2022

Outdated Measurements Are Still Useful For Multi-Sensor Linear Control Systems With Random Communication Delays

Jia Zhang

Chih-Chun Wang

Outdated Measurements Are Still Useful For Multi-Sensor Linear Control Systems With Random Communication Delays

Jia Zhang
School of ECE, Purdue University
Email: zhan2030@purdue.edu

Chih-Chun Wang
School of ECE, Purdue University
Email: chihw@purdue.edu

***Abstract*— *Linear systems* are a widely used model for the control tasks of modern cyber physical systems around their stationary state(s), e.g., smart grids, remote health applications, and autonomous driving systems. Specifically, each sensor first *compresses* its own measurement and then sends it to the controller. Due to the inevitable random communication delay, the controller needs to decide how to fuse the received information to compute the desired control action. Suppose a fusion center has received several measurements over time. One common belief is that *the control decision should be made solely based on the latest measurement of each sensor while ignoring the older/stale measurements from the same sensor.* This work shows that while such a strategy is optimal in a single-sensor environment, it can be strictly suboptimal for a multi-sensor system. Namely, if one properly fuses both the *latest* and *outdated measurements* from each of the sensors, one can strictly improve the underlying control system performance. The numerical evaluation shows that even at a very low communication rate of 8 bits per measurement per sensor, the proposed scheme achieves a state variance of only 5% away from the best possible achievable L2 norm. It is 15% better than the MMSE fusion scheme using exclusively the freshest measurements (while discarding outdated ones).**

***Index Terms*—Rate-stability tradeoff, random delay, information fusion, age of information, linear control systems**

I. Introduction

With the increasing number of Internet-of-Things applications such as smart grids, remote health applications, and autonomous driving systems, multi-sensor linear systems and the corresponding control schemes are widely deployed to control the system around its stationary system state [1]–[3]. With the sensors spread over multiple locations, the measurement of each sensor is often compressed first before transmission. Upon receiving the measurements, the controller then fuses the information and designs the control action(s) in a holistic fashion. One common information fusion strategy is as follows. Say there are K sensors in the system. The controller keeps the most recently received measurement of each sensor, one for each sensor, and thus totally K measurements (i.e., discards/ignores the older measurements from the same sensor(s)). Then *the control decision is made solely based on these K measurements.* This work shows that while such a strategy is optimal in a single-sensor environment, it can be strictly suboptimal for a multi-sensor system. Namely, if one properly fuses both the *latest* and *outdated measurements* from each of the K sensors, one can strictly improve the underlying control system performance.

From a more technical perspective, the goal of this work is to explore the tradeoff among compression ratio, random communication delay, and the multi-sensor control system performance. Specifically, we first derive an analytical lower bound of the L2 state norm for all possible data compression/fusion schemes. Then we analyze and design new schemes that approach the optimal performance (the aforementioned L2 lower bound) under any given delay distribution by explicitly capitalizing the benefits of *outdated measurements*.

A. Contribution 1: The smallest attainable L2 state norm - $\mathsf{av.D}_{\min}$

In a multi-sensor linear control system, the measurement of each sensor will experience its own random delay before arriving at the controller. It is clear that the longer the (random) delay, the worse the underlying control system performance due to the increased staleness of the information. Nonetheless, the analytical relationship between the multi-sensor random delay vector distribution and the achievable control system performance remains an open problem. To characterize the inherent tradeoff between the two, we use the *expected L2 state norm* as the performance metric. For any given delay vector distribution, we derive an L2 state norm lower bound, denoted by $\mathsf{av.D}_{\min}$, for all possible schemes while assuming the observation noise and state disturbance are both Gaussian distributed. Our approach starts by first converting the given *delay vector* distribution to the so-called Age-of-Information vector (AoI vector) distribution, a new concept that has attracted significant attention in the networking community [4]–[6]. The AoI vector distribution is then combined with the Riccati solutions of the *AoI-aware* Kalman Filters (KF) to derive the lower bound $\mathsf{av.D}_{\min}$.

The investigation of the lower bound $\mathsf{av.D}_{\min}$ is a sub-problem in the general framework of rate-cost tradeoff of linear control systems [7]–[9], and see the summaries in [8], [9]. Specifically, the rate-cost tradeoff results focus on the zero-delay (and/or deterministic delay) scenarios and study the tradeoff of *compression rate R* versus the expected *system*

This work was supported in parts by NSF under Grant CCF-1618475, Grant CCF-1816013, and Grant CCF-2008527 and Grant CNS-2107363.

state cost D [7], [8]. Since our $\mathsf{av.D}_{\min}$ lower bound places no constraint on the compression rate R, it can be viewed as the smallest D value when the compression rate is allowed to be arbitrarily large, i.e., $R \to \infty$. Comparing to the existing results on the entire (D, R) tradeoff curve [8], [9], our results have a narrower focus on one particular asymptote[1] (i.e., $R \to \infty$) but consider the more general multi-sensor random delay vector model than the existing one-sensor zero/deterministic delay settings in [8], [11], [12].

B. Contribution 2: A data fusion scheme that approaches the lower bound $\mathsf{av.D}_{\min}$

We propose a new information fusion scheme, CQE, which stands for Cumulative Quantized Estimation (CQE) fusion. The main idea is to combine cumulatively the entire history of *asynchronously arriving*, *randomly delayed* measurements from multiple sensors. For benchmarking purposes, we considered the MMSE scheme proposed in [13]. Specifically, under the setting of deterministic zero delay, [13] devises the optimal MMSE scheme that fuses the latest estimation from each of the K sensors. We modify the scheme in [13] so that it can handle the scenario of necessary data compression plus random delay considered herein. But we keep the same feature of optimally fusing only the latest estimations. We term such a scheme, LQE, which stands for Latest Quantized Estimation (LQE) fusion. Our experiment shows that with using the same average number of bits per sensor per second, CQE achieves an L2 state norm 15% smaller than LQE fusion.

In addition to the superior empirical performance, we also prove analytically that under a single sensor setting (i) LQE is provably optimal; and (ii) CQE will automatically discard the outdated measurements and thus result in the same mechanism as the optimal LQE scheme. However, for the multi-sensor setting, (iii) LQE is strictly suboptimal; and (iv) CQE fusion will intelligently utilize the outdated measurements (together with the latest measurements) and outperform the LQE scheme.

C. Comparison to Existing Results

Many researchers have studied covariance-based information fusion for networked control systems, see [14] for a detailed summary. Some example algorithms that were proposed for delay-free sensor networks are summarized as follows. [15] and [16] studied the fusion for two sensors using maximal likelihood estimation and addressed the effect of cross correlation between different sensors to the fusion performance. In [17], a covariance intersection method is proposed for the case where the correlations between sensors are unknown. [13] proposed a Kalman filter based fusion scheme for zero-delay settings, which minimizes the mean square estimation (fusion) error by linearly combining the freshest local estimates from sensors at the controller. Our LQE scheme is designed as an extension of [13] from the zero-delay to the random delay settings.

[18] considered a scenario where random delays and packet losses occur and observations are transmitted from sensors to the controller. They proposed a constant-gain estimation scheme that stores a few recently arrived observations. In comparison to their scheme, we assume that the "measurements" are generated by some local preprocessing at the sensor while [18] assumes the transmission of pure observations (without further processing). Sending pre-processed measurement is known to achieve superior performance in a single-sensor setting [8], [9], though at the cost of additional computation at the sensors. This type of local sensors computation/processing before transmission is commonly referred to as smart-sensors [19] schemes.

[20] studied a scenario where missing observations and bounded random delays occur, and that the systems were assumed inherently stable. A fusion method which utilizes the latest local information for each sensor was proposed to minimize the mean square estimation error. In [21], the authors considered sensor networks on unmanned vehicles where each sensor observes periodically, and the sensors have different observation cycles. The sensors detect the velocity and the location of an overtaking vehicle, and the central processor generates global estimates of sensors' local estimates. Three sensor-to-global fusion schemes (fusion with memory) are implemented for comparison, including adapted Kalman filter, covariance intersection, and information matrix fusion. However, they left out discussions for the network phenomena, such as quantization and random delays.

The rest of the paper is organized as following: Section II describes the system and problem formulation, and describes AoI and Kalman filter, two elements for extensive use in later sections. Section III presents a method to convert a feedback system with deterministic AoI vector to a non-delay augmented system and derives the analytical smallest achievable L2 state norm. Section IV presents the two newly-proposed fusion schemes. Section V demonstrates the experiment results. Section VI concludes the paper with a summary and a future work direction.

II. Problem Formulation

Consider a discrete time-invariant linear control system

$$\mathbf{x}(t+1) = A\mathbf{x}(t) + \mathbf{u}(t) + \mathbf{w}(t), \tag{2}$$

$$\mathbf{y}_k(t) = C_k\mathbf{x}(t) + \mathbf{v}_k(t), \quad \forall k \in [1, K]. \tag{3}$$

where $\mathbf{x}(t)$ is an N-dimensional column vector that represents the system state at time t; A is the $N \times N$ state evolution matrix; $\mathbf{u}(t)$ is the N-dimensional control action (column)

[1] As will be seen in Section V, even with very small $R = 8$ bits per measurement per sensor, our schemes are already within 5% of $\mathsf{av.D}_{\min}$ that assumes $R \to \infty$. This is also the reason that in our random delay setting, the analysis efforts are devoted to finding the $\mathsf{av.D}_{\min}$ with $R \to \infty$, rather than characterizing the entire tradeoff curve (D, R). From the analysis perspective, the other asymptote, i.e., $R_{\min}$ when $D \to \infty$, also sheds important intuition of the underlying system. Specifically [8]–[10] showed that for Gaussian linear control systems with a system state evolution matrix A:

$$R_{\min} = \sum_{\text{eigenvalues of } A : |\lambda_i| > 1} \log_2(|\lambda_i|). \tag{1}$$

Since this work focuses on practical applications with small D, the asymptote $R_{\min}$ with $D \to \infty$ is beyond our scope.

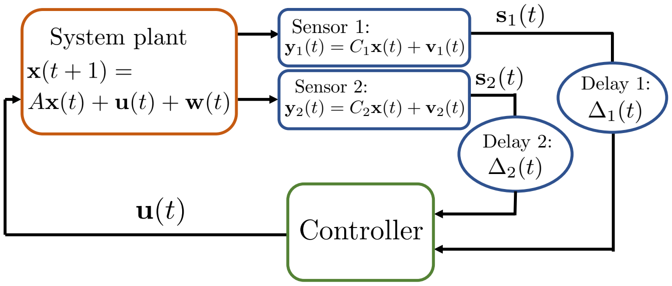

Fig. 1. linear control system with multi-sensors and random delays

vector at time t; and $\mathbf{w}(t)$ is the N-dimensional state disturbance, which is assumed to be i.i.d. Gaussian with zero mean and covariance matrix $\Sigma_{\mathbf{w}}$. We assume the system state is initialized with $\mathbf{x}(-1) = \mathbf{0}$ and $\mathbf{u}(-1) = \mathbf{0}$, and the goal is to design the control action $\mathbf{u}(t)$.

There are $K \geq 1$ sensors and the observation $\mathbf{y}_k(t)$ of sensor $k \in [1, K]$ is an M_k-dimensional column vector as described in (3), in which C_k is an $M_k \times N$ observation matrix; and $\mathbf{v}_k(t)$ is the M_k-dimensional i.i.d. Gaussian observation noise with zero mean and $M_k \times M_k$ covariance matrix $\Sigma_{\mathbf{v}_k}$.

We assume the overall system $\{A, [C_1^{\mathrm{T}}, C_2^{\mathrm{T}}, \cdots, C_K^{\mathrm{T}}]^{\mathrm{T}}\}$ is observable; the covariance matrices $\Sigma_{\mathbf{w}}$ and $\Sigma_{\mathbf{v}_k}$ are of full rank; and all quantities in (2) and (3) are of real-valued (no imaginary part).

At each time slot t, sensor k generates a binary bit string $\mathbf{s}_k(t)$ based on all its past observations $\{\mathbf{y}_k(\tau) : \tau \leq t\}$, i.e.,

$$\mathbf{s}_k(t) = f_{k,t}(\{\mathbf{y}_k(\tau) : \tau \leq t\}) \tag{4}$$

where $f_{k,t}$ is the *encoding function* at time t. We use $|\mathbf{s}_k(t)|$ to denote the length of $\mathbf{s}_k(t)$ and we allow *variable length encoding*, i.e., $|\mathbf{s}_k(t)|$ may depend on the input $\{\mathbf{y}_k(\tau) : \tau \leq t\}$.

