-})^{-1}(*)} \\
&\times \sqrt{\frac{(2\pi)^m}{|Y+(M_k^{i_k^-})^{-1}|}} \\
&\times e^{\frac{1}{2}(C\hat{x}_{k|k-1}^{i_k^-})'(M_k^{i_k^-})^{-1}[Y+(M_k^{i_k^-})^{-1}]^{-1}(M_k^{i_k^-})^{-1}C\hat{x}_{k|k-1}^{i_k^-}} \\
&\overset{(c)}{=} s_k + \frac{1-2s_k}{\sqrt{|(M_k^{i_k^-})Y+I|}} e^{-\frac{1}{2}(C\hat{x}_{k|k-1}^{i_k^-})'[Y^{-1}+(M_k^{i_k^-})]^{-1}(*)}, \quad (13)
\end{aligned}
$$

where (a) follows from (5); (b) follows from the Gaussian integral and (c) follows from the matrix inversion lemma. □

The following notions are defined.

$$
\begin{aligned}
\hat{x}_{k|k} &= \mathbb{E}\left\{x_k|\mathcal{I}_k\right\}, & \hat{x}_{k|k-1} &= \mathbb{E}\left\{x_k|\mathcal{I}_{k-1}\right\}, \\
e_{k|k} &= x_k - \hat{x}_{k|k}, & e_{k|k-1} &= x_k - \hat{x}_{k|k-1}, \\
P_{k|k} &= \mathbb{E}\left\{e_{k|k}e_{k|k}'\right\}, & P_{k|k-1} &= \mathbb{E}\left\{e_{k|k-1}e_{k|k-1}'\right\}.
\end{aligned}
$$

In view of Lemma 4 and Lemma 5, the following is straightforward from (3) and (4).

Theorem 6. *With the stochastic event-trigger sensor schedule and in the presence of packet drops, the conditional probability density functions (pdfs) of x_k are Gaussian mixture, i.e.,*

$$
\begin{aligned}
f(x_k|\mathcal{I}_k) &= \sum_{i=0}^{2^{k+1}-1} \alpha_{k|k}^i \mathcal{N}_{x_k}(\hat{x}_{k|k}^i, P_{k|k}^i), \\
f(x_k|\mathcal{I}_{k-1}) &= \sum_{i=0}^{2^{k}-1} \alpha_{k-1|k-1}^i \mathcal{N}_{x_k}(\hat{x}_{k|k-1}^i, P_{k|k-1}^i),
\end{aligned}
$$

where $\hat{x}_{k|k}^i, P_{k|k}^i, \hat{x}_{k|k-1}^i, P_{k|k-1}^i, \alpha_{k|k}^i, \alpha_{k|k-1}^i$ are computed in Lemma 4 and Lemma 5. Moreover, the optimal estimate with the corresponding estimation error covariance can be calculated by the Gaussian sum filter [15] and are given by

$$
\begin{aligned}
\hat{x}_{k|k} &= \sum_{i=0}^{2^{k+1}-1} \alpha_{k|k}^i \hat{x}_{k|k}^i, \\
P_{k|k} &= \sum_{i=0}^{2^{k+1}-1} \alpha_{k|k}^i \left(P_{k|k}^i + (\hat{x}_{k|k}^i - \hat{x}_{k|k})(\hat{x}_{k|k}^i - \hat{x}_{k|k})'\right), \\
\hat{x}_{k|k-1} &= \sum_{i=0}^{2^{k}-1} \alpha_{k-1|k-1}^i \hat{x}_{k|k-1}^i, \\
P_{k|k-1} &= \sum_{i=0}^{2^{k}-1} \alpha_{k-1|k-1}^i (P_{k|k-1}^i \\
&\qquad + (\hat{x}_{k|k-1}^i - \hat{x}_{k|k-1})(\hat{x}_{k|k-1}^i - \hat{x}_{k|k-1})').
\end{aligned}
$$

We can verify that when there are no packet drops, the optimal estimator degenerates to the one given in [7]. Besides, the time update of the MMSE estimator satisfies the following simple recursion

$$
\hat{x}_{k|k-1} = \sum_{i=0}^{2^{k}-1} \alpha_{k-1|k-1}^i A\hat{x}_{k-1|k-1}^i = A\hat{x}_{k-1|k-1},
$$

$$
\begin{aligned}
P_{k|k-1} &= \sum_{i=0}^{2^{k}-1} \alpha_{k-1|k-1}^i (AP_{k-1|k-1}^i A' + Q \\
&\quad + A(\hat{x}_{k-1|k-1}^i - \hat{x}_{k-1|k-1})(\hat{x}_{k-1|k-1}^i - \hat{x}_{k-1|k-1})'A') \\
&= AP_{k-1|k-1}A' + Q.
\end{aligned}
$$

However, there are no such simple relations for the measurement update.

