SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

State Estimation with Protecting Exogenous Inputs via Cramér-Rao Lower Bound Approach

Liping GUO[1], Jimin WANG[2], Yanlong ZHAO[1,3] & Ji-Feng ZHANG[4,1*]

[1]*Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China;*
[2]*School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China;*
[3]*School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China;*
[4]*School of Automation and Electrical Engineering, Zhongyuan University of Technology, Zheng Zhou 450007, China*

Abstract This paper addresses the real-time state estimation problem for dynamic systems while protecting exogenous inputs against adversaries, who may be honest-but-curious third parties or external eavesdroppers. The Cramér-Rao lower bound (CRLB) is employed to constrain the mean square error (MSE) of the adversary's estimate for the exogenous inputs above a specified threshold. By minimizing the MSE of the state estimate while ensuring a certain privacy level measured by CRLB, the problem is formulated as a constrained optimization. To solve the optimization problem, an explicit expression for CRLB is first provided. As the computational complexity of the CRLB increases with the time step, a low-complexity approach is proposed to make the complexity independent of time. Then, a relaxation approach is proposed to efficiently solve the optimization problem. Finally, a privacy-preserving state estimation algorithm with low complexity is developed, which also ensures (ϵ, δ)-differential privacy. Two illustrative examples, including a practical scenario for protecting building occupancy, demonstrate the effectiveness of the proposed algorithm.

Keywords Privacy preservation, state estimation, non-convex optimization, exogenous inputs, Cramér-Rao lower bound

Citation State Estimation with Protecting Exogenous Inputs via Cramér-Rao Lower Bound Approach. Sci China Inf Sci, for review

1 Introduction

An increasing array of applications necessitates users to send private data streams to a third party for signal processing and decision-making [1, 2]. Due to privacy concerns, users prefer to send information while protecting their sensitive data. As a result, privacy issues have gained increased attention in several research fields, such as mobile robotic systems [3], network systems [4, 5], and control systems [6–8].

In recent years, privacy has been widely studied in various fields, e.g., machine learning [9–11], Nash equilibrium [12, 13], decentralized optimization [14, 15], and real-time state estimation [1, 16, 17]. In practice, the state estimates or measurements could be acquired by honest-but-curious third parties or external eavesdroppers, leading to a privacy leakage [16–19]. For instance, in intelligent transportation systems, users are often required to send measurements to a third party for monitoring or control purposes, potentially compromising their privacy [1]. Besides, in environmental monitoring, the building occupancy could be reliably estimated from the CO_2 levels [17, 18]. Similarly, the thermal dynamics of a building may also result in a privacy leakage of occupancy [19]. Therefore, the privacy preservation problem is an important issue worth studying in real-time state estimation.

Among various privacy techniques, the most researched are encryption [15,20,21], information-theoretic approach [17, 22], and differential privacy [23, 24]. Due to its ease of implementation, the differential privacy stands out compared to its competitors. Besides, due to its resilience to post-processing, differential privacy makes reverse engineering of the private datasets difficult, and thus, has been widely adopted to protect privacy in stochastic aggregative games [25], distributed optimization [26], and real-time state estimation [1, 23, 27]. Especially for differentially private state estimation, the Kalman filtering problem with input and output perturbation mechanisms has been firstly addressed in [1]. This work was extended to multi-input multi-output systems, which broadens the applicability to multiple sensors for monitoring

[*] Corresponding author (email: jif@iss.ac.cn)

an environment [28]. Then, a two-stage architecture was proposed in [29] to overcome the drawback of output perturbation mechanisms.

Another common approach for privacy preservation is using metrics from information theory to measure private information leakage in response to a query on private datasets [30]. An information-theoretic measure of privacy often relies on the mutual information, which measures private information leakage by formulating the privacy problem as a generalized rate-distortion problem [31]. In addition, mutual information, commonly used to measure the correlation between two random variables, also appears in content related to real-time state estimation problems. For instance, an information-theoretic framework for the privacy-aware optimal state estimation was proposed in [17], where the private dataset was modeled as a first-order Markov process. However, as analyzed convincingly in [32], most mutual-information-based results lack an intuitive or interpretable bound on the statistics of the estimation error by an adversary. To address this limitation, a data-privacy approach was introduced in [33] by measuring the adversary's estimation error with the absolute error metric. However, the absolute error metric is generally mathematically challenging to handle. Consequently, the Fisher information matrix and the Cramér-Rao lower bound (CRLB) have been proposed as alternative frameworks [22,32,34,35]. Although successful in treating control problems [35], Fisher information matrix does not directly capture the performance of an adversary's estimation accuracy. Therefore, the CRLB, based on mean square error (MSE), is more widely adopted in practice [22,32,34]. However, existing studies [22,32,34] have focused on protecting parameters that are time-invariant only. In contrast, when addressing the protection of time-varying states in stochastic dynamic systems, the computational complexity continues to increase as time progresses, thereby posing significant challenges. As far as we know, research on CRLB-based privacy preservation is still lacking in real-time state estimation.

Motivated by the above analysis, in this paper, we investigate the real-time state estimation problem with protecting exogenous inputs via CRLB. The merit of adopting CRLB lies in that it constrains the MSE of the adversary's estimate for the exogenous inputs above a specified threshold. However, there exist some substantial difficulties in studying this problem. First, a primary issue is how to ensure privacy level and state estimation accuracy simultaneously? To do so, we face with solving a non-convex constrained optimization problem, which is challenging. Second, in CRLB-based privacy preservation, calculating CRLB is necessary, but an explicit expression for CRLB is generally difficult to obtain. Third, the computational efficiency is critically important for real-time state estimation, but the computational complexity of the CRLB tends to become greater over the time step. Fourth, given that both are based on noise perturbation strategy, is it possible to correlate the proposed CRLB-based privacy preservation with the differential privacy? These difficulties are solved in this paper and the main contributions are summarized as follows:

- This paper achieves real-time state estimation while protecting exogenous inputs. By employing the CRLB, the MSE of the adversary's estimate for the exogenous inputs is constrained above a specified threshold. By minimizing the MSE of state estimate while ensuring a certain privacy level measured by CRLB, a constrained optimization problem is constructed to ensure privacy level and state estimation accuracy simultaneously. Furthermore, a relaxation approach is proposed to efficiently solve the optimization problem.

- An explicit expression for the CRLB is provided, laying the foundation for CRLB-based privacy preservation. Furthermore, a low-complexity approach for calculating the CRLB is proposed. As a result, the computational complexity is significantly reduced from $\mathcal{O}(k^3)$ to $\mathcal{O}(1)$, which means that the computational complexity of the CRLB is reduced to be time-independent. This development makes valuable sense for real-time state estimation.

- A privacy-preserving state estimation algorithm with low complexity is developed. Moreover, the relationship between our proposed CRLB-based privacy preservation and differential privacy is established. Specifically, the proposed algorithm is proven to ensure (ϵ, δ)-differential privacy. Finally, the effectiveness of the proposed algorithm is demonstrated through two illustrative examples, including a practical scenario for protecting building occupancy.

The reminder of this paper is organized as follows. Section 2 formulates the problem. Section 3 provides the design of the privacy-preserving state estimate and an explicit expression for the CRLB. Section 4 presents the privacy-preserving state estimation algorithm with low complexity and its relation to differential privacy, followed by two examples in Section 5. Section 6 concludes this paper.

Notations. Scalars, vectors and matrices are denoted by lowercase letters, bold lowercase letters, and bold capital letters, respectively. Scalar 0, zero vector, and zero matrix are all denoted by 0 for simplicity. The set of all n-dimensional real vectors and all $n \times m$ real matrices are denoted by $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$, respectively. $\mathbb{Z}^+$ represents the set of positive integers and $\mathbb{S}^+$ represents the set of positive semi-definite matrices. For a square matrix $\mathbf{A}$, $\mathrm{tr}(\mathbf{A})$ denotes its trace. $\mathbf{A} \geqslant 0$ (or $\mathbf{A} > 0$) means that $\mathbf{A}$ is positive semi-definite (or positive definite). The block-diag$(\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_k)$ represents the block diagonal matrix with matrices $\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_k$ on the principal diagonal. For the sequence of square matrices $\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_k$ with the same dimension, we define $\prod_{i=0}^{k} \mathbf{A}_i = \mathbf{A}_k \mathbf{A}_{k-1} \cdots \mathbf{A}_0$, which means that they follow the descending order of the subscript. $\mathbf{I}_n$ represents the $n \times n$ identity matrix. For a vector $\mathbf{a}$, its Euclidean norm is denoted by $\|\mathbf{a}\|$. Further, $\|\mathbf{a}\|_{\mathbf{A}}$ denotes its Euclidean norm weighted with $\mathbf{A} > 0$, i.e., $\sqrt{\mathbf{a}^{\mathrm{T}}\mathbf{A}\mathbf{a}}$. $\mathbb{E}[\cdot]$ is the mathematical expectation operator. $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Besides, $f(x) = \mathcal{O}(g(x))$ if there exists a positive real number m such that $|f(x)| \leqslant mg(x)$.