The encoded bit string $\mathbf{s}_k(t)$ is then *time-stamped* and sent to the central controller at time t through a digital, noiseless but *randomly delayed* channel. That is, the string $\mathbf{s}_k(t)$ sent by sensor-k at time t will arrive at the controller at time $t+\Delta_k(t)$. We assume the experienced delay $\Delta_k(t)$ is i.i.d. distributed, of integer value, and with bounded support $\delta_{\max,k}$ for some sufficiently large but finite $\delta_{\max,k}$. The distribution of $\Delta_k(t)$ is thus determined by its marginal pmf $p_{k,\delta} = \mathsf{Prob}(\Delta_k(t) = \delta)$ for all $\delta \in [0, \delta_{\max,k}]$.

We assume the values of $\{p_{k,\delta} : \delta \in [0, \delta_{\max,k}]\}$ is known *a priori*. Note that the i.i.d. delay model allows for out-of-order delivery, i.e., the string $\mathbf{s}_k(t_1)$ sent at time $t_1 < t_2$ may arrive at the controller later than $\mathbf{s}_k(t_2)$ if $t_1+\Delta_k(t_1) > t_2+\Delta_k(t_2)$. Also see Fig. 1 for illustration.

At time t, the controller computes the control action $\mathbf{u}(t)$ based on the (delayed) measurements it has received from K sensors by time t. That is,

$$\mathbf{u}(t) = g_t(\{\mathbf{s}_k(\tau) : k \in [1, K], \tau + \Delta_k(\tau) \leq t\}) \tag{5}$$

where $g_t(\cdot)$ is the control action function at time t. The $\mathbf{u}(t)$ is then applied instantaneously to $\mathbf{x}(t)$ via (2). For every *sum compression rate* constraint R_{sum}, we aim to solve/approach the following L2 state-norm minimization problem.

$$\min_{\{f_{k,t}, g_t\}} \quad \sup_t \mathsf{E}\{\mathbf{x}(t)^{\mathrm{T}}\mathbf{x}(t)\} \tag{6}$$

$$\text{subject to } \sup_t \sum_{k=1}^{K} \mathsf{E}\{|\mathbf{s}_k(t)|\} \leq R_{\text{sum}} \tag{7}$$

The problem formulation is complete. We conclude this section by introducing two existing concepts that would be useful when describing our results.

A. Age of information and its distribution

The communication delay $\Delta_k(t)$ of sensor k is defined from the sensor's respective. We now define a new quantity, $\Theta_k(t)$, *Age of Information* (AoI) [9], [22], that describes the freshness of measurements at time t from the controller's perspective. That is,

$$\Theta_k(t) \triangleq t - \max_{\tau} \{\tau : \tau + \Delta_k(\tau) \leq t\} \tag{8}$$

describes *how old is the most recently received measurement from sensor* k. It is clear that the distribution of the random process $\Theta_k(t)$, though being closely related, is quite different from the distribution of $\Delta_k(t)$. For example, while $\Delta_k(t)$ is i.i.d. (i.e., order-0 Markovian), $\Theta_k(t)$ is order-1 Markovian. Recall that $p_{k,\delta} \triangleq \mathsf{Prob}(\Delta_k(t) = \delta)$. The marginal of $\Theta_k(t)$ can then be computed by

$$p_{k,\theta}^{[\text{AoI}]} \triangleq \mathsf{Prob}(\Theta_k(t) = \theta) \tag{9}$$

$$= \sum_{\delta=0}^{\theta} p_{k,\delta} \cdot \prod_{i=0}^{\theta-1} \Big(\sum_{\delta=i+1}^{\delta_{\max,k}} p_{k,\delta} \Big) \tag{10}$$

where the notation $p^{[\text{AoI}]}$ emphasizes the pmf is now focusing on the AoI Θ, not the delay Δ. More detailed discussion of (10) can be found in [9].

Because of the cross-channel independence assumption, the probability of $\vec{\Theta}(t) \triangleq (\Theta_1(t), \Theta_2(t), \cdots, \Theta_K(t))$ being $\vec{\theta} = (\theta_1, \cdots, \theta_K)$ is

$$p_{\vec{\theta}}^{[\text{AoI}]} \triangleq \mathsf{Prob}(\vec{\Theta}(t) = \vec{\theta}) = \prod_{k=1}^{K} p_{k,\theta_k}^{[\text{AoI}]}. \tag{11}$$

B. Kalman filter

Kalman filter (KF) is a commonly used technique and will be used intensively when describing our results in Sections III and IV. It is worth noting that KF is derived/defined over the simple *single-sensor* and *zero-delay* setting. In the sequel we summarize its computation in Algorithm II.1. To prepare for the discussion in Sections III and IV, Algorithm II.1 considers a more general plant model $\mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t) + \mathbf{w}(t)$ where B is the control matrix, with the observation formulation unchanged as in (3).

The physical meaning of matrix P_t is that it is the estimation error covariance matrix $P_t = \mathsf{E}\{(\mathbf{x}(t) - \hat{\mathbf{x}}(t))(\mathbf{x}(t) - \hat{\mathbf{x}}(t))^{\mathrm{T}}\}$ at time t.

Algorithm II.1 Kalman Filter (KF) computation

INPUT: Matrices A, B, C, $\Sigma_{\mathbf{w}}$, and $\Sigma_{\mathbf{v}}$; and two sequences $\{\mathbf{y}(t) : t \geq 0\}$ and $\{\mathbf{u}(t) : t \geq 0\}$.

OUTPUT: A sequence of estimation $\{\hat{\mathbf{x}}(t) : t \geq 0\}$ and two sequences of matrices $\{P_t : t \geq 0\}$ and $\{\Gamma_t : t \geq 0\}$.

Initialize $\hat{\mathbf{x}}(-1) = 0$ and $\mathbf{u}(-1) = 0$; and $P_{-1} = 0$.

for $t = 0$ to ∞ **do**

$$\Phi_t = AP_{t-1}A^{\mathrm{T}} + \Sigma_{\mathbf{w}} \tag{12}$$

$$\Gamma_t = \Phi_t C^{\mathrm{T}}(C\Phi_t C^{\mathrm{T}} + \Sigma_{\mathbf{v}})^{-1} \tag{13}$$

$$\hat{\mathbf{x}}(t) = A\hat{\mathbf{x}}(t-1) + B\mathbf{u}(t-1) + \hat{\mathbf{w}}(t-1) \tag{14}$$

$$\hat{\mathbf{w}}(t-1) = \Gamma_t \left(\mathbf{y}(t) - C(A\hat{\mathbf{x}}(t-1) + B\mathbf{u}(t-1))\right) \tag{15}$$

$$P_t = \Phi_t - \Gamma_t C\Phi_t \tag{16}$$

end for

Note that we are often interested in the limits $P = \lim_{t\to\infty} P_t$ and $\Gamma = \lim_{t\to\infty} \Gamma_t$, which can be computed by solving the Riccati matrix equations or by outputting P_t (resp. Γ_t) for a sufficiently large t.

III. The Smallest Attainable L2 State Norm

The goal of this subsection is to find a closed-form expression of the minimum L2 norm in (6) when there is no compression rate constraint, i.e., $R_{\text{sum}} \to \infty$. We denote such a value by $\mathsf{av.D}_{\min}$, which will then be used as a benchmark when comparing different schemes with finite R_{sum}.

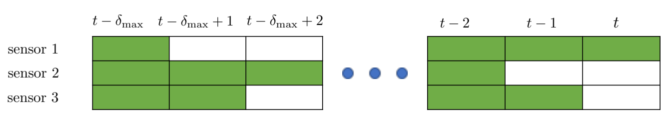

Fig. 2. Delay event 1 for a linear control system with three sensors

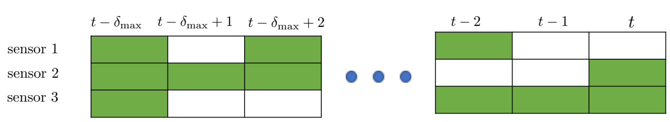

Fig. 3. Delay event 2 for a linear control system with three sensors

The main idea is as follows. With random delays and multiple sensors, the controller may encounter different events, depending on the realization of the random delay. In Fig. 3, we illustrate two delay events with different arrival patterns. A green box indicates an arrived packet, and a blank box (a hole) indicates a non-arrived local packet. Because the goal is to find the optimal $\mathsf{av.D}_{\min}$, we assume there is a genie that feeds the still missing packets from all sensors that are *earlier than the latest received packets*. Obviously, such a genie will improve the performance and thus further lower the achievable $\mathsf{av.D}_{\min}$ value. We also assume that each of the naturally received packets and the genie-aided packets contains the exact, non-quantized observation $\mathbf{y}_k(\tau)$.

Note that for each AoI realization $\vec{\Theta} = (\theta_1, \cdots, \theta_K)$, with the addition of the genie who fills the "holes", it is as if the controller is facing *deterministic* communication delays $\mathsf{Prob}(\Delta_k(t) = \theta_k, \forall k \in [1, K]) = 1$. (Note that in a deterministic delay setting, the communication delay and the AoI values are always identical, and there is no hole in the reception pattern since every packet experiences the same delay and thus will arrive in the same order as the departure.) Our idea is to first compute the smallest attainable L2 norm of this deterministic delay system, which we denote by $\mathsf{D}_{\min}^{(\vec{\theta})}$. Because the AoI value $\vec{\theta}$ faced by the controller is randomly distributed with distribution given by (9)-(11), the optimal $\mathsf{av.D}_{\min}$ can thus be computed by further averaging over $\vec{\theta}$. We then have

Proposition 1. *Denote the minimum L2 norm in* (6) *when* $R_{sum} \to \infty$ *by* $\mathsf{av.D}_{\min}$. *We have*

$$\mathsf{av.D}_{\min} \triangleq \sum_{\{\vec{\theta}\}} p_{\vec{\theta}}^{[AoI]} \cdot \mathsf{D}_{\min}^{(\vec{\theta})}. \tag{17}$$

The expression of $\mathsf{D}_{\min}^{(\vec{\theta})}$ *is provided in the subsequent Lemma 1.*

The rest of this section is dedicated to computing $\mathsf{D}_{\min}^{(\vec{\theta})}$ under an arbitrarily given deterministic vector $\vec{\theta}$. Note that the computation of $\mathsf{D}_{\min}^{(\vec{\theta})}$ assumes deterministic communication delays $\Delta_k(t) = \theta_k$ with probability one for all k and t. We now construct an equivalent, augmented system that "groups" all K deterministically-delayed sensors into a single sensor with zero delay. To that end, we first extend the system state $\mathbf{x}(t)$ to its vector version $\bar{\mathbf{x}}(t)$:

$$\bar{\mathbf{x}}(t) \triangleq (\mathbf{x}(t)^{\mathrm{T}}, \mathbf{x}(t-1)^{\mathrm{T}}, \cdots, \mathbf{x}(t-\delta_{\max})^{\mathrm{T}})^{\mathrm{T}}, \tag{18}$$

where $\delta_{\max} \triangleq \max_k \delta_{\max,k}$.