It is immediate from Theorem 6 that the optimal estimator requires exponentially increasing computation and memory with time, which cannot be applied to practical applications. Therefore, in the following, two sub-optimal estimators with constant resource requirements are proposed.

IV. Sub-optimal Estimators

In this section, we propose two sub-optimal estimators with constant resource requirements: the fixed memory estimator and the particle filter. The two sub-optimal estimators are obtained by limiting the hypothesis length and numbers, respectively. In the following, we will describe the two sub-optimal estimators in detail.

A. Fixed Memory Estimator

The problem considered in this paper is similar to the state estimation problem of linear systems with multiplicative and additive noises [16], where it is shown that the optimal nonlinear filter is obtained from a bank of Kalman filters, which requires ever increasing memory and computation with time. Fixed memory sub-optimal estimators are therefore proposed in [16] to overcome the computational complexity. The approximations consist of restricting the probability density $f(x_k|\mathcal{I}_k)$ to depend on at most the last N random variables $\gamma_k, \ldots, \gamma_{k-N+1}$ and approximate each conditional probability density $f(x_k|\gamma_k, \ldots, \gamma_{k-N+1}, \mathcal{I}_k)$ with a Gaussian distribution. Moreover, a hypothesis merging operation is introduced at every step to prevent the increase of hypothesis numbers with time. The same principle is utilized to derive a sub-optimal estimator for the problem considered in this paper. The detailed derivations are given below.

Let Υ_k^N denote the sequence $\{\gamma_k, \ldots, \gamma_{k-N+1}\}$. At time k, instead of conditioned on all the past history $\gamma_k, \ldots, \gamma_0$, we only conditioned on the past N steps Υ_k^N, where $N \geq 2$ and have the following relation

$$
f(x_k|\mathcal{I}_k) = \sum_{\Upsilon_k^N} \Pr(\Upsilon_k^N|\mathcal{I}_k) f(x_k|\Upsilon_k^N, \mathcal{I}_k). \quad (14)
$$

It is clear from Section III that $f(x_k|\Upsilon_k^N, \mathcal{I}_k)$ is mixture Gaussian with 2^{k+1-N} components. To obtain an approximate estimator, we make the approximation that

$$
f(x_k|\Upsilon_k^N, \mathcal{I}_k) \approx \mathcal{N}_{x_k}(\hat{x}_k(\Upsilon_k^N), P_k(\Upsilon_k^N)), \quad (15)
$$

where equality holds exactly when $k = N-1$.

Therefore an approximate estimator from (14) and (15) is given by

$$
\hat{x}_{k|k} = \sum_{\Upsilon_k^N} \Pr(\Upsilon_k^N|\mathcal{I}_k)\hat{x}_k(\Upsilon_k^N), \quad (16)
$$

$$
P_{k|k} = \sum_{\Upsilon_k^N} \Pr(\Upsilon_k^N|\mathcal{I}_k)(P_k(\Upsilon_k^N) + (\hat{x}_k(\Upsilon_k^N) - \hat{x}_{k|k})(*)'). \quad (17)
$$

In the following, we show the need to introduce a hypothesis merging step and how to recursively calculate $\hat{x}_k(\Upsilon_k^N)$, $P_k(\Upsilon_k^N)$ and $\Pr(\Upsilon_k^N|\mathcal{I}_k)$. Suppose at time $k-1$, (15) holds, we have

$$
f(x_{k-1}|\Upsilon_{k-1}^{N-1}, \mathcal{I}_{k-1})
$$

$$= \sum_{\gamma_{k-N}} f(x_{k-1}, \gamma_{k-N} | \Upsilon_{k-1}^{N-1}, \mathcal{I}_{k-1})$$
$$= \sum_{\gamma_{k-N}} \frac{f(x_{k-1}|\Upsilon_{k-1}^{N}, \mathcal{I}_{k-1}) \Pr(\Upsilon_{k-1}^{N}|\mathcal{I}_{k-1})}{\Pr(\Upsilon_{k-1}^{N-1}|\mathcal{I}_{k-1})}$$
$$= \sum_{\gamma_{k-N}} \frac{f(x_{k-1}|\Upsilon_{k-1}^{N}, \mathcal{I}_{k-1}) \Pr(\Upsilon_{k-1}^{N}|\mathcal{I}_{k-1})}{\sum_{\gamma_{k-N}} \Pr(\Upsilon_{k-1}^{N}|\mathcal{I}_{k-1})}.$$