2 Problem formulation

Consider the following stochastic time-varying dynamic system:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{d}_{k-1} + \mathbf{w}_{k-1}, \\ \mathbf{y}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k, \end{aligned} \tag{1}$$

where $k = 1, 2, \dots$ is the time index; $\mathbf{x}_k \in \mathbb{R}^{n_x}$, $\mathbf{y}_k \in \mathbb{R}^{n_y}$, and $\mathbf{d}_{k-1} \in \mathbb{R}^{n_d}$ are the state, the measurement, and the exogenous input that should be protected at time step k, respectively; $\mathbf{F}_{k-1} \in \mathbb{R}^{n_x\times n_x}$, $\mathbf{G}_{k-1} \in \mathbb{R}^{n_x\times n_d}$, and $\mathbf{H}_k \in \mathbb{R}^{n_y\times n_x}$ are known matrices; $\{\mathbf{w}_k\}$ and $\{\mathbf{v}_k\}$ are mutually independent Gaussian white noise sequences with zero mean and known covariance matrices $\mathbf{Q}_k \geqslant 0$ and $\mathbf{R}_k > 0$, respectively; $\mathbf{x}_0 \sim \mathcal{N}(\bar{\mathbf{x}}_0, \mathbf{P}_0)$ is the initial state independent of the noise sequences. We assume that there is no available prior information about the exogenous input $\mathbf{d}_{k-1}$. Under this case, $\mathbf{d}_{k-1}$ is generally modeled as a deterministic, time-varying, but unknown (or uncertain) quantity (see, e.g., [36, 37]).

Assumption 1. $\mathrm{rank}(\mathbf{H}_k\mathbf{G}_{k-1}) = \mathrm{rank}(\mathbf{G}_{k-1}) = n_d$, for all k.

Remark 1. Assumption 1 is commonly used in existing literature, e.g., [36, 37]. Note that Assumption 1 indicates that $n_x \geqslant n_d$ and $n_y \geqslant n_d$.

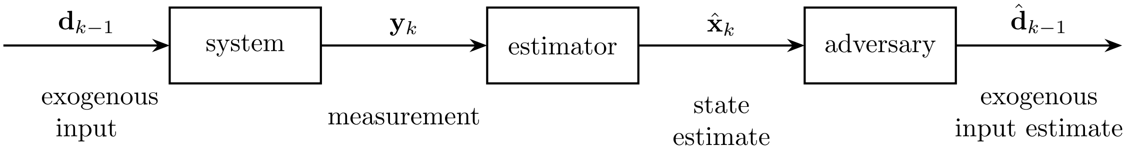


Figure 1 The privacy-preserving state estimation setup.

Let $\hat{\mathbf{x}}_k$ be the estimate of the state $\mathbf{x}_k$ using the measurements $\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_k$. In our setup, $\mathbf{d}_k$ drives the system, and the state estimate $\hat{\mathbf{x}}_k$ should be transmitted to a third party for signal processing or decision-making purposes. In this process, an adversary might have the ability to acquire $\hat{\mathbf{x}}_k$ and use it to infer $\mathbf{d}_{k-1}$, as illustrated in Fig. 1. In this work, the adversary could be an honest-but-curious third party or an external eavesdropper, and is assumed to have the following two abilities:

- It can acquire the system matrices $\mathbf{F}_{k-1}$, $\mathbf{G}_{k-1}$, $\mathbf{Q}_{k-1}$, $\mathbf{H}_k$ and $\mathbf{R}_k$;
- It can store and utilize the m ($m \in \mathbb{Z}^+$ and $m \geqslant 2$) state estimates $\hat{\mathbf{x}}_{k-m+1}, \hat{\mathbf{x}}_{k-m+2}, \dots, \hat{\mathbf{x}}_k$ to infer $\mathbf{d}_{k-1}$.

It should be noted that the second ability makes sense in practice since the adversary cannot store infinite data.

In this paper, we aim to design a state estimate $\hat{\mathbf{x}}_k$ that minimizes the MSE of $\mathbf{x}_k$ on the premise of protecting $\mathbf{d}_{k-1}$. It is worth emphasizing that we focus on protecting the latest exogenous input only at each time step. In another word, at time step k, our goal is to protect $\mathbf{d}_{k-1}$. This makes sense in many practical scenarios. For instance, in the motivating example given by Example 1 later, the adversary is interested in the current building occupancy rather than the past. Before further discussion, we first briefly review the optimal state estimate in the minimum MSE sense for the system (1) without privacy consideration, which is the unbiased minimum-variance state estimate proposed in [36].

2.1 Unbiased minimum-variance state estimate

Let $\hat{\mathbf{x}}_{k-1}^{\mathrm{umv}}$ and $\hat{\mathbf{S}}_{k-1}^{\mathrm{umv}}$, respectively, be the unbiased minimum-variance state estimate and its error covariance matrix at time step $k-1$. Let $\hat{\mathbf{x}}_k^-$ be the estimate of the state $\mathbf{x}_k$ using the measurements $\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{k-1}$, and $\hat{\mathbf{S}}_k^-$ be the error covariance matrix of $\hat{\mathbf{x}}_k^-$. Then, the one-step prediction is given as

$$\hat{\mathbf{x}}_k^- = \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1}^{\mathrm{umv}}, \tag{2}$$

$$\hat{\mathbf{S}}_k^- = \mathbf{F}_{k-1}\hat{\mathbf{S}}_{k-1}^{\mathrm{umv}}\mathbf{F}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1}. \tag{3}$$

Once receiving the measurement $\mathbf{y}_k$, the unbiased minimum-variance state estimate and its error covariance matrix at time step k are given as

$$\hat{\mathbf{x}}_k^{\mathrm{umv}} = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-), \tag{4}$$

$$\begin{aligned}\hat{\mathbf{S}}_k^{\mathrm{umv}} = {} & \hat{\mathbf{S}}_k^- - \hat{\mathbf{S}}_k^-\mathbf{H}_k^{\mathrm{T}}\mathbf{C}_k^{-1}\mathbf{H}_k\hat{\mathbf{S}}_k^- + (\mathbf{G}_{k-1} - \hat{\mathbf{S}}_k^-\mathbf{H}_k^{\mathrm{T}}\mathbf{C}_k^{-1}\mathbf{H}_k\mathbf{G}_{k-1})(\mathbf{G}_{k-1}^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}\mathbf{C}_k^{-1}\mathbf{H}_k\mathbf{G}_{k-1})^{-1} \\ & \cdot (\mathbf{G}_{k-1} - \hat{\mathbf{S}}_k^-\mathbf{H}_k^{\mathrm{T}}\mathbf{C}_k^{-1}\mathbf{H}_k\mathbf{G}_{k-1})^{\mathrm{T}},\end{aligned} \tag{5}$$

where

$$\begin{aligned}\mathbf{K}_k &= \hat{\mathbf{S}}_k^-\mathbf{H}_k^{\mathrm{T}}\mathbf{C}_k^{-1} + (\mathbf{G}_{k-1} - \hat{\mathbf{S}}_k^-\mathbf{H}_k^{\mathrm{T}}\mathbf{C}_k^{-1}\mathbf{H}_k\mathbf{G}_{k-1})(\mathbf{G}_{k-1}^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}\mathbf{C}_k^{-1}\mathbf{H}_k\mathbf{G}_{k-1})^{-1}\mathbf{G}_{k-1}^{\mathrm{T}}\mathbf{H}_k^{\mathrm{T}}\mathbf{C}_k^{-1}, \\ \mathbf{C}_k &= \mathbf{H}_k\hat{\mathbf{S}}_k^-\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k.\end{aligned}$$

Despite achieving good state estimation accuracy, the unbiased minimum-variance state estimate given by (4) cannot be transmitted directly as it may cause a privacy leakage of $\mathbf{d}_{k-1}$, as analyzed below.