The corresponding $(\delta_{\max}+1)N$ by $(\delta_{\max}+1)N$ state evolution matrix $\bar{A}$ becomes

$$\bar{A} \triangleq \begin{bmatrix} A, \mathbf{0}_{N\times((\delta_{\max}-1)N)} & \mathbf{0}_{N\times N} \\ \mathbf{I}_{(\delta_{\max}N)} & \mathbf{0}_{(\delta_{\max}N)\times N} \end{bmatrix}, \tag{19}$$

where $\mathbf{0}_{a\times b}$ is an a-by-b zero matrix and $\mathbf{I}_a$ is the a-by-a identity matrix. Also define the (control) matrix $\bar{B}$ and corresponding system disturbance $\bar{\mathbf{w}}(t)$ as

$$\bar{B} = [\mathbf{I}_N, \mathbf{0}_{N\times(\delta_{\max}N)}]^{\mathrm{T}}, \tag{20}$$

$$\bar{\mathbf{w}}(t) = (\mathbf{w}(t)^{\mathrm{T}}, \mathbf{0}_{1\times(\delta_{\max}N)})^{\mathrm{T}}. \tag{21}$$

Together we can rewrite (2) to the new augmented equation

$$\bar{\mathbf{x}}(t+1) = \bar{A}\bar{\mathbf{x}}(t) + \bar{B}\mathbf{u}(t) + \bar{\mathbf{w}}(t), \tag{22}$$

where $\bar{\mathbf{w}}(t)$ is zero-mean Gaussian with covariance

$$\Sigma_{\bar{\mathbf{w}}} \triangleq \begin{bmatrix} \Sigma_{\mathbf{w}}, \mathbf{0}_{N\times(\delta_{\max}N)} \\ \mathbf{0}_{(\delta_{\max}N)\times((\delta_{\max}+1)N)} \end{bmatrix}. \tag{23}$$

Recall that we assume deterministic delay $\Delta_k(t) = \theta_k = \Theta_k(t)$ for sensor k. Therefore, for time t, the controller can at best know the $\mathbf{y}_k(t-\theta_k)$ value. Since $\mathbf{x}(t-\theta_k)$ is the

(θ_k+1)-th coordinate of $\bar{\mathbf{x}}(t)$, we can define the *augmented* observation matrix $\bar{C}_k^{(\theta_k)}$ of sensor k by

$$\bar{C}_k^{(\theta_k)} \triangleq [\mathbf{0}_{M_k\times(\theta_k N)}, C_k, \mathbf{0}_{M_k\times((\delta_{\max}-\theta_k)N)}], \quad (24)$$

and we then have

$$\mathbf{y}_k^{(\theta_k)}(t) = \bar{C}_k^{(\theta_k)}\bar{\mathbf{x}}(t) + \mathbf{v}_k(t-\theta_k). \quad (25)$$

By vertically stacking all K sensors together, we have

$$\bar{\mathbf{y}}^{(\vec{\theta})}(t) = \bar{C}^{(\vec{\theta})}\bar{\mathbf{x}}(t) + \bar{\mathbf{v}}^{(\vec{\theta})}(t), \quad (26)$$

where

$$\bar{C}^{(\vec{\theta})} = [(\bar{C}_1^{(\theta_1)})^{\mathrm{T}}, \cdots, (\bar{C}_K^{(\theta_K)})^{\mathrm{T}}]^{\mathrm{T}}. \quad (27)$$

And

$$\bar{\mathbf{v}}^{(\vec{\theta})}(t) = (\mathbf{v}_1(t-\theta_1)^{\mathrm{T}}, \cdots, \mathbf{v}_K(t-\theta_K)^{\mathrm{T}})^{\mathrm{T}} \quad (28)$$

is zero-mean Gaussian with covariance matrix $\Sigma_{\bar{\mathbf{v}}^{(\vec{\theta})}} = \mathrm{diag}(\{\Sigma_{\mathbf{v}_k} : \forall k\})$, an $(\sum_{k=1}^K M_k)$ by $(\sum_{k=1}^K M_k)$ matrix with the diagonal sub-matrices being $\Sigma_{\mathbf{v}_k}$. Given any $\vec{\theta}$ vector, (22) and (26) jointly convert the corresponding K-sensor, deterministic delay model to a single-sensor, zero-delay model.

We use $\bar{A}$, $\bar{B}$, $\bar{C}^{(\vec{\theta})}$, and $\mathbf{u}(t)$ as the input to the KF subroutine described in Algorithm II.1. We then use $P_{\bar{\mathbf{x}}}^{(\vec{\theta})}$ to denote the corresponding limiting output matrix P. Finally, we denote the $N\times N$ sub-matrix of $P_{\bar{\mathbf{x}}}^{(\vec{\theta})}$ in the upper-left corner as $P^{(\vec{\theta})}$. We then have the following lemma:

Lemma 1. *The smallest achievable L2 norm of the K-sensor deterministic $\vec{\theta}$-delay model is*

$$\mathsf{D}_{\min}^{(\vec{\theta})} \triangleq \mathrm{tr}(AP^{(\vec{\theta})}A^{\mathrm{T}}) + \mathrm{tr}(\Sigma_{\mathbf{w}}). \quad (29)$$

The proof of Lemma 1 is straightforward since applying KF to the augmented single-sensor system in (22) and (26) is equivalent to finding the MMSE estimator of the K-sensor deterministic $\vec{\theta}$-delay model. The upper-left corner submatrix $P^{(\vec{\theta})}$ then describes the "distance" between the MMSE estimator and the true system state $\mathbf{x}(t)$. This, plus the state perturbation $\mathbf{w}(t)$, which is not controllable by the action $\mathbf{u}(t)$, will give the minimum achievable L2 norm of this genie-aided system. The complete proof is mitigated to Appendix A.

One contribution of Proposition 1 is to show that under a random delay model, the minimum achievable L2 norm $\mathsf{av.D}_{\min}$ can be computed by averaging the deterministic delay cases $\mathsf{D}_{\min}^{(\vec{\theta})}$ based on *the marginal AoI distribution* $\{p_{k,\theta}^{[AoI]} : k,\theta\}$, not the communication delay distribution $\{p_{k,\delta} : k,\delta\}$.

IV. Achievability schemes

With unlimited communication rate $R_{\text{sum}} \to \infty$, the $\mathsf{av.D}_{\min}$ described in Proposition 1 and Lemma 1 can be easily achieved as follows. That is, for every time t, each sensor k puts the entire history of observations $\{\mathbf{y}_k(\tau) : \tau \le t\}$ in the packet/string $\mathbf{s}_k(t)$ and sends that string to the controller. As a result, whenever the controller receives the latest packet $\mathbf{s}_k(t')$, it can fill the "hole" by the history $\{\mathbf{y}_k(\tau) : \tau \le t'\}$ carried by $\mathbf{s}_k(t')$. The performance of such a scheme will match the genie-aided scenario described in Sec. III.

Such a scheme is extremely wasteful from the perspective of bandwidth consumption R_{sum} since each $\mathbf{s}_k(t)$ carries the entire history. It turns out that for the single-sensor setting ($K=1$), the following simple scheme can achieve the optimal performance without sending the entire history. That is, the sensor first computes the KF estimator $\hat{\mathbf{x}}(t)$ locally using its own measurement $\mathbf{y}(t)$. Then it sends the quantized version of $\hat{\mathbf{x}}(t)$ to the controller. Essentially, such a scheme compresses the entire history $\{\mathbf{y}(\tau) : \tau \le t\}$ into a single estimate $\hat{\mathbf{x}}(t)$. Since the goal of the controller is to compute the MMSE estimator $\hat{\mathbf{x}}(t)$ and use it to design the control action $\mathbf{u}(t)$, directly sending the quantized version of $\hat{\mathbf{x}}(t)$ from the sensor attains superior performance with much smaller bandwidth consumption than re-sending the entire history $\{\mathbf{y}(\tau) : \tau \le t\}$. See [9], [23], [24].

Motivated by the superior performance of the above scheme in a single-sensor setting, in this work, we assume each of the K sensors computes a KF estimate $\hat{\mathbf{x}}_k(t)$ based on its own observation $\mathbf{y}_k(t)$. Then the string $\mathbf{s}_k(t)$ contains only a quantized version of $\hat{\mathbf{x}}_k(t)$ (not the entire history $\{\mathbf{y}_k(\tau) : \tau \le t\}$). For ease of exposition, we use $\hat{\mathbf{x}}_k^q(t)$ to denote the quantized version of the local estimator at sensor k. Our design efforts are then placed exclusively on how the controller should *fuse* all the available information $\{\hat{\mathbf{x}}_k^q(\tau) : k \in [1,K], \tau + \Delta_k(\tau) \le t\}$ at time t from all K sensors.

A. The compression scheme at sensor k

For completeness, we briefly describe how $\hat{\mathbf{x}}_k^q(t)$ is generated. Consider a specific sensor k. (All other sensors operate similarly.) Define the matrix $B = \mathbf{I}_N$, an N-by-N identity matrix. For any time t, sensor k uses the matrices A, B, C_k, $\Sigma_{\mathbf{w}}$, $\Sigma_{\mathbf{v}_k}$, and the past actions $\{\mathbf{u}(\tau) : \tau \le t-1\}$ as the input to the KF described in Algorithm II.1. We use $\hat{\mathbf{x}}_k(t)$ to denote the output local KF estimate at time t. We then pass $\hat{\mathbf{x}}_k(t)$ via a rotated rectangular lattice quantizer.

Specifically, define the following *generating matrix*

$$G_k = \alpha \cdot U_k \begin{bmatrix} \sqrt{\gamma_{k,1}} & 0 & \cdots & 0 \\ 0 & \sqrt{\gamma_{k,2}} & \cdots & 0 \\ & \ddots & & \\ 0 & 0 & \cdots & \sqrt{\gamma_{k,N}} \end{bmatrix}, \quad (30)$$

where the N-by-N matrix U_k contains the N (column) eigenvectors of the sensor-k estimation error covariance matrix P_k, which is the limiting ($t\to\infty$) estimation error covariance for the k-th local estimator $\hat{\mathbf{x}}_k(t)$ computed by Algorithm II.1; $\gamma_{k,i}$ is the corresponding i-th eigenvalue of P_k; α is a global scaling factor that will later be used to adjust the granularity of the quantization lattices for all K sensors simultaneously. The lattice points are then generated by

$$\Pi_k = \{G_k \cdot \vec{\mathbf{i}} : \vec{\mathbf{i}} \in \mathbf{Z}^N\} \quad (31)$$

where $\vec{\mathbf{i}}$ are the integer-valued index vectors. The quantization cells are then defined as the *Voronoi regions* of the lattice points in Π_k. Because the eigenvectors of P_k are orthogonal, each Voronoi region (each cell) is a *hypercube* with the rotated axes along the eigenvectors and all Voronoi regions (all cells) are stacked together.

We label each cell (i.e., each Voronoi region) by a distinct string. Then sensor k would compute and send $\mathbf{s}_k(t)$ by

$$\mathbf{s}_k(t) = \text{the string corresponding to the cell containing } \hat{\mathbf{x}}_k(t). \tag{32}$$

To design the cell-to-string mapping, for each cell, we run Monte-Carlo simulation offline to find the (simulated) probability that $\hat{\mathbf{x}}_k(t)$ falls into a given cell. We then order the cells in the descending order of these simulated probabilities. That is, the first cell is the one with the largest simulated probability, and so on so forth. Since our goal is to minimize the string length $|\mathbf{s}_k(t)|$, the first cell is assigned with the empty string $\emptyset$. The second and the third cells are assigned with 1-bit strings 0 and 1, respectively. The 4-th to the 7-th cells are assigned with the 2-bit strings, 00, 01, 10 and 11 respectively, and so on so forth. It is worth emphasizing that the cell-to-string mapping is computed offline. When actually running the algorithm, sensor-k computes the estimate $\hat{\mathbf{x}}_k(t)$ using the local observation $\mathbf{y}_k(t)$ in real time. It then generates the string $\mathbf{s}_k(t)$ using (32) and the pre-computed cell-to-string mapping.

We now describe how the controller interprets/processes the received string $\mathbf{s}_k(\tau)$. At time t, the controller has only received the strings $\mathbf{s}_k(\tau)$ satisfying $\tau + \Delta_k(\tau) \leq t$. For each received $\mathbf{s}_k(\tau)$, the controller knows which cell it is in, say in cell_j. The central controller then computes the *quantized estimates* $\hat{\mathbf{x}}_k^q(\tau)$ by

$$\hat{\mathbf{x}}_k^q(\tau) = \mathsf{E}(\hat{\mathbf{x}}_k(\tau) | \hat{\mathbf{x}}_k(\tau) \ \text{ is in } \mathsf{cell}_j). \tag{33}$$

That is, $\hat{\mathbf{x}}_k^q(\tau)$ is the probabilistic mean of $\hat{\mathbf{x}}_k(t)$ conditioning on $\hat{\mathbf{x}}_k(t)$ falls in cell_j, the one corresponding to the received string $\mathbf{s}_k(\tau)$. Note that the mapping from each cell_j to its conditional probabilistic mean can be pre-computed offline via Monte-Carlo simulation. Then when actually carrying out the algorithm, the controller can determine $\hat{\mathbf{x}}_k^q(\tau)$ based on the cell index j in the received string $\mathbf{s}_k(\tau)$ and the precomputed mapping from j to $\hat{\mathbf{x}}_k^q(\tau)$ in (33).

This concludes the basic quantization scheme used by all our fusion algorithms. The rest of this section will describe how the received $\{\hat{\mathbf{x}}_k^q(\tau) : k \in [1, K], \tau + \Delta_k(\tau) \leq t\}$ can be *fused* to generate the controller action $\mathbf{u}(t)$ at time t.