Therefore, if $f(x_{k-1}|\Upsilon_{k-1}^{N}, \mathcal{I}_{k-1})$ is Gaussian, $f(x_{k-1}|\Upsilon_{k-1}^{N-1}, \mathcal{I}_{k-1})$ is mixture Gaussian with 2 components. As a result, $f(x_k|\Upsilon_k^N, \mathcal{I}_k)$ is rarely Gaussian, which make (15) invalid. We therefore introduce a hypothesis merging step by applying a Gaussian mixture reduction to $f(x_{k-1}|\Upsilon_{k-1}^{N-1}, \mathcal{I}_{k-1})$ with moment match and make the approximation that

$$f(x_{k-1}|\Upsilon_{k-1}^{N-1}, \mathcal{I}_{k-1}) \approx \mathcal{N}_{x_{k-1}}(\hat{x}_{k-1}(\Upsilon_{k-1}^{N-1}), P_{k-1}(\Upsilon_{k-1}^{N-1})), \quad (18)$$

where

$$\hat{x}_{k-1}(\Upsilon_{k-1}^{N-1}) = \sum_{\gamma_{k-N}} \frac{\hat{x}_{k-1}(\Upsilon_{k-1}^{N}) \Pr(\Upsilon_{k-1}^{N}|\mathcal{I}_{k-1})}{\sum_{\gamma_{k-N}} \Pr(\Upsilon_{k-1}^{N}|\mathcal{I}_{k-1})}, \quad (19)$$

$$P_{k-1}(\Upsilon_{k-1}^{N-1}) = \sum_{\gamma_{k-N}} \frac{\Pr(\Upsilon_{k-1}^{N}|\mathcal{I}_{k-1})}{\sum_{\gamma_{k-N}} \Pr(\Upsilon_{k-1}^{N}|\mathcal{I}_{k-1})} (P_{k-1}(\Upsilon_{k-1}^{N}) + (\hat{x}_{k-1}(\Upsilon_{k-1}^{N}) - \hat{x}_{k-1}(\Upsilon_{k-1}^{N-1}))(*)'). \quad (20)$$

Under the approximation (18), $f(x_k|\Upsilon_k^N, \mathcal{I}_k)$ is Gaussian and its mean $\hat{x}_k(\Upsilon_k^N)$ and covariance $P_k(\Upsilon_k^N)$ can be obtained from $\hat{x}_{k-1}(\Upsilon_{k-1}^{N-1})$ and $P_{k-1}(\Upsilon_{k-1}^{N-1})$ via the Kalman filter with the new information $\{\gamma_k, \gamma_k s_k, \gamma_k s_k y_k\}$ in a similar way to Lemma 4.

For the hypothesis probability recursion, we first have that

$$\Pr(\Upsilon_{k-1}^{N-1}|\mathcal{I}_{k-1}) = \sum_{\gamma_{k-N}} \Pr(\Upsilon_{k-1}^{N}|\mathcal{I}_{k-1}). \quad (21)$$

Then $\Pr(\Upsilon_k^N|\mathcal{I}_k)$ can be obtained from $\Pr(\Upsilon_{k-1}^{N-1}|\mathcal{I}_{k-1})$ with the new information $\{\gamma_k s_k, \gamma_k s_k y_k\}$ in a similar way to Lemma 5. The fixed memory sub-optimal estimator is described in Algorithm 1.

Algorithm 1 Fixed Memory Estimator

For $k < N$, run the optimal estimator to obtain $\hat{x}_{k|k}$ and $P_{k|k}$.
For $k \geq N$,