2.2 Privacy issue

This section presents an illustrative example to demonstrate why the unbiased minimum-variance state estimate given by (4) may cause a privacy leakage of $\mathbf{d}_{k-1}$. As the inference approach employed by the adversary is neither unique nor the main concern of this paper, we next construct a straightforward but suboptimal estimate $\hat{\mathbf{d}}_{k-1}$ serving as an illustrative example. Specifically, from the state transition equation in (1), we have $\mathbf{G}_{k-1}\mathbf{d}_{k-1} = \mathbf{x}_k - \mathbf{F}_{k-1}\mathbf{x}_{k-1} - \mathbf{w}_{k-1}$. From Assumption 1, by multiplying both sides of the above equation by $\mathbf{G}_{k-1}^{\mathrm{T}}$ and plugging in $\hat{\mathbf{x}}_k$, $\hat{\mathbf{x}}_{k-1}$ and $\hat{\mathbf{w}}_{k-1} = 0$, we can construct the following estimate of $\mathbf{d}_{k-1}$:

$$\hat{\mathbf{d}}_{k-1} = (\mathbf{G}_{k-1}^{\mathrm{T}}\mathbf{G}_{k-1})^{-1}\mathbf{G}_{k-1}^{\mathrm{T}}(\hat{\mathbf{x}}_k - \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1}). \tag{6}$$

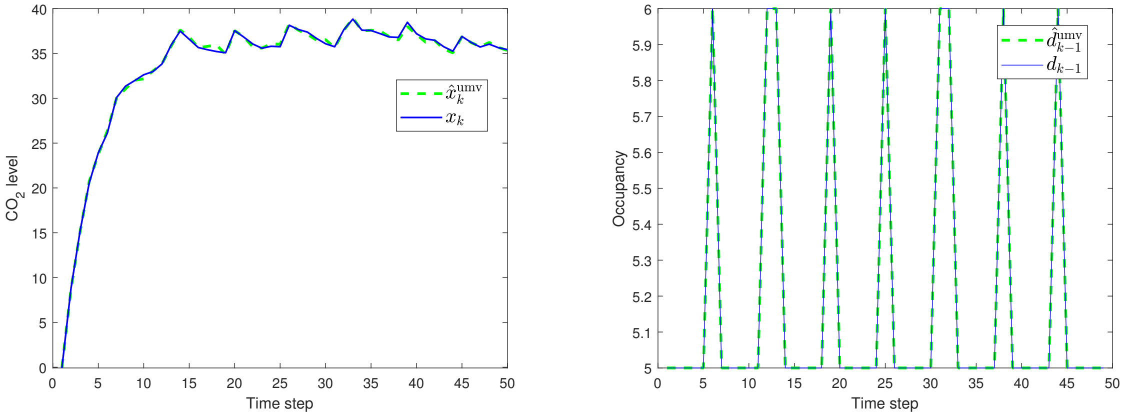
We know from (6) that the adversary could estimate $\mathbf{d}_{k-1}$ by using the state estimates $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{x}}_{k-1}$. It should be noted that $\hat{\mathbf{d}}_{k-1}$ given by (6) is a suboptimal estimate because only two state estimates, $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{x}}_{k-1}$, are used.

To demonstrate the necessity of protecting $\mathbf{d}_{k-1}$, several practical examples from the literature can be considered. For instance, in building automation systems, it is crucial to prevent the inference of occupancy levels from observable measurements such as CO$_2$ concentrations or temperature dynamics [17, 38]. Similarly, in smart grid applications, protecting a user's electricity consumption data from being accurately inferred by adversaries represents another key scenario where privacy is essential [39]. These examples highlight the broad relevance of protecting $\mathbf{d}_{k-1}$ across different domains. The following example is provided to illustrate the privacy leakage of $\mathbf{d}_{k-1}$ caused by the unbiased minimum-variance state estimate given by (4) in a practical scenario.

Example 1 (Building occupancy)**.** Similar to [17, 18], we consider the evolution of CO$_2$ in a building, which can be modeled by the dynamic equation $x_k = ax_{k-1} + bd_{k-1} + w_{k-1}$, where x_k is the level of CO$_2$ at time step k, d_{k-1} is the occupancy of the building, i.e., the number of people in the building, w_{k-1} is the process noise, and $a, b \in \mathbb{R}$ are the parameters. The measurement at time step k is collected by a CO$_2$ sensor with a measurement equation given by $y_k = x_k + v_k$, where v_k is the measurement noise.

The occupancy of the building is sensitive and highly private information that could be reliably estimated from the CO$_2$ levels (see, e.g., [17]). To demonstrate this fact in our setup, we simulate the CO$_2$

evolution by taking $a = 0.75$, $b = 1.75$, and model $\{w_k\}$ and $\{v_k\}$ as Gaussian white noise sequences with variances 0.1 and 0.05, respectively. The initial state is distributed from the Gaussian distribution, with a mean and variance of 0.01. The real privacy state is given as $d_{k-1} = \mathrm{round}(0.5\cos(k) + 5)$, where $\mathrm{round}(\cdot)$ is the rounding function.



(a) CO$_2$ level and its estimate by the unbiased minimum-variance state estimate.

(b) Occupancy and its estimate by the adversary using (6).

Figure 2 CO$_2$ level, occupancy and their estimates.

Fig. 2(a) depicts the trajectory of the CO$_2$ level in the building and the estimated trajectory by the unbiased minimum-variance state estimate given by (4). Fig. 2(b) illustrates the trajectory of the occupancy and the estimated trajectory by the adversary using (6), where the state estimates used in (6) are $\hat{\mathbf{x}}_k^{\mathrm{umv}}$ and $\hat{\mathbf{x}}_{k-1}^{\mathrm{umv}}$. The plots reveal that the CO$_2$ level and the occupancy are well estimated, suggesting that transmitting the CO$_2$ estimates to the third party could result in a privacy loss of the occupancy.

Based on the above analysis, we know that the unbiased minimum-variance state estimate cannot be transmitted directly, and thus, a privacy-preserving state estimate in the minimum MSE sense should be designed.

3 Design of the privacy-preserving state estimate via CRLB

In this section, we provide the privacy-preserving state estimation in the minimum MSE sense via CRLB. Inspired by the noise perturbation strategy, we consider perturbing the unbiased minimum-variance state estimate with a random noise.

3.1 Perturbed noise approach

We design the privacy-preserving state estimate $\hat{\mathbf{x}}_k$ by perturbing $\hat{\mathbf{x}}_k^{\mathrm{umv}}$ with a zero-mean Gaussian noise as follows:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^{\mathrm{umv}} + \boldsymbol{\alpha}_k, \tag{7}$$

where $\boldsymbol{\alpha}_k \sim \mathcal{N}(0, \boldsymbol{\Sigma}_k)$ is independent of $\hat{\mathbf{x}}_k^{\mathrm{umv}}$, and the covariance matrix $\boldsymbol{\Sigma}_k$ is to be determined. The determination of $\boldsymbol{\Sigma}_k$ is the main concern of our noise perturbation strategy since it determines not only the estimation accuracy of $\hat{\mathbf{x}}_k$, but also the privacy level of $\mathbf{d}_{k-1}$. To measure the privacy level, we employ the CRLB, as specified by the following lemma.

Lemma 1 (Cramér-Rao lower bound, [40]). Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ be a sample from $P \in \mathcal{P} = \{P_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$, where Θ is an open set in $\mathbb{R}^m$. Suppose that $T(X)$ is an estimator with $\mathbb{E}[T(X)] = g(\boldsymbol{\theta})$ being a differential function of $\boldsymbol{\theta}$, $P_{\boldsymbol{\theta}}$ has a probability density function $p_{\boldsymbol{\theta}}$ with respect to a measure ν for all $\boldsymbol{\theta} \in \Theta$, and $p_{\boldsymbol{\theta}}$ is differential as a function of $\boldsymbol{\theta}$ and satisfies $\frac{\partial}{\partial \boldsymbol{\theta}} \int h(\mathbf{x}) p_{\boldsymbol{\theta}}(\mathbf{x}) = \int h(\mathbf{x}) \frac{\partial}{\partial \boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\mathbf{x}) \mathrm{d}\nu$, $\boldsymbol{\theta} \in \Theta$, for $h(\mathbf{x}) \equiv 1$ and $h(\mathbf{x}) = T(\mathbf{x})$. Then,

$$\mathrm{Var}(T(X)) \geqslant \left(\frac{\partial}{\partial \boldsymbol{\theta}} g(\boldsymbol{\theta})\right)^{\mathrm{T}} \left(\mathcal{I}(\boldsymbol{\theta})\right)^{-1} \frac{\partial}{\partial \boldsymbol{\theta}} g(\boldsymbol{\theta}), \tag{8}$$

where $\mathcal{I}(\boldsymbol{\theta}) = \mathbb{E}[\frac{\partial}{\partial \boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(X)(\frac{\partial}{\partial \boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(X))^{\mathrm{T}}]$ is the Fisher information matrix and assumed to be positive definite for any $\boldsymbol{\theta} \in \Theta$.