B. Fusion schemes

1) LQE with random delays:
This scheme is an extension of the optimal MMSE fusion scheme proposed in [13] to the scenario with random delay. For any $\tau < t$, we define

$$\hat{\mathbf{x}}_k(t|\tau) \triangleq A^{t-\tau}\hat{\mathbf{x}}_k(\tau) + \sum_{s=\tau}^{t-1} A^{t-1-s}\mathbf{u}(s) \tag{34}$$

as the MMSE estimator for the system state $\mathbf{x}(t)$ based on the *k-th local estimator $\hat{\mathbf{x}}_k(\tau)$ at time τ*. That is, we pass $\hat{\mathbf{x}}_k(\tau)$ through the state evolution equation for $t - \tau$ times and also take into account the actions exerted during time τ to $t-1$.

For any fixed K-dimensional AoI vector $\vec{\theta} = (\theta_1, \cdots, \theta_K)$, we will compute K matrices $P_k^{(\theta_k)}(t)$ and C_K^2 (K-choose-2) matrices $P_{k_1,k_2}^{(\theta_{k_1},\theta_{k_2})}(t), t \geq 0$. Each matrix is of dimension N-by-N. The physical meanings of these matrices are:

$$P_k^{(\theta_k)}(t) \triangleq \mathsf{E}\{(\mathbf{x}(t) - \hat{\mathbf{x}}_k(t|t-\theta_k))(\mathbf{x}(t) - \hat{\mathbf{x}}_k(t|t-\theta_k))^{\mathrm{T}}\} \tag{35}$$

is the estimation error covariance if we use the observed estimate $\hat{\mathbf{x}}_k(t-\theta_k)$ to obtain the estimator $\hat{\mathbf{x}}_k(t|t-\theta_k)$ for the current time t. And

$$\begin{aligned} &P_{k_1,k_2}^{(\theta_{k_1},\theta_{k_2})}(t) \\ &\triangleq \mathsf{E}\{(\mathbf{x}(t) - \hat{\mathbf{x}}_{k_1}(t|t-\theta_{k_1}))(\mathbf{x}(t) - \hat{\mathbf{x}}_{k_2}(t|t-\theta_{k_2}))^{\mathrm{T}}\} \end{aligned} \tag{36}$$

is the cross estimation error covariance between two estimators $\hat{\mathbf{x}}_{k_1}(t|t-\theta_{k_1})$ and $\hat{\mathbf{x}}_{k_2}(t|t-\theta_{k_2})$. In the following we describe how to compute $P_k^{(\theta_k)}(t)$ and $P_{k_1,k_2}^{(\theta_{k_1},\theta_{k_2})}(t)$, respectively.

Step 1: We consider the *single-sensor-based augmented systems*. There are K such augmented systems. For each k, we consider only sensor k. Specifically, we consider the augmented system evolution matrix $\bar{A}$ in (19), the augmented control matrix $\bar{B}$ in (20), and the augmented state perturbation $\bar{\mathbf{w}}(t)$ in (21) with noise covariance in (23). However, we only have one sensor and thus the corresponding augmented observation matrix is $\bar{C}_k^{(\theta_k)}$ in (24) and the observation noise covariance is simply $\Sigma_{\mathbf{v}_k}$. Note that since there is only one sensor k being considered, there is no need for the vertical stacking described in (26) to (28).

Once the single-sensor augmented system is constructed for the given k value, we use $\bar{A}$, $\bar{B}$, $\Sigma_{\bar{\mathbf{w}}}$, $\bar{C}_k^{(\theta_k)}$, $\Sigma_{\mathbf{v}_k}$ as the input to the single-user zero-delay KF algorithm in Algorithm II.1. We denote the outputs by $\bar{P}_k^{(\theta_k)}(t)$ and $\bar{\Gamma}_k^{(\theta_k)}(t)$, respectively. Note that $\bar{P}_k^{(\theta_k)}(t)$ is of dimension $(1+\delta_{\max})N$-by-$(1+\delta_{\max})N$ and $\bar{\Gamma}_k^{(\theta_k)}(t)$ is of dimension $(1+\delta_{\max})N$-by-M_k.

Step 2: The $2K$ matrices $\bar{\Gamma}_k^{(\theta_k)}(t)$ and $\bar{C}_k^{(\theta_k)}(t)$ plus the augmented state evolution matrix $\bar{A}$ and the state perturbation covariance matrix $\Sigma_{\bar{\mathbf{w}}}$ will be used to compute $2 \cdot C_K^2$ matrices $P_{\mathrm{rcct},k_1,k_2}(t)$ and $\Phi_{\mathrm{rcct},k_1,k_2}(t)$. Both matrices are of dimension $(1+\delta_{\max})N$-by-$(1+\delta_{\max})N$ and each pair $(P_{\mathrm{rcct},k_1,k_2}(t), \Phi_{\mathrm{rcct},k_1,k_2}(t))$ is computed by solving the following Riccati equations.

$$\Phi_{\mathrm{rcct},k_1,k_2}(t) = \bar{A}P_{\mathrm{rcct},k_1,k_2}(t-1)\bar{A}^{\mathrm{T}} + \Sigma_{\bar{\mathbf{w}}}, \tag{37}$$

$$\begin{aligned} P_{\mathrm{rcct},k_1,k_2}(t) = &(\mathbf{I}_{(\delta_{\max}+1)N} - \bar{\Gamma}_{k_1}^{(\theta_{k_1})}(t)\bar{C}_{k_1}^{(\theta_{k_1})}) \\ &\cdot \Phi_{\mathrm{rcct},k_1,k_2}(t) \\ &\cdot (\mathbf{I}_{(\delta_{\max}+1)N} - \bar{\Gamma}_{k_2}^{(\theta_{k_2})}(t)\bar{C}_{k_2}^{(\theta_{k_2})})^{\mathrm{T}}. \end{aligned} \tag{38}$$

The computed matrices $\bar{P}_k^{(\theta_k)}(t)$ and $P_{\mathrm{rcct},k_1,k_2}(t)$ satisfy the following lemma.

Lemma 2. *The N-by-N matrix $P_k^{(\theta_k)}(t)$ defined in* (35) *is the upper-left corner of the* $(\delta_{\max}+1)N$*-by-*$(\delta_{\max}+1)N$

matrix $\bar{P}_k^{(\theta_k)}(t)$. *The* N*-by-*N *matrix* $P_{k_1,k_2}^{(\theta_{k_1},\theta_{k_2})}(t)$ *is the upper-left corner of the* $(\delta_{\max}+1)N$*-by-*$(\delta_{\max}+1)N$ *matrix* $P_{\mathrm{rcct},k_1,k_2}(t)$.

This is basically because the first N dimensions of the augmented state vector $\bar{\mathbf{x}}(t)$ is the original state vector $\mathbf{x}(t)$. Considering only the first N dimensional estimation error in $P_{\mathrm{rcct},k_1,k_2}(t)$ provides the KF estimation error covariance matrix of $\mathbf{x}(t)$. The complete proof is mitigated to the appendix B.

Step 3: The resulting matrices $P_k^{(\theta_k)}(t)$ and $P_{k_1,k_2}^{(\theta_{k_1},\theta_{k_2})}(t)$ in Lemma 2 are then put into a $KN\times KN$ matrix $P_{\mathrm{LQE}}^{(\vec{\theta})}(t)$, for which the (k,k)-th block matrix of size $N\times N$ equals $P_k^{(\theta_k)}(t)$ of sensor k and the (k_1,k_2)-th block matrix of size $N\times N$ represents $P_{k_1,k_2}^{(\theta_{k_1},\theta_{k_2})}(t)$ when $k_1\neq k_2$. We then use $P_{\mathrm{LQE}}^{(\vec{\theta})}(t)$ to generate K different N-by-N weighting matrices $\{W_k^{(\vec{\theta})}(t), k\in[1,K]\}$:

$$\begin{aligned}&[W_1^{(\vec{\theta})}(t)^T, W_2^{(\vec{\theta})}(t)^T,\cdots,W_K^{(\vec{\theta})}(t)^T]^{\mathrm{T}}\\&=(P_{\mathrm{LQE}}^{(\vec{\theta})}(t))^{-1}\cdot\mathbf{e}\cdot(\mathbf{e}^{\mathrm{T}}(P_{\mathrm{LQE}}^{(\vec{\theta})}(t))^{-1}\mathbf{e})^{-1},\end{aligned}\tag{39}$$

where $\mathbf{e}=[I_{N\times N},\cdots,I_{N\times N}]^{\mathrm{T}}$ is of dimension $KN\times N$ and consists of K identity matrices of size $N\times N$. Note that the value of each $W_k^{(\vec{\theta})}(t)$ depends on the entire AoI vector $\vec{\theta}$ (not just θ_k). We thus put the entire vector $\vec{\theta}$ to the superscript. These K weighting matrices $(W_1^{(\vec{\theta})}(t),\cdots,W_K^{(\vec{\theta})}(t))$ are saved and will be used later for the online computation. We repeat this process for all possible instances of the K-dimensional AoI vector $\vec{\theta}=(\theta_1,\cdots,\theta_K)$. Notice that in our simulation, the weighting matrices $\{W_k^{(\vec{\theta})}(t), k\in[1,K]\}$ converge after around $t=30$ time slots. Thus for each instance of AoI vector, we records $\{W_k^{(\vec{\theta})}(t), k\in[1,K]\}$ for $t\in[1,30]$. For $t>30$, we use the converged result $W_k^{(\vec{\theta})}(t)=W_k^{(\vec{\theta})}(30), k\in[1,K]$.

The Online Operation of The LQE Algorithm. During the online execution of LQE, the receiver/controller first observes the AoI vector $\vec{\theta}(t)$ at the current time t. The $\vec{\theta}(t)$ value is then used to retrieve the values of the precomputed weighting matrices $(W_1^{(\vec{\theta})}(t),\cdots,W_K^{(\vec{\theta})}(t))$. We then run Algorithm IV.1.

Algorithm IV.1 LQE with random delays

1: **INPUT:** The current AoI vector $\vec{\theta}(t)$ at time t; the corresponding precomputed matrices $(W_1^{(\vec{\theta})}(t),\cdots,W_K^{(\vec{\theta})}(t))$ where $\vec{\theta}=\vec{\theta}(t)$; and the quantized local estimates $\hat{\mathbf{x}}_k^q(t-\theta_k)$ received from the k-th sensor.

2: Generate the prediction $\hat{\mathbf{x}}_k^q(t|t-\theta_k(t))$ based on $\hat{\mathbf{x}}_k^q(t-\theta_k(t))$ by

$$\begin{aligned}&\hat{\mathbf{x}}_k^q(t|t-\theta_k(t))\\&=A^{\theta_k(t)}\hat{\mathbf{x}}_k^q(t-\theta_k(t))+\sum_{\tau=t-\theta_k(t)}^{t-1}A^{t-1-\tau}\mathbf{u}(\tau).\end{aligned}\tag{40}$$

3: Linearly combine $\hat{\mathbf{x}}_k^q(t|t-\theta_k(t)), k\in[1,K]$ using the precomputed weighting matrices $W_k^{(\vec{\theta})}(t)$:

$$\hat{\mathbf{x}}_{\mathrm{LQE}}(t)=\sum_{k=1}^{K}W_k^{(\vec{\theta})}(t)\hat{\mathbf{x}}_k^q(t|t-\theta_k(t)).\tag{41}$$

4: Set the control action as $\mathbf{u}(t)=-A\hat{\mathbf{x}}_{\mathrm{LQE}}(t)$ and send it to the plant to control the future state value $\mathbf{x}(t+1)$.

2) CQE with quantization error reduction:

In this section, we propose a fusion scheme which takes advantages of both outdated information and the newly received quantized local estimates at time t.

To that end, we first define the following extended state vector

$$\bar{\mathbf{x}}_p(t)\triangleq(\mathbf{x}(t)^{\mathrm{T}},\hat{\mathbf{x}}_1(t)^{\mathrm{T}},\cdots,\hat{\mathbf{x}}_K(t)^{\mathrm{T}},\hat{\mathbf{x}}_1^q(t)^{\mathrm{T}},\cdots,\hat{\mathbf{x}}_K^q(t)^{\mathrm{T}})^{\mathrm{T}}.\tag{48}$$

It consists of the state vector, the state local estimates and their quantized versions.