1) **Hypothesis merging**: compute $\hat{x}_{k-1}(\Upsilon_{k-1}^{N-1}), P_{k-1}(\Upsilon_{k-1}^{N-1})$ from (19) and (20).
2) **Hypothesis time and measurement update**:
   - If $\gamma_k = 0$,
   $$\hat{x}_k(\Upsilon_k^N) = A\hat{x}_{k-1}(\Upsilon_{k-1}^{N-1}),$$
   $$P_k(\Upsilon_k^N) = AP_{k-1}(\Upsilon_{k-1}^{N-1})A' + Q.$$
   - If $\gamma_k = 1$,
   $$\hat{x}_k^- = A\hat{x}_{k-1}(\Upsilon_{k-1}^{N-1}),$$
   $$P_k^- = AP_{k-1}(\Upsilon_{k-1}^{N-1})A' + Q,$$
   $$K_k = P_k^- C'[CP_k^- C' + R + (1 - s_k\gamma_k)Y^{-1}]^{-1},$$
   $$\hat{x}_k(\Upsilon_k^N) = (I - K_k C)\hat{x}_k^- + K_k s_k \gamma_k y_k,$$
   $$P_k(\Upsilon_k^N) = (I - K_k C)P_k^-.$$
3) **Hypothesis probability merging**: compute $\Pr(\Upsilon_{k-1}^{N-1}|\mathcal{I}_{k-1})$ via (21).
4) **Hypothesis probability time and measurement update**: compute
   $$\Pr(\Upsilon_k^N|\mathcal{I}_k) \propto \Pr(\Upsilon_k^N|\mathcal{I}_{k-1}) \Pr(s_k\gamma_k|\Upsilon_k^N, \mathcal{I}_{k-1}),$$
   where
   - if $\gamma_k = 0$,
   $$\Pr(\Upsilon_k^N|\mathcal{I}_{k-1}) = p \Pr(\Upsilon_{k-1}^{N-1}|\mathcal{I}_{k-1}),$$
   $$\Pr(s_k\gamma_k|\Upsilon_k^N, \mathcal{I}_{k-1}) = 1 - s_k\gamma_k.$$
   - if $\gamma_k = 1$,
   $$\Pr(\Upsilon_k^N|\mathcal{I}_{k-1}) = (1 - p) \Pr(\Upsilon_{k-1}^{N-1}|\mathcal{I}_{k-1}),$$
   $$\Pr(s_k\gamma_k|\Upsilon_k^N, \mathcal{I}_{k-1}) = s_k\gamma_k + \frac{1 - 2s_k\gamma_k}{\sqrt{|M_k Y + I|}} e^{-\frac{1}{2}(C\hat{x}_k^-)'[Y^{-1}+M_k]^{-1}(*)},$$
   where $M_k = CP_k^- C' + R$.
5) **State estimate**: compute $\hat{x}_{k|k}$ and $P_{k|k}$ from (16) and (17).

B. Particle Filter

Particle filter is a well-established numerical method to approximate non-linear and non-Gaussian probability distributions by using a set of samples [17]. We can utilize the particle filter to approximate the posterior distribution $f(x_k, \Gamma_k|\mathcal{I}_k)$ and then to obtain a sub-optimal estimator. However, the structure of the considered problem allows us to use the Rao-Blackwellization method [18] and work more efficiently by sampling only from a conditional distribution to reduce the computational burden. Specifically, since $f(x_k|\Gamma_k, \mathcal{I}_k)$ is Gaussian, it can be analytically calculated from the Kalman filter. We only need to use samples to approximate $\Pr(\Gamma_k|\mathcal{I}_k)$ and then merge the two parts together to obtain an approximation to the desired posterior distribution $f(x_k|\mathcal{I}_k)$. The detailed derivation is given below.

In our estimation problem, it is clear from (3) that the difficulty is caused by the ever expanding probability space for the hypothesis Γ_k. The particle filter motivates us to use finite samples to approximate the entire probability space of Γ_k, i.e.,

$$\Pr(\Gamma_k|\mathcal{I}_k) \approx \sum_{i=1}^{N} \delta(\Gamma - \Gamma_k^i) \Pr(\Gamma_k^i|\mathcal{I}_k),$$

where δ is the Dirac delta measure and N is the number of samples. With a slight abuse of notion, we use Γ_k^i here to denote the i-th sample drawn from $\Pr(\Gamma_k|\mathcal{I}_k)$. Therefore, an approximate to the posterior is given by

$$f(x_k|\mathcal{I}_k) \approx \sum_{i=1}^{N} f(x_k|\Gamma_k^i, \mathcal{I}_k) \Pr(\Gamma_k^i|\mathcal{I}_k), \quad (22)$$

based on which we can obtain a sub-optimal estimator.