In this work, $X = \{\hat{\mathbf{x}}_{k-m+1}, \hat{\mathbf{x}}_{k-m+2}, \ldots, \hat{\mathbf{x}}_k\}$, $T(X) = \hat{\mathbf{d}}_{k-1}$, $\boldsymbol{\theta} = [\mathbf{d}_0^{\mathrm{T}}, \mathbf{d}_1^{\mathrm{T}}, \ldots, \mathbf{d}_{k-1}^{\mathrm{T}}]^{\mathrm{T}}$, $g(\boldsymbol{\theta}) = \mathbf{d}_{k-1}$. We know from (8) that CRLB provides a lower bound for the MSE matrices of all the unbiased estimators, i.e., provides an intuitive quantified metric for the performance of all the unbiased estimators. In this paper, the merit of adopting CRLB lies in that it constrains the MSE of the adversary's estimate for the $\mathbf{d}_{k-1}$ above a specified threshold. We next provide a way of determining the covariance matrix $\boldsymbol{\Sigma}_k$ through a constrained optimization with privacy constraint measured by CRLB.

3.2 Constrained optimization subject to privacy

We determine the covariance matrix $\boldsymbol{\Sigma}_k$ by minimizing the MSE of (7) as follows:

$$\mathrm{tr}\mathbb{E}\Big[(\hat{\mathbf{x}}_k - \mathbf{x}_k)(\hat{\mathbf{x}}_k - \mathbf{x}_k)^{\mathrm{T}}\Big] = \mathrm{tr}\mathbb{E}\Big[(\hat{\mathbf{x}}_k^{\mathrm{umv}} - \mathbf{x}_k)(\hat{\mathbf{x}}_k^{\mathrm{umv}} - \mathbf{x}_k)^{\mathrm{T}}\Big] + \mathbb{E}\Big[\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^{\mathrm{T}}\Big] = \mathrm{tr}(\hat{\mathbf{S}}_k^{\mathrm{umv}} + \boldsymbol{\Sigma}_k), \tag{9}$$

and by constraining the MSE of the adversary's estimate for the $\mathbf{d}_{k-1}$ above a specified threshold, say γ, simultaneously. Thus, we construct the following constrained optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\Sigma}_k \in \mathbb{S}^+} \quad & \mathrm{tr}(\boldsymbol{\Sigma}_k) \\ \text{s.t.} \quad & \mathrm{tr}\big(\mathrm{CRLB}(\mathbf{d}_{k-1})\big) \geqslant \gamma, \ \boldsymbol{\Sigma}_k \geqslant \sigma \mathbf{I}_{n_x}, \end{aligned} \tag{10}$$

where $\mathrm{CRLB}(\mathbf{d}_{k-1})$ represents the CRLB of $\mathbf{d}_{k-1}$, γ is a given value to quantify privacy level, and σ is a small positive real number for numerical stability. The minimizer of (10) is the sought-after $\boldsymbol{\Sigma}_k$.

We know from (9) that the greater $\boldsymbol{\Sigma}_k$, the lower the state estimation accuracy. In addition, we know from (10) that the greater γ, the higher the privacy level. Thus, by solving the problem (10), we achieve the following two goals simultaneously:

- *Utility*: To minimize the MSE of the state estimate $\hat{\mathbf{x}}_k$;
- *Privacy*: To ensure that the privacy level of $\mathbf{d}_{k-1}$ is no less than a pre-set value γ.

Before solving the problem (10), we should first calculate $\mathrm{CRLB}(\mathbf{d}_{k-1})$. It is worth noting that the state estimates employed by the adversary to estimate $\mathbf{d}_{k-1}$ are $\hat{\mathbf{x}}_{k-m+1}, \hat{\mathbf{x}}_{k-m+2}, \ldots, \hat{\mathbf{x}}_k$ with the latest state estimate $\hat{\mathbf{x}}_k$ depending on $\boldsymbol{\Sigma}_k$. Thus, $\mathrm{CRLB}(\mathbf{d}_{k-1})$ is a function of $\boldsymbol{\Sigma}_k$. We next provide an explicit expression for $\mathrm{CRLB}(\mathbf{d}_{k-1})$ with respect to $\boldsymbol{\Sigma}_k$.

3.3 Explicit expression for CRLB

It follows from (8) that calculating the Fisher information matrix is the premise of calculating the CRLB. Thus, we start by calculating the Fisher information matrix.

1) Calculation for Fisher information matrix. At time step k, we should calculate the Fisher information matrix for the history of exogenous inputs $\{\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{k-1}\}$ based on the state estimates $\hat{\mathbf{x}}_{k-m+1}, \hat{\mathbf{x}}_{k-m+2}, \ldots, \hat{\mathbf{x}}_k$. Denote

$$\begin{aligned} &\mathbf{d}_{0:k-1} := (\mathbf{d}_0^{\mathrm{T}}, \mathbf{d}_1^{\mathrm{T}}, \ldots, \mathbf{d}_{k-1}^{\mathrm{T}})^{\mathrm{T}}, \boldsymbol{\alpha}_{k':k} := (\boldsymbol{\alpha}_{k'}^{\mathrm{T}}, \boldsymbol{\alpha}_{k'+1}^{\mathrm{T}}, \ldots, \boldsymbol{\alpha}_k^{\mathrm{T}})^{\mathrm{T}}, \mathbf{x}_{k':k} := (\mathbf{x}_{k'}^{\mathrm{T}}, \mathbf{x}_{k'+1}^{\mathrm{T}}, \ldots, \hat{\mathbf{x}}_k^{\mathrm{T}})^{\mathrm{T}}, \\ &\hat{\mathbf{x}}_{k':k} := (\hat{\mathbf{x}}_{k'}^{\mathrm{T}}, \hat{\mathbf{x}}_{k'+1}^{\mathrm{T}}, \ldots, \hat{\mathbf{x}}_k^{\mathrm{T}})^{\mathrm{T}}, \hat{\mathbf{x}}_{k':k}^{\mathrm{umv}} := \Big(\big(\hat{\mathbf{x}}_{k'}^{\mathrm{umv}}\big)^{\mathrm{T}}, \big(\hat{\mathbf{x}}_{k'+1}^{\mathrm{umv}}\big)^{\mathrm{T}}, \ldots, \big(\hat{\mathbf{x}}_k^{\mathrm{umv}}\big)^{\mathrm{T}}\Big)^{\mathrm{T}}, k' := k - m + 1. \end{aligned}$$

Then, we have $\hat{\mathbf{x}}_{k':k} = \hat{\mathbf{x}}_{k':k}^{\mathrm{umv}} + \boldsymbol{\alpha}_{k':k}$. Correspondingly, the covariance matrices of $\hat{\mathbf{x}}_{k':k}$ and $\hat{\mathbf{x}}_{k':k}^{\mathrm{umv}}$, denoted by $\hat{\mathbf{P}}_{k':k}$ and $\hat{\mathbf{P}}_{k':k}^{\mathrm{umv}}$, respectively, are correlated by

$$\hat{\mathbf{P}}_{k':k} = \hat{\mathbf{P}}_{k':k}^{\mathrm{umv}} + \boldsymbol{\Lambda}_{\Sigma,k}, \tag{11}$$

where $\boldsymbol{\Lambda}_{\Sigma,k} = \text{block-diag}(\boldsymbol{\Sigma}_{k'}, \boldsymbol{\Sigma}_{k'+1}, \ldots, \boldsymbol{\Sigma}_k)$. Due to the linearity of the system (1) and the estimate (4), we know that $\hat{\mathbf{x}}_{k':k}$ obeys a Gaussian distribution:

$$\hat{\mathbf{x}}_{k':k} \sim \mathcal{N}(\mathbb{E}[\hat{\mathbf{x}}_{k':k}], \hat{\mathbf{P}}_{k':k}). \tag{12}$$