The evolution equations for $\mathbf{x}(t)$ and $\hat{\mathbf{x}}_k(t)$ are defined in (2) and (14) respectively. We now define the evolution equation for $\hat{\mathbf{x}}_k^q(t)$, which is

$$\hat{\mathbf{x}}_k^q(t)=\hat{\mathbf{x}}_k(t)+\mathbf{q}_k(t),\tag{49}$$

where the quantization noise $\mathbf{q}_k(t)$ is modelled as an i.i.d. Gaussian noise independent to the disturbance $\mathbf{w}(t)$, observation noises $\mathbf{v}_k(t)$, and delays $\Delta_k(t)$. It has the distribution $\mathbf{q}_k(t)\sim\mathcal{N}(\mathbf{0},\Sigma_{\mathbf{q}_k})$, with the covariance matrix being

$$\Sigma_{\mathbf{q}_k}=U_k\frac{\alpha^2}{12}U_k^{\mathrm{T}}.\tag{50}$$

Both α and U_k are quantization lattice parameters introduced in Section IV-A.

In order to design the control action $\mathbf{u}(t)$, we will first generate the MMSE estimate of $\bar{\mathbf{x}}_p(t)$ based on all received information by time t. Then utilize the estimate's first N dimensions, which represents the estimate of $\mathbf{x}(t)$, to design $\mathbf{u}(t)$.

We first construct the evolution equation of $\bar{\mathbf{x}}_p(t)$ based on the individual evolution equations of $\mathbf{x}(t)$, $\hat{\mathbf{x}}_k(t)$ and $\hat{\mathbf{x}}_k^q(t)$. It is established as a time-varying linear control system,

$$\bar{\mathbf{x}}_p(t+1)=\bar{A}_p(t)\bar{\mathbf{x}}_p(t)+\bar{B}_p\mathbf{u}(t)+\bar{D}_p(t)\bar{\mathbf{w}}_p(t),\tag{51}$$

where $\bar{\mathbf{w}}_p(t)$ is an extended noise vector

$$\begin{aligned}\bar{\mathbf{w}}_p(t)=&(\mathbf{w}(t)^{\mathrm{T}},\mathbf{v}_1(t+1)^{\mathrm{T}},\cdots,\mathbf{v}_K(t+1)^{\mathrm{T}},\\&\mathbf{q}_1(t+1)^{\mathrm{T}},\cdots,\mathbf{q}_K(t+1)^{\mathrm{T}})^{\mathrm{T}}.\end{aligned}\tag{52}$$

We now construct the system matrices $\bar{A}_p(t)$, $\bar{B}_p$ and $\bar{D}_p(t)$ for the extended system (51). The basic idea is to ensure that every N rows in (51) is reducible to the original evolution equations of (3), (14) and (49). (51) is a time-varying system because in the evolution equation of $\hat{\mathbf{x}}_k(t)$ and $\hat{\mathbf{x}}_k^q(t)$, the Kalman filter gain matrice $\Gamma_k(t)$ is time varying. In the following elaboration, we use the notation $X_{[n_1:n_2]}$ to represent row n_1 to row n_2 of a matrix X.

Because the first N dimensions of $\bar{\mathbf{x}}_p(t+1)$ represents $\mathbf{x}(t+1)$, the first N rows in (51) represent the evolution equation

Algorithm IV.2 CQE with quantization error reduction

At time t, the controller updates the K $\delta_{\max}$-length buffers so that each one of them keeps all $\mathbf{x}_k^q(t)$s received from that sensor with time stamps $t-\delta_{\max} \le \tau \le t$.

Load the estimate $\mu_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t-\delta_{\max}-1)$ and its estimation error covariance matrix $P_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t-\delta_{\max}-1)$ computed at time $t-1$.

Initialize

$$\begin{aligned}&\mu_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t-\delta_{\max}-1)\\&=\mu_{\bar{\mathbf{x}}_p}^{t-1,t-\delta_{\max}-1}(t-\delta_{\max}-1),\end{aligned} \tag{42}$$

$$\begin{aligned}&P_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t-\delta_{\max}-1)\\&=P_{\bar{\mathbf{x}}_p}^{t-1,t-\delta_{\max}-1}(t-\delta_{\max}-1).\end{aligned} \tag{43}$$

for $\tau = t-\delta_{\max} : t$ **do**

  Compute $\mu_{\bar{\mathbf{x}}_p}^{t,\tau-1}(\tau)$ and $P_{\bar{\mathbf{x}}_p}^{t,\tau-1}(\tau)$:

$$\begin{aligned}\mu_{\bar{\mathbf{x}}_p}^{t,\tau-1}(\tau)=&\bar{A}(\tau-1)\mu_{\bar{\mathbf{x}}_p}^{t,\tau-1}(\tau-1)\\&+\bar{B}\mathbf{u}(\tau-1),\end{aligned} \tag{44}$$

$$\begin{aligned}P_{\bar{\mathbf{x}}_p}^{t,\tau-1}(\tau)=&\bar{A}(\tau-1)P_{\bar{\mathbf{x}}_p}^{t,\tau-1}(\tau-1)\bar{A}(\tau-1)^{\mathrm{T}}\\&+\bar{D}(\tau-1)\Sigma_{\bar{\mathbf{w}}_p}\bar{D}(\tau-1)^{\mathrm{T}}.\end{aligned} \tag{45}$$

  Construct $\bar{\mathbf{x}}_p^{\mathrm{kn},t}(\tau)$ by concatenating all received $\hat{\mathbf{x}}_k^q(\tau)$s.

  Construct $\mu_{\bar{\mathbf{x}}_p^{\mathrm{kn}}}^{t,\tau-1}(\tau)$ and $\mu_{\bar{\mathbf{x}}_p^{\mathrm{ukn}}}^{t,\tau-1}(\tau)$ from $\mu_{\bar{\mathbf{x}}_p}^{t,\tau-1}(\tau)$.

  Construct $P_{\bar{\mathbf{x}}_p^{\mathrm{kn}}}^{t,\tau-1}(\tau)$, $P_{\mathrm{rcct}}^{t,\tau-1}(\tau)$ and $P_{\bar{\mathbf{x}}_p^{\mathrm{ukn}}}^{t,\tau-1}(\tau)$ from $P_{\bar{\mathbf{x}}_p}^{t,\tau-1}(\tau)$.

  Compute $\mu_{\bar{\mathbf{x}}_p^{\mathrm{ukn}}}^{t,\tau}(\tau)$,

$$\begin{aligned}&\mu_{\bar{\mathbf{x}}_p^{\mathrm{ukn}}}^{t,\tau}(\tau)\\=&\mu_{\bar{\mathbf{x}}_p^{\mathrm{ukn}}}^{t,\tau-1}(\tau)+P_{\mathrm{rcct}}^{t,\tau-1}(\tau)\cdot P_{\bar{\mathbf{x}}_p^{\mathrm{kn}}}^{t,\tau-1}(\tau)^{-1}\cdot(\bar{\mathbf{x}}_p^{\mathrm{kn},t}(\tau)\\&-\mu_{\bar{\mathbf{x}}_p^{\mathrm{kn}}}^{t,\tau-1}(\tau)).\end{aligned} \tag{46}$$

  Compute $P_{\bar{\mathbf{x}}_p^{\mathrm{ukn}}}^{t,\tau}(\tau)$:

$$\begin{aligned}&P_{\bar{\mathbf{x}}_p^{\mathrm{ukn}}}^{t,\tau}(\tau)\\=&P_{\bar{\mathbf{x}}_p^{\mathrm{ukn}}}^{t,\tau-1}(\tau)-P_{\mathrm{rcct}}^{t,\tau-1}(\tau)\cdot P_{\bar{\mathbf{x}}_p^{\mathrm{kn}}}^{t,\tau-1}(\tau)^{-1}\cdot\\&P_{\mathrm{rcct}}^{t,\tau-1}(\tau)^{\mathrm{T}}.\end{aligned} \tag{47}$$

  Construct $\mu_{\bar{\mathbf{x}}_p}^{t,\tau}(\tau)$ by reunioning $\mu_{\bar{\mathbf{x}}_p^{\mathrm{ukn}}}^{t,\tau}(\tau)$ and $\mu_{\bar{\mathbf{x}}_p^{\mathrm{kn}}}^{t,\tau}(\tau)$.

  Construct $P_{\bar{\mathbf{x}}_p}^{t,\tau}(\tau)$ by initializing it as a $(2K+1)N\times(2K+1)N$ zero matrix and filling the $N\times N$ blocks from $P_{\bar{\mathbf{x}}_p^{\mathrm{ukn}}}^{t,\tau}(\tau)$ into their corresponding positions in $P_{\bar{\mathbf{x}}_p}^{t,\tau}(\tau)$.

end for

Store $\mu_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}}(t-\delta_{\max})$ and $P_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}}(t-\delta_{\max})$ for next time slot estimation.

Let $\hat{\mathbf{x}}_{\mathrm{CQE}}(t)=[\mu_{\bar{\mathbf{x}}_p}^{t,t}(t)]_{1:N}$.

Set the control action as $\mathbf{u}(t)=-A\hat{\mathbf{x}}_{\mathrm{CQE}}(t)$ and send it back to the plant for controlling $\mathbf{x}(t+1)$.

of $\mathbf{x}(t)$, which is given in (3). We thus define Row 1 to Row N of the system matrices $\bar{A}_p(t)$, $\bar{B}_p$ and $\bar{D}_p(t)$ as following:

$$\bar{A}_p(t)_{[1:N]}=\left[A,\mathbf{0}_{N\times(2KN)}\right], \tag{53}$$

$$\bar{B}_{p[1:N]}=B, \tag{54}$$

$$\bar{D}_p(t)_{[1:N]}=\left[\mathbf{I}_{N\times N},\mathbf{0}_{N\times(\sum_{i=1}^K M_i+KN)}\right]. \tag{55}$$

We can easily verify that

$$\begin{aligned}&\bar{\mathbf{x}}_p(t+1)_{[1:N]}\\&=\bar{A}_p(t)_{[1:N]}\bar{\mathbf{x}}_p(t)+\bar{B}_{p[1:N]}\mathbf{u}(t)+\bar{D}_p(t)_{[1:N]}\bar{\mathbf{w}}_p(t)\end{aligned} \tag{56}$$

is equivalent to (2) by substituting (53), (54) and (55) into (56).

Then we construct row $kN+1$ to row $(k+1)N$ of the system matrices for $k\in[1,K]$. Each N rows from row $kN+1$ to row $(k+1)N$ for $k\in[1,K]$ in (56) represent the evolution equation of $\hat{\mathbf{x}}_k(t)$ for some k. The corresponding N-row block of $\bar{A}_p(t)$, $\bar{B}_p$ and $\bar{D}_p(t)$ are defined as

$$\begin{aligned}&\bar{A}_p(t)_{[kN+1:(k+1)N]}\\&=[\Gamma_k(t+1)C_kA,\mathbf{0}_{N\times(k-1)N},(\mathbf{I}_{N\times N}-\Gamma_k(t+1)C_k)A,\\&\quad\mathbf{0}_{N\times(2K-k)N}],\end{aligned} \tag{57}$$

$$\begin{aligned}&\bar{B}_{p[kN+1:(k+1)N]}\\&=B,\end{aligned} \tag{58}$$

$$\begin{aligned}&\bar{D}_p(t)_{[kN+1:(k+1)N]}\\&=[\Gamma_k(t+1)C_k,\mathbf{0}_{N\times\sum_{i=1}^{k-1}M_i},\Gamma_k(t+1),\\&\quad\mathbf{0}_{N\times(\sum_{i=k+1}^K M_i+KN)}],\end{aligned} \tag{59}$$

for $k\in[1,K]$.

By plugging the constructed matrices, (57), (58) and (59), into (51), one can verify that

$$\begin{aligned}&\bar{\mathbf{x}}_p(t+1)_{[kN+1:(k+1)N]}\\&=\bar{A}_p(t)_{[kN+1:(k+1)N]}\bar{\mathbf{x}}_p(t)+\bar{B}_{p[kN+1:(k+1)N]}\mathbf{u}(t)\\&\quad+\bar{D}_p(t)_{[kN+1:(k+1)N]}\bar{\mathbf{w}}_p(t)\end{aligned} \tag{60}$$

is equivalent to (14).