However, since $\Pr(\Gamma_k|\mathcal{I}_k)$ is unknown, we can not directly sample from this probability distribution. The commonly used method to overcome the circumstance is to sample particles $\{\Gamma_k^i\}_{i=1,\ldots,N}$ from an importance density $q(\Gamma_k|\mathcal{I}_k)$. Then we can approximate $\Pr(\Gamma_k|\mathcal{I}_k)$ with

$$\Pr(\Gamma_k|\mathcal{I}_k) \approx \sum_{i=1}^{N} w_k^i \delta(\Gamma_k - \Gamma_k^i), \quad (23)$$

where w_k^i is the normalized importance weight and $w_k^i \propto \Pr(\Gamma_k^i|\mathcal{I}_k)/q(\Gamma_k^i|\mathcal{I}_k)$. Moreover, if the importance density is chosen to factorize such that

$$q(\Gamma_k|\mathcal{I}_k) = q(\gamma_k|\Gamma_{k-1}, \mathcal{I}_k)q(\Gamma_{k-1}|\mathcal{I}_{k-1}),$$

one can obtain new particle $\Gamma_k^i \sim q(\Gamma_k|\mathcal{I}_k)$ by augmenting each existing particle $\Gamma_{k-1}^i \sim q(\Gamma_{k-1}|\mathcal{I}_{k-1})$ with the new state $\gamma_k^i \sim q(\gamma_k|\Gamma_{k-1},\mathcal{I}_k)$.

It has been proved that the degeneracy problem is inevitable with the above sequential importance sampling [17]. That is, after a few iterations, all but one particle will have weights that are very close to zero. Therefore, a large computation is devoted to updating particles that have negligible contribution to the final estimate. One way to alleviate this problem is to select a good importance density. It is shown in [18] that the optimal importance density to minimize some degeneracy measure is given by

$$q(\gamma_k|\Gamma_{k-1}^i,\mathcal{I}_k) = \Pr(\gamma_k|\Gamma_{k-1}^i,\mathcal{I}_k).$$

For the considered problem in this paper, we can analytically calculate $\Pr(\gamma_k|\Gamma_{k-1}^i,\mathcal{I}_k)$, which is given as follows. If $\gamma_k s_k = 1$, $\Pr(\gamma_k|\Gamma_{k-1}^i,\mathcal{I}_k) = \gamma_k$. If $\gamma_k s_k = 0$,

$$\begin{aligned}\Pr(\gamma_k|\Gamma_{k-1}^i,\mathcal{I}_k) &= \Pr(\gamma_k|\Gamma_{k-1}^i,\gamma_k s_k = 0,\mathcal{I}_{k-1})\\ &\propto \Pr(\gamma_k,\gamma_k s_k = 0|\Gamma_{k-1}^i,\mathcal{I}_{k-1})\\ &= \Pr(\gamma_k s_k = 0|\Gamma_k^i,\mathcal{I}_{k-1})\Pr(\gamma_k).\end{aligned}$$

Therefore, we have if $\gamma_k s_k = 0$,

$$\begin{aligned}&\Pr(\gamma_k = 0|\Gamma_{k-1}^i,\mathcal{I}_k)\\ &\propto p\Pr(\gamma_k s_k = 0|\gamma_k^i = 0,\Gamma_{k-1}^i,\mathcal{I}_{k-1}) = p,\end{aligned} \tag{24}$$

and

$$\begin{aligned}&\Pr(\gamma_k = 1|\Gamma_{k-1}^i,\mathcal{I}_k)\\ &\propto (1-p)\Pr(\gamma_k s_k = 0|\gamma_k^i = 1,\Gamma_{k-1}^i,\mathcal{I}_{k-1})\\ &\overset{(a)}{=} \frac{1-p}{\sqrt{|M_k^i Y + I|}} e^{-\frac{1}{2}(C\hat{x}_{k|k-1}^i)'[Y^{-1}+M_k^i]^{-1}(*)},\end{aligned} \tag{25}$$

where $M_k^i = CP_{k|k-1}^i C' + R$, and (a) can be calculated similarly as (13).

As $\Pr(\gamma_k|\Gamma_{k-1}^i,\mathcal{I}_k)$ is known, we can select it as the importance density to alleviate the degeneracy problem. Moreover, we can verify that the particle weight becomes $w_k^i = 1/N$ with this optimal importance density. Based on the approximations (22) and (23), we have the following state estimate

$$\hat{x}_{k|k} = \frac{1}{N}\sum_{i=1}^N \hat{x}_{k|k}^i, \tag{26}$$

$$P_{k|k} = \frac{1}{N}\sum_{i=1}^N \left(P_{k|k}^i + (\hat{x}_{k|k}^i - \hat{x}_{k|k})(\hat{x}_{k|k}^i - \hat{x}_{k|k})'\right), \tag{27}$$

where $\hat{x}_{k|k}^i$ and $P_{k|k}^i$ are the mean and covariance of $f(x_k|\Gamma_k^i,\mathcal{I}_k)$, respectively. The detailed particle filter is described in Algorithm 2.