Denote

$$\mathbf{L}_k = \mathbf{L}_{\mathrm{DK},k}\mathbf{L}_{\mathrm{HF},k}\mathbf{\Lambda}_{\mathrm{G},k-1}, \tag{13}$$

where

$$\begin{aligned}
\mathbf{L}_{\mathrm{DK},k} &= \begin{pmatrix} \prod_{i=1}^{k'}\mathbf{D}_i & \prod_{i=2}^{k'}\mathbf{D}_i & \ldots & \mathbf{I}_{n_x} & & & \\ \prod_{i=1}^{k'+1}\mathbf{D}_i & \prod_{i=2}^{k'+1}\mathbf{D}_i & \ldots & \ldots & \mathbf{I}_{n_x} & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \\ \prod_{i=1}^{k}\mathbf{D}_i & \prod_{i=2}^{k}\mathbf{D}_i & \ldots & \ldots & \ldots & \ldots & \mathbf{I}_{n_x} \end{pmatrix}\mathbf{\Lambda}_{\mathrm{K},k}, \\
\mathbf{L}_{\mathrm{HF},k} &= \mathbf{\Lambda}_{\mathrm{H},k}\begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ \mathbf{I}_{n_x} & 0 & \cdots & 0 & 0 \\ \prod_{i=1}^{1}\mathbf{F}_i & \mathbf{I}_{n_x} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \prod_{i=1}^{k-2}\mathbf{F}_i & \prod_{i=2}^{k-2}\mathbf{F}_i & \ldots & \mathbf{I}_{n_x} & 0 \\ \prod_{i=1}^{k-1}\mathbf{F}_i & \prod_{i=2}^{k-1}\mathbf{F}_i & \ldots & \mathbf{F}_{k-1} & \mathbf{I}_{n_x} \end{pmatrix}, \\
\mathbf{D}_k &= (\mathbf{I}_{n_x} - \mathbf{K}_k\mathbf{H}_k)\mathbf{F}_{k-1},\ \mathbf{\Lambda}_{\mathrm{K},k} = \text{block-diag}(\mathbf{K}_0, \mathbf{K}_1, \ldots, \mathbf{K}_k), \\
\mathbf{\Lambda}_{\mathrm{H},k} &= \text{block-diag}(\mathbf{H}_0, \mathbf{H}_1, \ldots, \mathbf{H}_k),\ \mathbf{\Lambda}_{\mathrm{G},k-1} = \text{block-diag}(\mathbf{G}_0, \mathbf{G}_1, \ldots, \mathbf{G}_{k-1}).
\end{aligned}$$

Then, we provide an explicit expression for the Fisher information matrix of $\mathbf{d}_{0:k-1}$, as given in the following theorem.

Theorem 1. For the system (1) and $X = \{\hat{\mathbf{x}}_{k-m+1}, \hat{\mathbf{x}}_{k-m+2}, \ldots, \hat{\mathbf{x}}_k\}$, the Fisher information matrix of $\mathbf{d}_{0:k-1}$ is given as

$$\mathcal{I}(\mathbf{d}_{0:k-1}) = \mathbf{L}_k^{\mathrm{T}}\hat{\mathbf{P}}_{k':k}^{-1}\mathbf{L}_k. \tag{14}$$

To provide a proof of Theorem 1, we need the following three lemmas, where the mean $\mathbb{E}[\hat{\mathbf{x}}_{k':k}]$ and the covariance matrix $\hat{\mathbf{P}}_{k':k}$ in (12) are expressed with respect to $\mathbf{d}_{0:k-1}$. They are the basis of the proof for Theorem 1.

Lemma 2 (Fisher information matrix, [41])**.** Let $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{\theta}), \mathbf{\Sigma}(\boldsymbol{\theta}))$ be an n-variate Gaussian random vector with parameter $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_m)^{\mathrm{T}}$. Then, for $1 \leqslant i, j \leqslant m$, the (i, j) entry of the Fisher information matrix is

$$\mathcal{I}_{i,j} = \frac{\partial\boldsymbol{\mu}}{\partial\theta_i}\mathbf{\Sigma}^{-1}\frac{\partial\boldsymbol{\mu}^{\mathrm{T}}}{\partial\theta_j} + \frac{1}{2}\mathrm{tr}\left(\mathbf{\Sigma}^{-1}\frac{\partial\mathbf{\Sigma}}{\partial\theta_i}\mathbf{\Sigma}^{-1}\frac{\partial\mathbf{\Sigma}}{\partial\theta_j}\right). \tag{15}$$

Based on Lemma 2, computing the Fisher information matrix $\mathcal{I}(\mathbf{d}_{0:k-1})$ requires first determining $\boldsymbol{\mu} = \mathbb{E}[\hat{\mathbf{x}}_{k':k}]$ and $\mathbf{\Sigma} = \hat{\mathbf{P}}_{k':k}$, as presented in the following two lemmas.

Lemma 3. $\mathbb{E}[\hat{\mathbf{x}}_{k':k}] = \mathbf{L}_k\mathbf{d}_{0:k-1} + \mathbf{c}_k$ with $\mathbf{c}_k$ being a constant vector independent of $\mathbf{d}_{0:k-1}$.

Proof. Denote $\mathbf{x}_{0:k} \coloneqq (\mathbf{x}_0^{\mathrm{T}}, \mathbf{x}_1^{\mathrm{T}}, \ldots, \mathbf{x}_k^{\mathrm{T}})^{\mathrm{T}}, \mathbf{y}_{0:k} \coloneqq (\mathbf{y}_0^{\mathrm{T}}, \mathbf{y}_1^{\mathrm{T}}, \ldots, \mathbf{y}_k^{\mathrm{T}})^{\mathrm{T}}, \mathbf{w}_{0:k-1} \coloneqq (\mathbf{w}_0^{\mathrm{T}}, \mathbf{w}_1^{\mathrm{T}}, \ldots, \mathbf{w}_{k-1}^{\mathrm{T}})^{\mathrm{T}}$, and $\mathbf{v}_{0:k} \coloneqq (\mathbf{v}_0^{\mathrm{T}}, \mathbf{v}_1^{\mathrm{T}}, \ldots, \mathbf{v}_k^{\mathrm{T}})^{\mathrm{T}}$. Then, we have

$$\begin{aligned}
\mathbf{x}_{0:k} &= \mathbf{L}_{\mathrm{F},k-1}\big(\mathbf{x}_0^{\mathrm{T}}, \mathbf{w}_{0:k-1}^{\mathrm{T}}\big)^{\mathrm{T}} + \mathbf{L}_{\tilde{\mathrm{F}},k}\mathbf{\Lambda}_{\mathrm{G},k-1}\mathbf{d}_{0:k-1}, \\
\mathbf{y}_{0:k} &= \mathbf{\Lambda}_{\mathrm{H},k}\mathbf{x}_{0:k} + \mathbf{v}_{0:k},
\end{aligned} \tag{16}$$

where

$$\mathbf{L}_{\mathrm{F},k-1} = \begin{pmatrix} \mathbf{I}_{n_x} & & & & \\ \mathbf{F}_0 & \mathbf{I}_{n_x} & & & \\ \mathbf{F}_1\mathbf{F}_0 & \mathbf{F}_1 & \mathbf{I}_{n_x} & & \\ \vdots & \vdots & \vdots & \ddots & \\ \prod_{i=0}^{k-1}\mathbf{F}_i & \prod_{i=1}^{k-1}\mathbf{F}_i & \prod_{i=2}^{k-1}\mathbf{F}_i & \cdots & \mathbf{I}_{n_x} \end{pmatrix},\ \mathbf{L}_{\tilde{\mathrm{F}},k} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ \mathbf{I}_{n_x} & 0 & \cdots & 0 & 0 \\ \prod_{i=1}^{1}\mathbf{F}_i & \mathbf{I}_{n_x} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \prod_{i=1}^{k-2}\mathbf{F}_i & \prod_{i=2}^{k-2}\mathbf{F}_i & \ldots & \mathbf{I}_{n_x} & 0 \\ \prod_{i=1}^{k-1}\mathbf{F}_i & \prod_{i=2}^{k-1}\mathbf{F}_i & \ldots & \mathbf{F}_{k-1} & \mathbf{I}_{n_x} \end{pmatrix}.$$