Row $(k+1)N+1$ to row $(2k+1)N$ in (51) for $k\in[1,K]$ correspond to the evolution equation of $\hat{\mathbf{x}}_k^q(t)$. We construct these N-row blocks in $\bar{A}_p(t)$, $\bar{B}_p$, and $\bar{D}_p(t)$ as following

$$\begin{aligned}&\bar{A}_p(t)_{[(K+k)N+1:(K+k+1)N]}\\&=\bar{A}_p(t)_{[kN+1:(k+1)N]},\end{aligned} \tag{61}$$

$$\begin{aligned}&\bar{B}_{p[(K+k)N+1:(K+k+1)N]}\\&=B,\end{aligned} \tag{62}$$

$$\begin{aligned}&\bar{D}_p(t)_{[(K+k)N+1:(K+k+1)N]}\\&=[\Gamma_k(t+1)C_k,\mathbf{0}_{N\times\sum_{i=1}^{k-1}M_i},\Gamma_k(t+1),\\&\quad\mathbf{0}_{N\times(\sum_{i=k+1}^K M_i+(k-1)N)},\mathbf{I}_{N\times N},\mathbf{0}_{N\times(K-k)N}],\end{aligned} \tag{63}$$

for $k\in[1,K]$.

(63) and (59) are slightly different with (63) having one more $N\times N$ identity matrix used as the coefficient matrix of $\mathbf{q}_k(t)$.

(51) is fully described with $\bar{\mathbf{w}}_p(t)$ being defined and the system matrices being constructed. Then we describe how to generate the MMSE estimate of $\bar{\mathbf{x}}_p(t)$ based on all arrived $\hat{\mathbf{x}}_k^q(\tau)$s by time t. The estimation uses an important property about the sub-components in $\bar{\mathbf{x}}_p(t)$, that is

$$\mathbf{x}(t), \hat{\mathbf{x}}_k(t)\text{s}, \hat{\mathbf{x}}_k^q(t)\text{s are jointly Gaussian random vectors}$$
$$\text{to each other conditioned on all arrived } \hat{\mathbf{x}}_k^q(t)\text{s.}$$

This comes from the fact that the extended noises $\bar{\mathbf{w}}_p(t)$ are i.i.d. Gaussian random vectors, and then the state $\bar{\mathbf{x}}_p(t)$ is a linear combination of Gaussian vectors with the mean shifted by the linear combination of control actions.

There is a useful subset to define before we start the estimation:

$$\mathcal{X}(t, t_1) \triangleq \{\hat{\mathbf{x}}_k^q(\tau), \tau + \Delta_k(\tau) \le t \text{ and } \tau \le t_1, \text{ where } t_1 \le t \text{ and } k \in [1, K]\}. \tag{64}$$

It consists of the arrived $\hat{\mathbf{x}}_k^q(\tau)$s by time t with time stamps $< t_1$ for some $t_1 < t$. This set will be useful when we estimate $\bar{\mathbf{x}}_p(t')$ for some t' where $t_1 < t' < t$. We denote the estimate of $\bar{\mathbf{x}}_p(t')$ conditioned on $\mathcal{X}(t, t_1)$ as

$$\mu_{\bar{\mathbf{x}}_p}^{t,t_1}(t') \triangleq \mathsf{E}\{\bar{\mathbf{x}}_p(t') | \mathcal{X}(t, t_1)\}. \tag{65}$$

The corresponding estimation error covariance matrix is denoted as

$$P_{\bar{\mathbf{x}}_p}^{t,t_1}(t') \triangleq \mathsf{E}\{(\bar{\mathbf{x}}_p(t') - \mu_{\bar{\mathbf{x}}_p}^{t,t_1}(t'))(\bar{\mathbf{x}}_p(t') - \mu_{\bar{\mathbf{x}}_p}^{t,t_1}(t'))^{\mathrm{T}} | \mathcal{X}(t, t_1)\}. \tag{66}$$

The controller uses a $\delta_{\max}$-length buffer for each sensor to keep the arrived $\hat{\mathbf{x}}_k^q(\tau)$s with time stamps $t - \delta_{\max} \le \tau \le t$ from sensor k. Only a $\delta_{\max}$-length buffer is needed because the delays are assumed to be bounded by $\delta_{\max}$, and then no updates for $\hat{\mathbf{x}}_k^q(\tau)$s with $\tau \le t - \delta_{\max} - 1$ happen after time $t - \delta_{\max} - 1$.

Now we start to estimate $\bar{\mathbf{x}}_p(t)$ using all arrivals by time t. The general idea is to generate the MMSE estimate of $\bar{\mathbf{x}}_p(\tau)$ for $\tau \in [t - \delta_{\max}, t]$ conditioning on $\mathcal{X}(t, \tau)$ recursively.

At time t, the controller retrieves the estimation result from time $t-1$: the conditional mean $\mu_{\bar{\mathbf{x}}_p}^{t-1,t-\delta_{\max}-1}(t - \delta_{\max} - 1)$ and the estimation error covariance matrix $P_{\bar{\mathbf{x}}_p}^{t-1,t-\delta_{\max}-1}(t - \delta_{\max} - 1)$ to prepare for the estimation of $\bar{\mathbf{x}}_p(t - \delta_{\max})$. Based on the arrival pattern of $\hat{\mathbf{x}}_k^q(t - \delta_{\max})$ at time t, i.e. $\hat{\mathbf{x}}_k^q(t - \delta_{\max})$ for which sensors have arrived and for which sensors have not, we consider $\bar{\mathbf{x}}_p(t - \delta_{\max})$ in two parts: $\bar{\mathbf{x}}_p^{\text{kn},t}(t - \delta_{\max})$ consists of the arrived quantized local estimates $\hat{\mathbf{x}}_k^q(t - \delta_{\max})$s, where 'kn' stands for 'known'; and $\bar{\mathbf{x}}_p^{\text{ukn},t}(t - \delta_{\max})$ consists of the state variables $\mathbf{x}(t - \delta_{\max})$, $\hat{\mathbf{x}}_k(t - \delta_{\max})$s and not-arrived $\hat{\mathbf{x}}_k^q(t - \delta_{\max})$s, which are unknown by time t, with the 'ukn' stands for 'unknown'. For example, at time $t = 3$ for a system with three sensors, if $\hat{\mathbf{x}}_1^q(2)$ and $\hat{\mathbf{x}}_3^q(2)$ have arrived by $t = 3$ but $\hat{\mathbf{x}}_2^q(2)$ has not, $\bar{\mathbf{x}}_p^{\text{ukn},3}(2)$ and $\bar{\mathbf{x}}_p^{\text{kn},3}(2)$ are respectively:

$$\bar{\mathbf{x}}_p^{\text{ukn},3}(2) = (\mathbf{x}(2), \hat{\mathbf{x}}_1(2), \hat{\mathbf{x}}_2(2), \hat{\mathbf{x}}_3(2), \hat{\mathbf{x}}_2^q(2)), \tag{67}$$

$$\bar{\mathbf{x}}_p^{\text{kn},3}(2) = (\hat{\mathbf{x}}_1^q(2), \hat{\mathbf{x}}_3^q(2)). \tag{68}$$

To adapt the notations of the conditional mean (65) and the estimation error covariance (66) for $\bar{\mathbf{x}}_p^{\text{ukn},t}(t - \delta_{\max})$ and $\bar{\mathbf{x}}_p^{\text{kn},t}(t - \delta_{\max})$, we will replace the subscript $\bar{\mathbf{x}}_p$ in (65) and (66) by $\bar{\mathbf{x}}_p^{\text{ukn}}$ and $\bar{\mathbf{x}}_p^{\text{kn}}$ respectively.

We first initialize the estimate for $\bar{\mathbf{x}}_p(t - \delta_{\max})$ at time t. The estimation for $\bar{\mathbf{x}}_p(t - \delta_{\max} - 1)$ based on the arrived information by time t is

$$\mu_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t - \delta_{\max} - 1) = \mu_{\bar{\mathbf{x}}_p}^{t-1,t-\delta_{\max}-1}(t - \delta_{\max} - 1) \tag{69}$$

$$P_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t - \delta_{\max} - 1) = P_{\bar{\mathbf{x}}_p}^{t-1,t-\delta_{\max}-1}(t - \delta_{\max} - 1), \tag{70}$$

because

$$\mathcal{X}(t, t - \delta_{\max} - 1) = \mathcal{X}(t-1, t - \delta_{\max} - 1). \tag{71}$$

By applying a one-time-slot prediction to (69) and (70), we have

$$\mu_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t - \delta_{\max}) = \bar{A}_p(t - \delta_{\max} - 1)\mu_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t - \delta_{\max} - 1) + \bar{B}_p \mathbf{u}(t - \delta_{\max} - 1), \tag{72}$$

$$P_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t - \delta_{\max}) = \bar{A}_p(t - \delta_{\max} - 1) P_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t - \delta_{\max} - 1) \bar{A}_p(t - \delta_{\max} - 1)^{\mathrm{T}} + \Sigma_{\bar{\mathbf{w}}_p}. \tag{73}$$

Then we re-organize the predictive information (72) and (73) according to the arrival pattern of $\hat{\mathbf{x}}_k^q(t - \delta_{\max})$s by time t. $\mu_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t - \delta_{\max})$ is separated into $\mu_{\bar{\mathbf{x}}_p^{\text{kn}}}^{t,t-\delta_{\max}-1}(t - \delta_{\max})$ and $\mu_{\bar{\mathbf{x}}_p^{\text{ukn}}}^{t,t-\delta_{\max}-1}(t - \delta_{\max})$ so that $\mu_{\bar{\mathbf{x}}_p^{\text{kn}}}^{t,t-\delta_{\max}-1}(t - \delta_{\max})$ consists of the same state variables appeared in $\bar{\mathbf{x}}_p^{\text{kn},t}(t - \delta_{\max})$, and $\mu_{\bar{\mathbf{x}}_p^{\text{ukn}}}^{t,t-\delta_{\max}-1}(t - \delta_{\max})$ consists of the same state variables appeared in $\bar{\mathbf{x}}_p^{\text{ukn},t}(t - \delta_{\max})$. In the same arrival example for (67) and (68), we have

$$\mu_{\bar{\mathbf{x}}_p^{\text{ukn}}}^{t,t-\delta_{\max}-1}(t - \delta_{\max}) = (\mu_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t - \delta_{\max})_{[1,4N]}, \mu_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t - \delta_{\max})_{[5N+1,6N]}) \tag{74}$$

$$\mu_{\bar{\mathbf{x}}_p^{\text{kn}}}^{t,t-\delta_{\max}-1}(t - \delta_{\max}) = (\mu_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t - \delta_{\max})_{[4N+1,5N]}, \mu_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t - \delta_{\max})_{[6N+1,7N]}). \tag{75}$$

Based on the arrival pattern of $\hat{\mathbf{x}}_k^q(t - \delta_{\max})$s, $P_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t - \delta_{\max})$ is re-arranged into three sub-matrices. We first divide $P_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t - \delta_{\max})$ into $(2K+1)^2$ block matrices of size $N \times N$. Each of these block matrices represents an estimation error covariance matrix of one sensor or a cross estimation error covariance matrix between two sensors. Then by selecting the sub-matrices of size $N \times N$ according to their physical meanings, $P_{\bar{\mathbf{x}}_p}^{t,t-\delta_{\max}-1}(t - \delta_{\max})$ is divided into three sub-matrices: $P_{\bar{\mathbf{x}}_p^{\text{ukn}}}^{t,t-\delta_{\max}-1}(t - \delta_{\max})$ represents the covariance matrix of the estimation error for the unknown state variables by time t, $\bar{\mathbf{x}}_p^{\text{ukn},t}(t - \delta_{\max}) - \mu_{\bar{\mathbf{x}}_p^{\text{ukn}}}^{t,t-\delta_{\max}-1}(t - \delta_{\max})$;

$P^{t,t-\delta_{\max}-1}_{\bar{\mathbf{x}}^{\text{kn}}_p}(t-\delta_{\max})$ represents the covariance matrix of the estimation error for the arrived quantized local estimates by time t, $\bar{\mathbf{x}}^{\text{kn},t}_p(t-\delta_{\max}) - \mu^{t,t-\delta_{\max}-1}_{\bar{\mathbf{x}}^{\text{kn}}_p}(t-\delta_{\max})$; the third matrix, denoted as $P^{t,t-\delta_{\max}-1}_{\text{rcct}}(t-\delta_{\max})$, represents the cross estimation error covariance matrix between the unknown state variables and the arrived quantized local estimates by time t.

$$\begin{aligned}&P^{t,t-\delta_{\max}-1}_{\text{rcct}}(t-\delta_{\max})\\&\triangleq \mathsf{E}\{(\bar{\mathbf{x}}^{\text{ukn},t}_p(t-\delta_{\max}) - \mu^{t,t-\delta_{\max}-1}_{\bar{\mathbf{x}}^{\text{ukn}}_p}(t-\delta_{\max}))\\&(\bar{\mathbf{x}}^{\text{kn},t}_p(t-\delta_{\max}) - \mu^{t,t-\delta_{\max}-1}_{\bar{\mathbf{x}}^{\text{kn}}_p}(t-\delta_{\max}))^{\mathrm{T}}\\&|\mathcal{X}(t,t-\delta_{\max}-1)\}. \end{aligned} \tag{76}$$

A quick example for constructing the three sub-matrices for the same setting used in (67) and (68) is: the sub-matrix of size $N\times N$ on the first row and second column of $P^{t,t-\delta_{\max}-1}_{\text{rcct}}(t-\delta_{\max})$ represents the cross estimation error covariance of $\mathbf{x}(2)$ and $\hat{\mathbf{x}}^q_3(2)$. It is thus the sub-matrix of size $N\times N$ on the first row and seventh column in $P^{t,t-\delta_{\max}-1}_{\bar{\mathbf{x}}_p}(t-\delta_{\max})$.