Remark 7. *The variational Bayesian (VB) method can also be adopted here to design sub-optimal estimators with constant resource requirements. The VB method approximates the complex posterior distribution with a proposal distribution, which is parameterized in certain forms to represent necessary statistics. These parameters will be determined by optimizing the statistical distance between the implicit posterior distribution and the proposal distribution. The VB method has been utilized to approximate Gaussian mixture distributions in [19]. Interesting readers can refer to [19] and references therein for details.*

Remark 8. *The proposed estimators can be extended to the case with Markovian packet drops. Since $f(x_k|\Gamma_k^i,\mathcal{I}_k)$ is Gaussian if packet drops are independent with the system state, the conditional posterior distribution $f(x_k|\mathcal{I}_k)$ is still mixture Gaussian even for Markovian packet drops. We can use similar approach to derive the MMSE estimator and sub-optimal estimators. The main difference compared with the i.i.d. case is that the iteration of $\Pr(\Gamma_k^i|\mathcal{I}_k)$ is different and more complex for correlated packet drops.*

Algorithm 2 Particle Filter

1) **Time update for each particle**: For $i = 1,\dots,N$,
   - If $k = 0$, then
   $$x_{0|-1}^i = 0, P_{0|-1}^i = \Sigma_0.$$
   - If $k > 0$, then
   $$\begin{aligned}\hat{x}_{k|k-1}^i &= A\hat{x}_{k-1|k-1}^i,\\ P_{k|k-1}^i &= AP_{k-1|k-1}^i A' + Q.\end{aligned}$$
2) **Sampling new particles**: If $\gamma_k s_k = 1$, let $\gamma_k^i = 1$. If $\gamma_k s_k = 0$, generate γ_k^i from the distribution described by (24), (25).
3) **Measurement update for each particle**:
   - If $\gamma_k^i = 0$,
   $$\hat{x}_{k|k}^i = \hat{x}_{k|k-1}^i, P_{k|k}^i = P_{k|k-1}^i.$$
   - If $\gamma_k^i = 1$,
   $$\begin{aligned}K_k^i &= P_{k|k-1}^i C'[CP_{k|k-1}^i C' + R + (1 - s_k\gamma_k)Y^{-1}]^{-1},\\ \hat{x}_{k|k}^i &= (I - K_k^i C)\hat{x}_{k|k-1}^i + s_k\gamma_k K_k^i y_k,\\ P_{k|k}^i &= P_{k|k-1}^i - K_k^i CP_{k|k-1}^i.\end{aligned}$$
4) **State estimate**: compute $\hat{x}_{k|k}$ and $P_{k|k}$ from (26) and (27).

V. Simulations

In simulations, we adopt the same system parameters as in [7], which are

$$A = \begin{bmatrix}0.8 & \\ & 0.95\end{bmatrix}, C = [1, 1], \Sigma_0 = Q = \begin{bmatrix}1 & \\ & 1\end{bmatrix}, R = 1.$$

We conduct simulations with the optimal estimator, the oracle estimator, the OLSET-KF estimator in [7], the fixed memory estimator and the particle filter. The OLSET-KF estimator does not consider packet drops. When the estimator fails to receive a packet, it always assumes that this is caused by the hold of transmission from the scheduler. The oracle estimator is the optimal estimator under the assumption that the estimator knows the value of γ_k at each step, which is given in Algorithm 3. Clearly, the oracle estimator has the smallest mean square error (MSE) and can be used as a benchmark to evaluate the performance of other estimators. In simulations, the schedule parameter Y is selected as $Y = 1$, $N = 2$ is selected for the fixed memory estimator and the particle numbers is set to 20 in the particle filter. The source code for all the simulations in this section is available at [20].

Firstly, we compare the performance of different estimators and we adopt Monte Carlo methods with 1500 independent experiments to evaluate the sum of MSE $\sum_{k=0}^{9}\mathbb{E}\left\{\|x_k - \hat{x}_{k|k}\|^2\right\}$ under different packet drop rates. The simulation results are illustrated in Fig. 2, where the relative sum of MSE is plotted. The relative sum of MSE is defined as the sum of MSE of an estimator divided by the sum of MSE of the oracle estimator. It is clear from Fig. 2 that the estimation error of the fixed memory estimator and the particle filter are close to that of the optimal estimator and is much smaller than the OLSET-KF, which shows the superior performance of the proposed sub-optimal estimators and also indicates the advantage of considering packet drops in the remote state estimation problem.