Further, we have

$$\begin{aligned}
\hat{\mathbf{x}}_k^{\mathrm{umv}} &= \mathbf{K}_k\mathbf{y}_k + \mathbf{D}_k\hat{\mathbf{x}}_{k-1}^{\mathrm{umv}} \\
&= \mathbf{K}_k\mathbf{y}_k + \mathbf{D}_k\mathbf{K}_{k-1}\mathbf{y}_{k-1} + \mathbf{D}_k\mathbf{D}_{k-1}\hat{\mathbf{x}}_{k-2}^{\mathrm{umv}} \\
&= \Big(\textstyle\prod_{i=1}^{k}\mathbf{D}_i\mathbf{K}_0,\ \prod_{i=2}^{k}\mathbf{D}_i\mathbf{K}_1,\ \ldots,\ \mathbf{K}_k\Big)\mathbf{y}_{0:k} + \displaystyle\prod_{i=1}^{k}\mathbf{D}_i(\mathbf{I}_{n_x}-\mathbf{H}_0\mathbf{K}_0)\bar{\mathbf{x}}_0, \\
\hat{\mathbf{x}}_{k':k}^{\mathrm{umv}} &= \mathbf{L}_{\mathrm{DK},k}\mathbf{y}_{0:k} + \Big(\big(\textstyle\prod_{i=1}^{k'}\mathbf{D}_i\big)^{\mathrm{T}},\ \big(\prod_{i=1}^{k'+1}\mathbf{D}_i\big)^{\mathrm{T}},\ldots,\ \big(\prod_{i=1}^{k}\mathbf{D}_i\big)^{\mathrm{T}}\Big)^{\mathrm{T}}(\mathbf{I}_{n_x}-\mathbf{H}_0\mathbf{K}_0)\bar{\mathbf{x}}_0 \\
&= \mathbf{L}_{\mathrm{DK},k}\mathbf{L}_{\mathrm{HF},k}\mathbf{\Lambda}_{\mathrm{G},k-1}\mathbf{d}_{0:k-1} + \mathbf{L}_{\mathrm{DK},k}\Big(\mathbf{\Lambda}_{\mathrm{H},k}\mathbf{L}_{\mathrm{F},k-1}\big(\mathbf{x}_0^{\mathrm{T}},\mathbf{w}_{0:k-1}^{\mathrm{T}}\big)^{\mathrm{T}} + \mathbf{v}_{0:k}\Big) \\
&\quad + \Big(\big(\textstyle\prod_{i=1}^{k'}\mathbf{D}_i\big)^{\mathrm{T}},\ \big(\prod_{i=1}^{k'+1}\mathbf{D}_i\big)^{\mathrm{T}},\ldots,\ \big(\prod_{i=1}^{k}\mathbf{D}_i\big)^{\mathrm{T}}\Big)^{\mathrm{T}}(\mathbf{I}_{n_x}-\mathbf{H}_0\mathbf{K}_0)\bar{\mathbf{x}}_0, \\
&= \mathbf{L}_{\mathrm{DK},k}\mathbf{L}_{\mathrm{HF},k}\mathbf{\Lambda}_{\mathrm{G},k-1}\mathbf{d}_{0:k-1} + \mathbf{L}_{\mathrm{DK},k}\Big(\mathbf{\Lambda}_{\mathrm{H},k}\mathbf{L}_{\mathrm{F},k-1}\big(\mathbf{x}_0^{\mathrm{T}},\mathbf{w}_{0:k-1}^{\mathrm{T}}\big)^{\mathrm{T}} + \mathbf{v}_{0:k}\Big) + \tilde{\mathbf{c}}_k,
\end{aligned} \tag{17}$$

where

$$\tilde{\mathbf{c}}_k = \Big(\big(\textstyle\prod_{i=1}^{k'}\mathbf{D}_i\big)^{\mathrm{T}},\ \big(\prod_{i=1}^{k'+1}\mathbf{D}_i\big)^{\mathrm{T}},\ldots,\ \big(\prod_{i=1}^{k}\mathbf{D}_i\big)^{\mathrm{T}}\Big)^{\mathrm{T}}(\mathbf{I}_{n_x}-\mathbf{H}_0\mathbf{K}_0)\bar{\mathbf{x}}_0.$$

Thus, we can obtain $\mathbb{E}[\hat{\mathbf{x}}_{k':k}] = \mathbb{E}[\hat{\mathbf{x}}_{k':k}^{\mathrm{umv}} + \boldsymbol{\alpha}_{k':k}] = \mathbb{E}[\hat{\mathbf{x}}_{k':k}^{\mathrm{umv}}] = \mathbf{L}_k\mathbf{d}_{0:k-1} + \mathbf{c}_k$, where

$$\mathbf{c}_k = \mathbf{L}_{\mathrm{DK},k}\Big(\mathbf{\Lambda}_{\mathrm{H},k}\mathbf{L}_{\mathrm{F},k-1}\big(\bar{\mathbf{x}}_0^{\mathrm{T}},0\big)^{\mathrm{T}}\Big) + \tilde{\mathbf{c}}_k$$

is a constant independent of $\mathbf{d}_{0:k-1}$. □

Lemma 4. $\hat{\mathbf{P}}_{k':k}$ is independent of $\mathbf{d}_{0:k-1}$.

Proof. From (16) and the first equality of (17), we have

$$\hat{\mathbf{P}}_{k':k}^{\mathrm{umv}} = \mathbf{L}_{\mathrm{DK},k}(\mathbf{\Lambda}_{\mathrm{R},k} + \mathbf{\Lambda}_{\mathrm{H},k}\mathbf{P}_{0:k}\mathbf{\Lambda}_{\mathrm{H},k}^{\mathrm{T}})\mathbf{L}_{\mathrm{DK},k}^{\mathrm{T}}, \tag{18}$$

where $\mathbf{P}_{0:k} = \mathbb{E}[(\mathbf{x}_{0:k} - \mathbb{E}[\mathbf{x}_{0:k}])(\mathbf{x}_{0:k} - \mathbb{E}[\mathbf{x}_{0:k}])]^{\mathrm{T}} = \mathbf{L}_{\mathrm{F},k-1}\text{block-diag}(\mathbf{P}_0, \mathbf{\Lambda}_{\mathrm{Q},k-1})\mathbf{L}_{\mathrm{F},k-1}^{\mathrm{T}}$, $\mathbf{\Lambda}_{\mathrm{Q},k-1} = \text{block-diag}(\mathbf{Q}_0,\mathbf{Q}_1,\ldots,\mathbf{Q}_{k-1})$, and $\mathbf{\Lambda}_{\mathrm{R},k} = \text{block-diag}(\mathbf{R}_0,\mathbf{R}_1,\ldots,\mathbf{R}_k)$. We know from (18) that $\hat{\mathbf{P}}_{k':k}^{\mathrm{umv}}$ is independent of $\mathbf{d}_{k-1}$. Moreover, from (11) and the mutual independence between the perturbed noise and $\mathbf{d}_{k-1}$, we know that $\hat{\mathbf{P}}_{k':k}$ is also independent of $\mathbf{d}_{0:k-1}$. □

Based on the above three lemmas, we now provide a proof of Theorem 1.

Proof of Theorem 1. From (15) and Lemmas 3–4, by substituting $\boldsymbol{\mu} = \mathbb{E}[\hat{\mathbf{x}}_{k':k}]$ and $\mathbf{\Sigma} = \hat{\mathbf{P}}_{k':k}$ into (15), we have

$$\mathcal{I}(\mathbf{d}_{0:k-1}) = \frac{\partial\mathbb{E}[\hat{\mathbf{x}}_{k':k}]}{\partial\mathbf{d}_{0:k-1}}\hat{\mathbf{P}}_{k':k}^{-1}\frac{\partial\mathbb{E}[\hat{\mathbf{x}}_{k':k}^{\mathrm{T}}]}{\partial\mathbf{d}_{0:k-1}} = \frac{\partial(\mathbf{L}_k\mathbf{d}_{0:k-1})}{\partial\mathbf{d}_{0:k-1}}\hat{\mathbf{P}}_{k':k}^{-1}\frac{\partial(\mathbf{L}_k\mathbf{d}_{0:k-1})^{\mathrm{T}}}{\partial\mathbf{d}_{0:k-1}} = \mathbf{L}_k^{\mathrm{T}}\hat{\mathbf{P}}_{k':k}^{-1}\mathbf{L}_k,$$

which completes the proof. □

Based on the explicit Fisher information matrix given by (14), we next provide an explicit expression for CRLB($\mathbf{d}_{k-1}$).