Using the pieces we have constructed, the controller then generates the MMSE estimate of $\bar{\mathbf{x}}^{\text{ukn},t}_p(t-\delta_{\max})$ conditioning on $\mathcal{X}(t,t-\delta_{\max})$,

$$\begin{aligned}&\mu^{t,t-\delta_{\max}}_{\bar{\mathbf{x}}^{\text{ukn}}_p}(t-\delta_{\max})\\=&\mu^{t,t-\delta_{\max}-1}_{\bar{\mathbf{x}}^{\text{ukn}}_p}(t-\delta_{\max}) + P^{t,t-\delta_{\max}-1}_{\text{rcct}}(t-\delta_{\max})\cdot\\&P^{t,t-\delta_{\max}-1}_{\bar{\mathbf{x}}^{\text{kn}}_p}(t-\delta_{\max})^{-1}(\bar{\mathbf{x}}^{\text{kn},t}_p(t-\delta_{\max})\\&-\mu^{t,t-\delta_{\max}-1}_{\bar{\mathbf{x}}^{\text{kn}}_p}(t-\delta_{\max})),\end{aligned} \tag{77}$$

and the estimation error covariance matrix of $\bar{\mathbf{x}}^{\text{ukn},t}_p(t-\delta_{\max})$

$$\begin{aligned}&P^{t,t-\delta_{\max}}_{\bar{\mathbf{x}}^{\text{ukn}}_p}(t-\delta_{\max})\\=&P^{t,t-\delta_{\max}-1}_{\bar{\mathbf{x}}^{\text{ukn}}_p}(t-\delta_{\max}) - P^{t,t-\delta_{\max}-1}_{\text{rcct}}(t-\delta_{\max})\cdot\\&P^{t,t-\delta_{\max}-1}_{\bar{\mathbf{x}}^{\text{kn}}_p}(t-\delta_{\max})^{-1}\cdot P^{t,t-\delta_{\max}-1}_{\text{rcct}}(t-\delta_{\max})^{\mathrm{T}}.\end{aligned} \tag{78}$$

Because the sub-components in $\bar{\mathbf{x}}^{\text{kn},t}_p(t-\delta_{\max})$ have arrived by time t, the counterparts for $\bar{\mathbf{x}}^{\text{kn}}_p(t)$ are

$$\mu^{t,t-\delta_{\max}}_{\bar{\mathbf{x}}^{\text{kn}}_p}(t-\delta_{\max}) = \bar{\mathbf{x}}^{\text{kn},t}_p(t-\delta_{\max}), \tag{79}$$

$$P^{t,t-\delta_{\max}}_{\bar{\mathbf{x}}^{\text{kn}}_p}(t-\delta_{\max}) = \mathbf{0}, \tag{80}$$

$$P^{t,t-\delta_{\max}}_{\text{rcct}}(t-\delta_{\max}) = \mathbf{0}. \tag{81}$$

By uniting (77) and (79), we obtain $\mu^{t,t-\delta_{\max}}_{\bar{\mathbf{x}}_p}(t-\delta_{\max})$. Specifically, each state variable in $\mu^{t,t-\delta_{\max}}_{\bar{\mathbf{x}}^{\text{ukn}}_p}(t-\delta_{\max})$ and $\mu^{t,t-\delta_{\max}}_{\bar{\mathbf{x}}^{\text{kn}}_p}(t-\delta_{\max})$ are input into its defined position in $\mu^{t,t-\delta_{\max}}_{\bar{\mathbf{x}}_p}(t-\delta_{\max})$. Then we construct $P^{t,t-\delta_{\max}}_{\bar{\mathbf{x}}_p}(t-\delta_{\max})$ by first initializing it as a zero matrix of dimension $(2K+1)N\times(2K+1)N$. Then we split $P^{t,t-\delta_{\max}}_{\bar{\mathbf{x}}^{\text{ukn}}_p}(t-\delta_{\max})$ into sub-matrices of size $N\times N$ and fill these sub-matrices into $P^{t,t-\delta_{\max}}_{\bar{\mathbf{x}}_p}(t-\delta_{\max})$. To demonstrate the construction of $P^{t,t-\delta_{\max}}_{\bar{\mathbf{x}}_p}(t-\delta_{\max})$, we use the same setting as for (67) and (68). The sub-matrix of size $N\times N$ on the fifth row and fifth column in $P^{t,t-\delta_{\max}}_{\bar{\mathbf{x}}^{\text{ukn}}_p}(t-\delta_{\max})$ represents the estimation error covariance matrix of $\hat{\mathbf{x}}^q_2(t-\delta_{\max})$ conditioning on $\mathcal{X}(t,t-\delta_{\max})$. It is thus put into the sixth row and sixth column of a $N\times N$-size block in $P^{t,t-\delta_{\max}}_{\bar{\mathbf{x}}_p}(t-\delta_{\max})$.

We have finished generating the estimate $\mu^{t,t-\delta_{\max}}_{\bar{\mathbf{x}}_p}(t-\delta_{\max})$ and the estimation error covariance matrix $P^{t,t-\delta_{\max}}_{\bar{\mathbf{x}}_p}(t-\delta_{\max})$. These two pieces of information are then used to estimate $\bar{\mathbf{x}}_p(t-\delta_{\max}+1)$ using the same procedure for estimating $\bar{\mathbf{x}}_p(t-\delta_{\max})$.

This recursive procedure stops when $\mu^{t,t}_{\bar{\mathbf{x}}_p}(t)$ and $P^{t,t}_{\bar{\mathbf{x}}_p}(t)$ are reached. $\mu^{t,t}_{\bar{\mathbf{x}}_p}(t)$ is then used to design $\mathbf{u}(t)$. We summarize the alogirhtm formally in Algorithm IV.2.

V. Experiment

We implemented Algorithm IV.1 and Algorithm IV.2 on a feedback Gaussian linear control system with random delays in Matlab. The system plant is associated with three sensors. The system matrices are $A = \begin{matrix}1.3&0.5&0&0&0\\0&1.1&0.5&0&0\\0&0&0.9&0.5&0\\0&0&0&1.6&0.5\\0&0&0&0&1.5\end{matrix}$, $C_1 = \begin{matrix}1&0&1&0&0\\0&1&0&0&1\end{matrix}$, $C_2 = \begin{matrix}1&1&0&0&0\\0&1&1&0&0\end{matrix}$, $C_3 = \begin{matrix}1&0&0&0.5&0\\0&1&1&0&0.8\end{matrix}$. The system disturbance $\mathbf{w}(t)$ is i.i.d. Gaussian with distribution $\mathcal{N}(\mathbf{0},\mathbf{I}_{N\times N})$, and the observation noises $\mathbf{v}_k(t)$ are also i.i.d., Gaussians with distribution $\mathcal{N}(\mathbf{0},\mathbf{I}_{M_k\times M_k})$. The bounded delays are set as $\Delta_1(t)\in[0,5]$, $\Delta_2(t)\in[0,6]$, and $\Delta_3(t)\in[0,7]$.

Figure 4 compares the Monte Carlo simulation results for Algorithm IV.1 (the blue curve) and Algorithm IV.2 (the red curve) applied on the given system. Figure 5 compares the Monte Carlo simulation results of Algorithm IV.2 (the red curve) and a modified version of Algorithm IV.2 (the green curve), where the quantization effect is ignored, and $\hat{\mathbf{x}}^q_k(t), k\in[1,K]$ are omitted from $\bar{\mathbf{x}}_p(t)$, $\hat{\mathbf{x}}_k(t)$ is set as $\hat{\mathbf{x}}_k(t) = \hat{\mathbf{x}}^q_k(t)$. The orange vertical line in Figure 4 indicates $\mathsf{av.D}_{\min}$. The other vertical lines in Figure 4 and Figure 5 indicate the analytical smallest achievable L2 norms by each particular fusion algorithm being applied.

Each Monte Carlo simulation performance curve consists of 20 data points. Each of these points represents a rate-cost data for a particular quantization cell size. From left to right on each curve, the quantization cell scaling factor α increases from 0.01 to 3.75 with equal distance (≈ 0.2 between two consecutive values). To obtain one rate-cost data on the curve, we simulate the system ((2) and (3)) for 35 time slots. For each time slot, this process is repeated 10^5 times with $\mathbf{w}(t)$, $\mathbf{v}_k(t)$ and $\delta_k(t)$ generated randomly. In Figure 4, the vertical line for $\mathsf{av.D}_{\min}$, which is the smallest attainable L2 state norm that no scheme can bypass, is generated based on the result in Proposition 1. The other vertical lines indicate the analytical L2 state norm each algorithm can achieve when negligible quantization effect exists. They are computed by taking the average of the converged estimation error covariance matrices $P^{35,35}_{\bar{\mathbf{x}}_p}(35)$ of the 10^5 samples when $\alpha = 0.01$, then taking the upper left sub-matrix of size $N\times N$ of it and computing the trace.

Comparing the results in Figure 4, we see that CQE with noise reduction achieves a smaller analytical L2 state norm

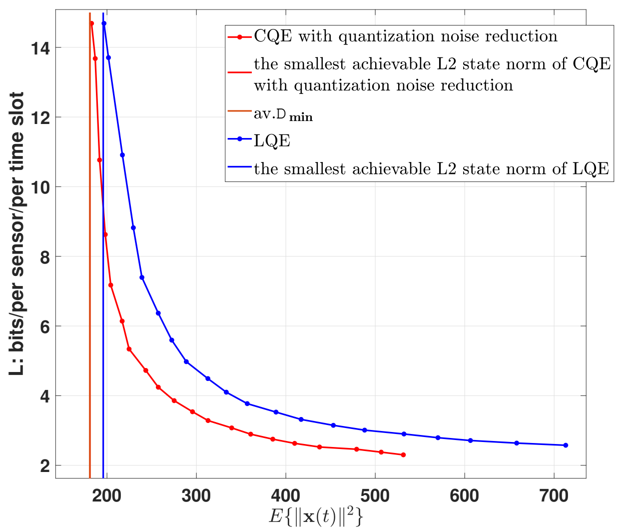

Fig. 4. Scheme performances for CQE with quantization noise reduction and LQE

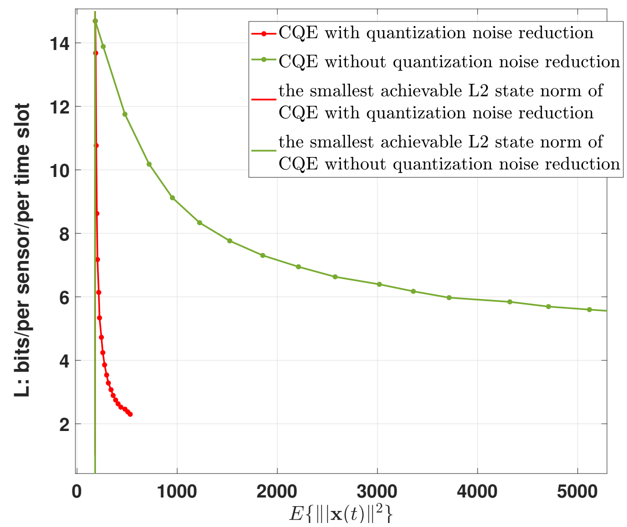

Fig. 5. Scheme performances for CQE with quantization noise reduction and CQE without quantization noise reduction

(180.9) than LQE does (195.8). The analytical best performance (180.9) of CQE with noise reduction is also consistent with $\mathsf{av.D}_{\min}$ for the intrinsic system. Moreover, comparing the two performance curves horizontally on Figure 4, with the same number of bits per sensor per time slot, CQE with quantization noise reduction achieves a L2 state norm 15% smaller than LQE. With decreasing the number of bits per sensor per time slot, CQE with quantization noise reduction out-performs LQE even more significantly. This result explicitly implies that the outdated information is useful to improve the fusion performance in the multi-sensor scenario. Both the performances of Algorithm IV.1 and Algorithm IV.2 decline when less average bits are applied for each time slot each sensor. As shown in Figure IV.1, $\mathsf{av.D}_{\min}$ is achieved by CQE with quantization noise reduction when negligible quantization effect is applied. It shows that our analytical result, $\mathsf{av.D}_{\min}$, is tight.