Moreover, in the case of $p = 0$ and $p = 1$, the sum of MSE of all the estimators are equal. This is because when $p = 0$ ($p = 1$), the

Algorithm 3 Oracle Estimator

1) **Initialization**:

$$\hat{x}_{0|-1} = 0, P_{0|-1} = \Sigma_0.$$

2) **Measurement update**:
   - If $\gamma_k = 0$,

$$\hat{x}_{k|k} = \hat{x}_{k|k-1}, P_{k|k} = P_{k|k-1}.$$

   - If $\gamma_k = 1$,

$$\begin{aligned}\hat{x}_{k|k} &= (I - K_k C)\hat{x}_{k|k-1} + K_k s_k \gamma_k y_k,\\ P_{k|k} &= P_{k|k-1} - K_k C P_{k|k-1},\\ K_k &= P_{k|k-1}C'[CP_{k|k-1}C' + R + (1 - s_k\gamma_k)Y^{-1}]^{-1}.\end{aligned}$$

3) **Time update**:

$$\begin{aligned}\hat{x}_{k+1|k} &= A\hat{x}_{k|k},\\ P_{k+1|k} &= AP_{k|k}A' + Q.\end{aligned}$$

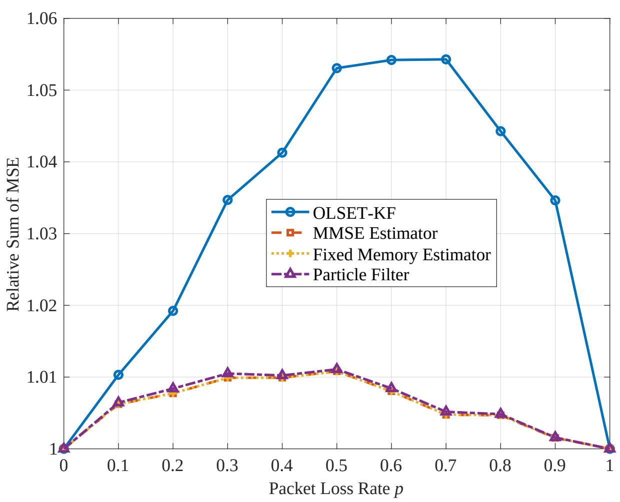

Fig. 2. Relative sum of MSE of different estimator under different packet drop rate

optimal estimator (sub-optimal estimators) assigns zero probability to all the hypotheses with a $\gamma_k = 0$ ($\gamma_k = 1$). Therefore, only the hypothesis with $\gamma_k = 1$ ($\gamma_k = 0$) for all k is preserved. As a result, the estimate of the optimal estimator (sub-optimal estimators) is the same with the oracle estimator. Therefore, for the case that $p = 0$ ($p = 1$), the optimal estimator (sub-optimal estimators) and the oracle estimator have the same sum of MSE. The recursions of the OLSET-KF and the oracle estimator are the same for the case $p = 0$, where there are no packet drops. For the case that $p = 1$, even though the recursions of the OLSET-KF and the oracle estimator are different, since they both start with $\hat{x}_{0|-1} = 0$, their estimates would always be $\hat{x}_{k|k} = 0$. Therefore, for the case that $p = 0$ and $p = 1$, the OLSET-KF and the oracle estimator have the same sum of MSE.

In the second simulation, we evaluate how the execution time of the MMSE estimator and the sub-optimal estimators grow with time. We let $p = 0.5$ and generate a sequence of noisy observations. Then we use the optimal estimator and the sub-optimal estimators to estimate the system state. The simulation is conducted with python/numpy and evaluated on Macbook Pro with 2.3 GHz Intel Core i5 processor. The program execution time with respect to time is illustrated in Fig. 3. It is clear that the program execution time of the optimal estimator is

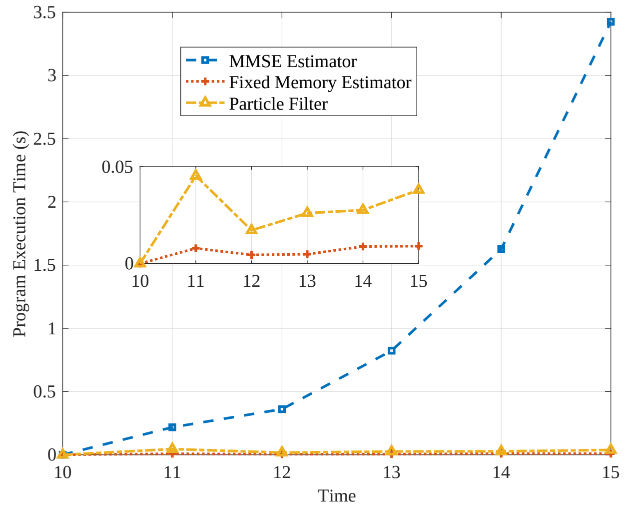

Fig. 3. Execution time of the optimal estimator and the sub-optimal estimators with respect to time

increasing exponentially with time. However, the program execution time of the two sub-optimal estimators does not change much when time increases, which demonstrates the effectiveness of the proposed sub-optimal estimators in reducing the computational complexity.