2) Calculation for CRLB. At time step k, we focus on the privacy of $\mathbf{d}_{k-1}$. Thus, following the information inequality given by (8), we have

$$\begin{aligned}
\mathrm{CRLB}(\mathbf{d}_{k-1}) &= \Big(\frac{\partial}{\partial\mathbf{d}_{0:k-1}}g(\mathbf{d}_{0:k-1})\Big)^{\mathrm{T}}\big(\mathcal{I}(\mathbf{d}_{0:k-1})\big)^{-1}\frac{\partial}{\partial\mathbf{d}_{0:k-1}}g(\mathbf{d}_{0:k-1}) \\
&= \Big(0\ \mathbf{I}_{n_d}\Big)\big(\mathbf{L}_k^{\mathrm{T}}\hat{\mathbf{P}}_{k':k}^{-1}\mathbf{L}_k\big)^{-1}\Big(0\ \mathbf{I}_{n_d}\Big)^{\mathrm{T}},
\end{aligned} \tag{19}$$

where $g(\mathbf{d}_{0:k-1}) = \mathbf{d}_{k-1}$ with g being a projection operator. We know from (19) that CRLB($\mathbf{d}_{k-1}$) is the $n_d \times n_d$ block at the right bottom of the inverse of the Fisher information matrix given by (14). However, (19) cannot be employed directly since it is implicit with respect to the optimization variable $\mathbf{\Sigma}_k$. In fact, adopting (19) directly will make (10) difficult to solve. Therefore, we need to simplify (19) further to an explicit form with respect to $\boldsymbol{\Sigma}_k$. By denoting the following matrices as partitioned forms:

$$\hat{\mathbf{P}}^{\mathrm{umv}}_{k':k}=\begin{pmatrix}\hat{\mathbf{P}}^{\mathrm{umv}}_{k':k-1} & \hat{\mathbf{P}}^{\mathrm{umv}}_{k':k-1,k}\\ \hat{\mathbf{P}}^{\mathrm{umv}}_{k,k':k-1} & \hat{\mathbf{P}}^{\mathrm{umv}}_{k}\end{pmatrix},\hat{\mathbf{P}}_{k':k}=\begin{pmatrix}\hat{\mathbf{P}}_{k':k-1} & \hat{\mathbf{P}}^{\mathrm{umv}}_{k':k-1,k}\\ \hat{\mathbf{P}}^{\mathrm{umv}}_{k,k':k-1} & \hat{\mathbf{P}}^{\mathrm{umv}}_{k}+\boldsymbol{\Sigma}_k\end{pmatrix},\mathbf{L}_k=\begin{pmatrix}\mathbf{L}_{11}(k) & 0\\ \mathbf{L}_{21}(k) & \mathbf{G}_{k-1}\end{pmatrix}, \tag{20}$$

where $\mathbf{L}_{11}(k)\in\mathbb{R}^{(m-1)n_x\times(k-1)n_d}$, $\mathbf{L}_{21}(k)\in\mathbb{R}^{n_x\times(k-1)n_d}$, we then provide an explicit expression for $\mathrm{CRLB}(\mathbf{d}_{k-1})$ with respect to $\boldsymbol{\Sigma}_k$ in the following theorem.

Theorem 2. For the system (1) and $X=\{\hat{\mathbf{x}}_{k-m+1},\hat{\mathbf{x}}_{k-m+2},\ldots,\hat{\mathbf{x}}_k\}$, an explicit expression for $\mathrm{CRLB}(\mathbf{d}_{k-1})$ with respect to $\boldsymbol{\Sigma}_k$ is given by

$$\mathrm{CRLB}(\mathbf{d}_{k-1})=\left(\mathbf{G}^{\mathrm{T}}_{k-1}(\boldsymbol{\Sigma}_k+\mathbf{A}_k)^{-1}\mathbf{G}_{k-1}\right)^{-1}, \tag{21}$$

where

$$\begin{aligned}\mathbf{A}_k=&\,\hat{\mathbf{P}}^{\mathrm{umv}}_{k}-\hat{\mathbf{P}}^{\mathrm{umv}}_{k,k':k-1}\hat{\mathbf{P}}^{-1}_{k':k-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k':k-1,k}+\left(\mathbf{L}_{21}(k)-\hat{\mathbf{P}}^{\mathrm{umv}}_{k,k':k-1}\hat{\mathbf{P}}^{-1}_{k':k-1}\mathbf{L}_{11}(k)\right)\\ &\cdot\left(\mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}^{-1}_{k':k-1}\mathbf{L}_{11}(k)\right)^{-1}\left(\mathbf{L}_{21}(k)^{\mathrm{T}}-\mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}^{-1}_{k':k-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k':k-1,k}\right).\end{aligned} \tag{22}$$

Proof. Denote $\hat{\mathbf{P}}^{-1}_{k':k}=\begin{pmatrix}\boldsymbol{\Gamma}_{11} & \boldsymbol{\Gamma}_{12}\\ \boldsymbol{\Gamma}_{21} & \boldsymbol{\Gamma}_{22}\end{pmatrix}$. Then, we have

$$\begin{aligned}&\boldsymbol{\Gamma}_{11}=\hat{\mathbf{P}}^{-1}_{k':k-1}+\hat{\mathbf{P}}^{-1}_{k':k-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k':k-1,k}\boldsymbol{\Delta}^{-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k,k':k-1}\hat{\mathbf{P}}^{-1}_{k':k-1},\boldsymbol{\Delta}=\hat{\mathbf{P}}^{\mathrm{umv}}_{k}+\boldsymbol{\Sigma}_k-\hat{\mathbf{P}}^{\mathrm{umv}}_{k,k':k-1}\hat{\mathbf{P}}^{-1}_{k':k-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k':k-1,k},\\ &\boldsymbol{\Gamma}_{12}=-\hat{\mathbf{P}}^{-1}_{k':k-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k':k-1,k}\boldsymbol{\Delta}^{-1},\ \boldsymbol{\Gamma}_{21}=-\boldsymbol{\Delta}^{-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k,k':k-1}\hat{\mathbf{P}}^{-1}_{k':k-1},\ \boldsymbol{\Gamma}_{22}=\boldsymbol{\Delta}^{-1}.\end{aligned}$$

Further, denote

$$\mathbf{L}^{\mathrm{T}}_k\hat{\mathbf{P}}^{-1}_{k':k}\mathbf{L}_k=\begin{pmatrix}\mathbf{L}_{11}(k)^{\mathrm{T}} & \mathbf{L}_{21}(k)^{\mathrm{T}}\\ 0 & \mathbf{G}^{\mathrm{T}}_{k-1}\end{pmatrix}\begin{pmatrix}\boldsymbol{\Gamma}_{11} & \boldsymbol{\Gamma}_{12}\\ \boldsymbol{\Gamma}_{21} & \boldsymbol{\Gamma}_{22}\end{pmatrix}\begin{pmatrix}\mathbf{L}_{11}(k) & 0\\ \mathbf{L}_{21}(k) & \mathbf{G}_{k-1}\end{pmatrix}=\begin{pmatrix}\boldsymbol{\Phi}_{11} & \boldsymbol{\Phi}_{12}\\ \boldsymbol{\Phi}_{21} & \boldsymbol{\Phi}_{22}\end{pmatrix}.$$