Figure 5 illustrates the effectiveness of modeling the quantization noise as i.i.d. Gaussian noises. By comparing the simulation result of CQE without quantization noise reduction (the green curve) and the simulation result of CQE with quantization noise reduction (the red curve), one can see that considering the quantization error improves the fusion performance significantly. The assumption of i.i.d. Gaussian quantization noise is effective because rectangle lattice quantization is applied to the Gaussian distributed quantization source $\hat{\mathbf{x}}_k(t)$.

VI. Conclusions

In this work, we proposed the smallest attainable L2 state norm, $\mathsf{av.D}_{\min}$, for Gaussian linear control systems with multiple sensors and bounded random delays. The $\mathsf{av.D}_{\min}$ serves as the benchmark for evaluating the performance of any achievability scheme. We designed two achievability schemes, where in QLE only the freshest arrivals are used for fusion, and in CQE with quantization noise reduction the complete historty of arrivals are used. The experiment demonstrates that using outdated information in the multi-sensor scenario improves the fusion performance significantly compared to using only the freshest information. For future work, it is an attractive direction to complete the analytical lower bound by deriving the rate-stability trade-off curve.

References

[1] X. Ge, F. Yang, and Q.-L. Han, "Distributed networked control systems: A brief overview," *Information Sciences*, vol. 380, pp. 117–131, 2017.

[2] X.-M. Zhang, Q.-L. Han, X. Ge, D. Ding, L. Ding, D. Yue, and C. Peng, "Networked control systems: a survey of trends and techniques," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 1, pp. 1–17, 2019.

[3] F.-Y. Wang and D. Liu, "Networked control systems," *Theory and Applications*, 2008.

[4] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE Journal on Selected Areas in Communications*, 2021.

[5] M. Klügel, M. H. Mamduhi, S. Hirche, and W. Kellerer, "Aoi-penalty minimization for networked control systems with packet loss," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 189–196.

[6] O. Ayan, M. Vilgelm, M. Klügel, S. Hirche, and W. Kellerer, "Age-of-information vs. value-of-information scheduling for cellular networked control systems," in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, 2019, pp. 109–117.

[7] G. N. Nair, F. Fagnani, S. Zampieri, and R. J. Evans, "Feedback control under data rate constraints: An overview," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 108–137, 2007.

[8] V. Kostina and B. Hassibi, "Rate-cost tradeoffs in control," *IEEE Transactions on Automatic Control*, vol. 64, no. 11, pp. 4525–4540, 2019.

[9] J. Zhang and C.-C. Wang, "On the rate-cost of gaussian linear control systems with random communication delays," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 2441–2445.

[10] K. Tsumura and J. Maciejowski, "Stabilizability of siso control systems under constraints of channel capacities," in *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, vol. 1. IEEE, 2003, pp. 193–198.
[11] S. Tatikonda, A. Sahai, and S. Mitter, "Stochastic linear control over a communication channel," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1549–1561, 2004.
[12] A. Sahai and S. Mitter, "The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link. part ii: vector systems,"," *IEEE Trans. Inform. Theory*, 2005.
[13] S.-L. Sun and Z.-L. Deng, "Multi-sensor optimal information fusion kalman filter," *Automatica*, vol. 40, no. 6, pp. 1017–1023, 2004.
[14] S. Sun, H. Lin, J. Ma, and X. Li, "Multi-sensor distributed fusion estimation with applications in networked systems: A review paper," *Information Fusion*, vol. 38, pp. 122–134, 2017.
[15] K.-C. Chang, R. K. Saha, and Y. Bar-Shalom, "On optimal track-to-track fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 4, pp. 1271–1276, 1997.
[16] Y. Bar-Shalom, "On the track-to-track correlation problem," *IEEE Transactions on Automatic control*, vol. 26, no. 2, pp. 571–572, 1981.
[17] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proceedings of the 1997 American Control Conference (Cat. No. 97CH36041)*, vol. 4. IEEE, 1997, pp. 2369–2373.
[18] L. Schenato, "Optimal sensor fusion for distributed sensors subject to random delay and packet loss," in *2007 46th IEEE Conference on Decision and Control*. IEEE, 2007, pp. 1547–1552.
[19] J. Hu, Z. Wang, D. Chen, and F. E. Alsaadi, "Estimation, filtering and fusion for networked systems with network-induced phenomena: new progress and prospects," *Information Fusion*, vol. 31, pp. 65–75, 2016.
[20] B. Chen, W.-A. Zhang, and L. Yu, "Distributed fusion estimation with missing measurements, random transmission delays and packet dropouts," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1961–1967, 2014.
[21] M. Aeberhard, S. Schlichtharle, N. Kaempchen, and T. Bertram, "Track-to-track fusion with asynchronous sensors using information matrix fusion for surround environment perception," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1717–1726, 2012.
[22] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Transactions on Information Theory*, vol. 63, no. 11, pp. 7492–7508, 2017.
[23] V. Kostina and B. Hassibi, "Rate-cost tradeoffs in control. part ii: achievable scheme," *arXiv preprint arXiv:1612.02128*, 2016.
[24] H. Li and Z. Han, "Distributed source coding for controlling cyber physical systems with application in smart grids," in *2014 IEEE Global Communications Conference*. IEEE, 2014, pp. 1516–1521.

APPENDIX

A. Proof for Lemma 1

Proof. By definition, we know that

$$\{\bar{\mathbf{y}}^{(\vec{\theta})}(\tau), \tau \in [0,t]\} = \{\mathbf{y}_k^{(\theta_k)}(\tau), \tau \in [0,t], k \in [1,K]\}. \tag{82}$$

Thus, we re-write the MMSE estimator of $\bar{\mathbf{x}}(t)$

$$\begin{aligned}&\mathsf{E}\{\bar{\mathbf{x}}(t)|\bar{\mathbf{y}}^{(\vec{\theta})}(\tau), \tau \in [0,t]\}\\ &= \mathsf{E}\{\bar{\mathbf{x}}(t)|\mathbf{y}_k^{(\theta_k)}(\tau), \tau \in [0,t], k \in [1,K]\}.\end{aligned} \tag{83}$$

Because the first N dimensions of $\bar{\mathbf{x}}(t)$ is $\mathbf{x}(t)$, the first N dimensions of (83) give the MMSE estimator of $\mathbf{x}(t)$. It is denoted as $\hat{\mathbf{x}}^{(\vec{\theta})}(t)$,

$$\hat{\mathbf{x}}^{(\vec{\theta})}(t) \triangleq \mathsf{E}\{\mathbf{x}(t)|\mathbf{y}_k^{(\theta_k)}(\tau), \tau \in [0,t], k \in [1,K]\}. \tag{84}$$

Using the certainty equivalence law, the optimal control action is

$$\mathbf{u}^*(t) = -A\hat{\mathbf{x}}^{(\vec{\theta})}(t). \tag{85}$$

By applying $\mathbf{u}^*(t)$, the average L2 state norm $E\{\|\mathbf{x}(t+1)\|^2\}$ can be bounded from above

$$\begin{aligned}&\mathsf{E}\{\|\mathbf{x}(t+1)\|^2\}\\ &= \mathsf{E}\{\|A\mathbf{x}(t) + \mathbf{u}(t) + \mathbf{w}(t)\|^2\} &(86)\\ &= \mathsf{E}\{\|A\mathbf{x}(t) + \mathbf{u}(t)\|^2\} + \mathrm{tr}(\Sigma_{\mathbf{w}}) &(87)\\ &\le \mathsf{E}\{\|A\mathbf{x}(t) - A\hat{\mathbf{x}}^{(\vec{\theta})}(t)\|^2\} + \mathrm{tr}(\Sigma_{\mathbf{w}}) &(88)\\ &= \mathrm{tr}(AP^{(\vec{\theta})}(t)A^{\mathrm{T}}) + \mathrm{tr}(\Sigma_{\mathbf{w}}). &(89)\end{aligned}$$

(87) is because of the independence of $\mathbf{w}(t)$. After applying (85), we have (88). And by re-writing the first term in (88) using the KF estimation error covariance matrix $P^{(\vec{\theta})}(t)$, we obtain (89).

This completes the proof for Lemma 1. □

Averaging $\mathsf{D}_{\min}^{(\vec{\theta})}$ over all possible AoI vectors, we obtain $\mathsf{av.D}_{\min}$.

B. Proof for Lemma 2

Proof. By definition, we know that

$$\begin{aligned}&\bar{P}_k^{(\theta_k)}(t)\\ =&\mathsf{E}\{(\bar{\mathbf{x}}(t) - \mathsf{E}\{\bar{\mathbf{x}}(t)|\mathbf{y}_k^{(\theta_k)}(\tau), \tau \in [0,t]\})\\ &(\bar{\mathbf{x}}(t) - \mathsf{E}\{\bar{\mathbf{x}}(t)|\mathbf{y}_k^{(\theta_k)}(\tau), \tau \in [0,t]\})^{\mathrm{T}}\}.\end{aligned} \tag{90}$$

The first N dimensions of $\mathsf{E}\{\bar{\mathbf{x}}(t)|\mathbf{y}_k^{(\theta_k)}(\tau), \tau \in [0,t]\}$ gives $\mathsf{E}\{\mathbf{x}(t)|\mathbf{y}_k(\tau), \tau \in [0, t-\theta_k]\}$, because

$$\{\mathbf{y}_k^{(\theta_k)}(\tau), \tau \in [0,t]\} = \{\mathbf{y}_k(\tau), \tau \in [0, t-\theta_k]\}. \tag{91}$$

Therefore, considering the estimation error covariance matrix of the first N dimensions of $\bar{\mathbf{x}}(t)$, which is the $N \times N$ upper-left corner of $\bar{P}_k^{(\theta_k)}(t)$, gives $P_k^{(\theta_k)}(t)$.

Similarly, by definition we know that

$$\begin{aligned}&\bar{P}_{\mathrm{rcct},k_1,k_2}(t)\\ =&\mathsf{E}\{(\bar{\mathbf{x}}(t) - \mathsf{E}\{\bar{\mathbf{x}}(t)|\mathbf{y}_{k_1}^{(\theta_{k_1})}(\tau), \tau \in [0,t]\})\\ &(\bar{\mathbf{x}}(t) - \mathsf{E}\{\bar{\mathbf{x}}(t)|\mathbf{y}_{k_2}^{(\theta_{k_2})}(\tau), \tau \in [0,t]\})^{\mathrm{T}}\}.\end{aligned} \tag{92}$$

Because the first N dimensions of $\mathsf{E}\{\bar{\mathbf{x}}(t)|\mathbf{y}_{k_1}^{(\theta_{k_1})}(\tau), \tau \in [0,t]\}$ gives

$$\mathsf{E}\{\mathbf{x}(t)|\mathbf{y}_{k_1}(\tau), \tau \in [0, t-\theta_{k_1}]\}, \tag{93}$$

and the first N dimensions of $\mathsf{E}\{\bar{\mathbf{x}}(t)|\mathbf{y}_{k_2}^{(\theta_{k_2})}(\tau), \tau \in [0,t]\}$ gives

$$\mathsf{E}\{\mathbf{x}(t)|\mathbf{y}_{k_2}(\tau), \tau \in [0, t-\theta_{k_2}]\}, \tag{94}$$

by considering the cross estimation error covariance matrix of the first N dimensions in $\bar{\mathbf{x}}(t)$ for sensor k_1 and sensor k_2, we obtain $P_{k_1,k_2}^{(\theta_{k_1},\theta_{k_2})}(t)$, which is the $N \times N$ upper-left corner of $\bar{P}_{\mathrm{rcct},k_1,k_2}(t)$. □