VI. Conclusions

This paper studies the remote state estimation problem of linear systems with stochastic event-triggered sensor schedulers in the presence of packet drops. The posterior distributions at the estimator side are computed. Recursive MMSE estimators are derived. Sub-optimal estimators to reduce the computational complexity are proposed. However, the performance of sub-optimal estimators can only be evaluated via simulations. Sub-optimal estimators with provable performance guarantees are to be proposed.

References

[1] C. Yang and L. Shi, "Deterministic sensor data scheduling under limited communication resource," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 5050–5056, 2011.

[2] L. Shi, P. Cheng, and J. Chen, "Sensor data scheduling for optimal state estimation with communication energy constraint," *Automatica*, vol. 47, no. 8, pp. 1693–1698, 2011.

[3] Y. Mo, E. Garone, and B. Sinopoli, "On infinite-horizon sensor scheduling," *Systems & control letters*, vol. 67, pp. 65–70, 2014.

[4] M. Xia, V. Gupta, and P. J. Antsaklis, "Networked state estimation over a shared communication medium," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1729–1741, 2017.

[5] S. Trimpe, "Stability analysis of distributed event-based state estimation," in *Proceedings of the 53rd Annual Conference on Decision and Control*, (Los Angeles, California, USA), pp. 2013–2019, 2014.

[6] J. Wu, Q. Jia, K. H. Johansson, and L. Shi, "Event-based sensor data scheduling: Trade-off between communication rate and estimation quality," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 1041–1046, 2013.

[7] D. Han, Y. Mo, J. Wu, S. Weerakkody, B. Sinopoli, and L. Shi, "Stochastic event-triggered sensor schedule for remote state estimation," *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2661–2675, 2015.

[8] S. Trimpe and R. D'Andrea, "Event-based state estimation with variance-based triggering," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3266–3281, 2014.

[9] Y. Li, J. Wu, and T. Chen, "Transmit power control and remote state estimation with sensor networks: A bayesian inference approach," *Automatica*, vol. 97, pp. 292–300, 2018.

[10] A. S. Leong, S. Dey, and D. E. Quevedo, "Sensor scheduling in variance based event triggered estimation with packet drops," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1880–1895, 2017.

[11] E. Kung, J. Wu, D. Shi, and L. Shi, "On the nonexistence of event-based triggers that preserve gaussian state in presence of package-drop," in *Proceedings of the 2017 American Control Conference*, (Seattle, USA), pp. 1233–1237, 2017.

[12] L. Xu, Y. Mo, and L. Xie, "Remote state estimation with stochastic event-triggered sensor schedule in the presence of packet drops," in *Proceedings of the 2019 American Control Conference*, (Philadelphia, PA, USA), pp. 5780–5785, 2019.

[13] O. L. d. V. Costa, M. D. Fragoso, and R. P. Marques, *Discrete-time Markov jump linear systems*. Probability and its applications, London: Springer, 2005.

[14] H. Lin, H. Su, Z. Shu, Z.-G. Wu, and Y. Xu, "Optimal estimation in udp-like networked control systems with intermittent inputs: stability analysis and suboptimal filter design," *IEEE Transactions on Automatic Control*, vol. 61, no. 7, pp. 1794–1809, 2016.

[15] B. D. Anderson and J. B. Moore, *Optimal filtering*. New Jersey: Prentice-Hall, Inc., 1979.

[16] A. G. Jaffer and S. C. Gupta, "On estimation of discrete processes under multiplicative and additive noise conditions," *Information Sciences*, vol. 3, no. 3, pp. 267–276, 1971.

[17] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.

[18] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.

[19] Y. Ma, S. Zhao, and B. Huang, "Multiple-model state estimation based on variational bayesian inference," *IEEE Transactions on Automatic Control*, vol. 64, no. 4, pp. 1679–1685, 2019.

[20] "gaussian-mixture-simulation, github reopsitory." https://github.com/lxutn/gaussian-mixture-simulation, 2019.