Then, we have

$$\begin{aligned}\boldsymbol{\Phi}_{11}=&\,\mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}^{-1}_{k':k-1}\mathbf{L}_{11}(k)+\mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}^{-1}_{k':k-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k':k-1,k}\boldsymbol{\Delta}^{-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k,k':k-1}\hat{\mathbf{P}}^{-1}_{k':k-1}\mathbf{L}_{11}(k)-\mathbf{L}_{21}(k)^{\mathrm{T}}\boldsymbol{\Delta}^{-1}\\ &\cdot\hat{\mathbf{P}}^{\mathrm{umv}}_{k,k':k-1}\hat{\mathbf{P}}^{-1}_{k':k-1}\mathbf{L}_{11}(k)+\mathbf{L}_{21}(k)^{\mathrm{T}}\boldsymbol{\Delta}^{-1}\mathbf{L}_{21}(k)-\mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}^{-1}_{k':k-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k':k-1,k}\boldsymbol{\Delta}^{-1}\mathbf{L}_{21}(k),\\ \boldsymbol{\Phi}_{12}=&\,\mathbf{L}_{21}(k)^{\mathrm{T}}\boldsymbol{\Delta}^{-1}\mathbf{G}_{k-1}-\mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}^{-1}_{k':k-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k':k-1,k}\boldsymbol{\Delta}^{-1}\mathbf{G}_{k-1},\\ \boldsymbol{\Phi}_{21}=&\,\mathbf{G}^{\mathrm{T}}_{k-1}\boldsymbol{\Delta}^{-1}\mathbf{L}_{21}(k)-\mathbf{G}^{\mathrm{T}}_{k-1}\boldsymbol{\Delta}^{-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k,k':k-1}\hat{\mathbf{P}}^{-1}_{k':k-1}\mathbf{L}_{11}(k),\boldsymbol{\Phi}_{22}=\mathbf{G}^{\mathrm{T}}_{k-1}\boldsymbol{\Delta}^{-1}\mathbf{G}_{k-1}.\end{aligned}$$

Substituting into (19), we have

$$\begin{aligned}\mathrm{CRLB}(\mathbf{d}_{k-1})=&\begin{pmatrix}0 & \mathbf{I}_{n_d}\end{pmatrix}\begin{pmatrix}\boldsymbol{\Phi}_{11} & \boldsymbol{\Phi}_{12}\\ \boldsymbol{\Phi}_{21} & \boldsymbol{\Phi}_{22}\end{pmatrix}^{-1}\begin{pmatrix}0 & \mathbf{I}_{n_d}\end{pmatrix}^{\mathrm{T}}\\ =&\Big(\mathbf{G}^{\mathrm{T}}_{k-1}\boldsymbol{\Delta}^{-1}\mathbf{G}_{k-1}-\left(\mathbf{G}^{\mathrm{T}}_{k-1}\boldsymbol{\Delta}^{-1}\mathbf{L}_{21}(k)-\mathbf{G}^{\mathrm{T}}_{k-1}\boldsymbol{\Delta}^{-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k,k':k-1}\hat{\mathbf{P}}^{-1}_{k':k-1}\mathbf{L}_{11}(k)\right)\\ &\cdot\big(\mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}^{-1}_{k':k-1}\mathbf{L}_{11}(k)+\mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}^{-1}_{k':k-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k':k-1,k}\boldsymbol{\Delta}^{-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k,k':k-1}\hat{\mathbf{P}}^{-1}_{k':k-1}\mathbf{L}_{11}(k)\\ &-\mathbf{L}_{21}(k)^{\mathrm{T}}\boldsymbol{\Delta}^{-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k,k':k-1}\hat{\mathbf{P}}^{-1}_{k':k-1}\mathbf{L}_{11}(k)+\mathbf{L}_{21}(k)^{\mathrm{T}}\boldsymbol{\Delta}^{-1}\mathbf{L}_{21}(k)-\mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}^{-1}_{k':k-1}\\ &\cdot\hat{\mathbf{P}}^{\mathrm{umv}}_{k':k-1,k}\boldsymbol{\Delta}^{-1}\mathbf{L}_{21}(k)\big)^{-1}\left(\mathbf{L}_{21}(k)^{\mathrm{T}}\boldsymbol{\Delta}^{-1}\mathbf{G}_{k-1}-\mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}^{-1}_{k':k-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k':k-1,k}\boldsymbol{\Delta}^{-1}\mathbf{G}_{k-1}\right)\Big)^{-1}.\end{aligned}$$

By some simplifications, we then have

$$\begin{aligned}\mathrm{CRLB}(\mathbf{d}_{k-1})=\Bigg(&\mathbf{G}^{\mathrm{T}}_{k-1}\Big(\boldsymbol{\Delta}^{-1}-\boldsymbol{\Delta}^{-1}\Big(\mathbf{L}_{21}(k)-\hat{\mathbf{P}}^{\mathrm{umv}}_{k,k':k-1}\hat{\mathbf{P}}^{-1}_{k':k-1}\mathbf{L}_{11}(k)\Big)\Big(\mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}^{-1}_{k':k-1}\mathbf{L}_{11}(k)\\ &+\left(\mathbf{L}_{21}(k)^{\mathrm{T}}-\mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}^{-1}_{k':k-1}\hat{\mathbf{P}}^{\mathrm{umv}}_{k,k':k-1}\right)\boldsymbol{\Delta}^{-1}\left(\mathbf{L}_{21}(k)-\hat{\mathbf{P}}^{\mathrm{umv}}_{k,k':k-1}\hat{\mathbf{P}}^{-1}_{k':k-1}\mathbf{L}_{11}(k)\right)\Big)^{-1}\end{aligned}$$

$$\cdot \Big(\mathbf{L}_{21}(k)^{\mathrm{T}} - \mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}_{k':k-1}^{-1}\hat{\mathbf{P}}_{k':k-1,k}^{\mathrm{umv}}\Big)\boldsymbol{\Delta}^{-1}\Big)\mathbf{G}_{k-1}\Big)^{-1}. \tag{23}$$

Using the Woodbury matrix identity (see Page 258 of [42]) for the right-hand side of (23), we have

$$\begin{aligned}
\mathrm{CRLB}(\mathbf{d}_{k-1}) &= \Big(\mathbf{G}_{k-1}^{\mathrm{T}}\Big(\boldsymbol{\Delta} + \big(\mathbf{L}_{21}(k) - \hat{\mathbf{P}}_{k,k':k-1}^{\mathrm{umv}}\hat{\mathbf{P}}_{k':k-1}^{-1}\mathbf{L}_{11}(k)\big)\big(\mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}_{k':k-1}^{-1}\mathbf{L}_{11}(k)\big)^{-1} \\
&\quad \cdot \big(\mathbf{L}_{21}(k)^{\mathrm{T}} - \mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}_{k':k-1}^{-1}\hat{\mathbf{P}}_{k':k-1,k}^{\mathrm{umv}}\big)\Big)^{-1}\mathbf{G}_{k-1}\Big)^{-1} \\
&= \Big(\mathbf{G}_{k-1}^{\mathrm{T}}\Big(\hat{\mathbf{P}}_{k}^{\mathrm{umv}} + \boldsymbol{\Sigma}_k - \hat{\mathbf{P}}_{k,k':k-1}^{\mathrm{umv}}\hat{\mathbf{P}}_{k':k-1}^{-1}\hat{\mathbf{P}}_{k':k-1,k}^{\mathrm{umv}} + \big(\mathbf{L}_{21}(k) - \hat{\mathbf{P}}_{k,k':k-1}^{\mathrm{umv}}\hat{\mathbf{P}}_{k':k-1}^{-1}\mathbf{L}_{11}(k)\big) \\
&\quad \cdot \big(\mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}_{k':k-1}^{-1}\mathbf{L}_{11}(k)\big)^{-1}\big(\mathbf{L}_{21}(k)^{\mathrm{T}} - \mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}_{k':k-1}^{-1}\hat{\mathbf{P}}_{k':k-1,k}^{\mathrm{umv}}\big)\Big)^{-1}\mathbf{G}_{k-1}\Big)^{-1} \\
&= \big(\mathbf{G}_{k-1}^{\mathrm{T}}(\boldsymbol{\Sigma}_k + \mathbf{A}_k)^{-1}\mathbf{G}_{k-1}\big)^{-1},
\end{aligned}$$

where $\mathbf{A}_k$ is given by (22). □

Based on the explicit expression for $\mathrm{CRLB}(\mathbf{d}_{k-1})$ given in Theorem 2, the optimization problem (10) can be rewritten more clearly as follows:

$$\begin{aligned}
\min_{\boldsymbol{\Sigma}_k \in \mathbb{S}^+} \quad & \mathrm{tr}(\boldsymbol{\Sigma}_k) \\
\text{s.t.} \quad & \mathrm{tr}\Big(\big(\mathbf{G}_{k-1}^{\mathrm{T}}(\boldsymbol{\Sigma}_k + \mathbf{A}_k)^{-1}\mathbf{G}_{k-1}\big)^{-1}\Big) \geqslant \gamma,\ \boldsymbol{\Sigma}_k \geqslant \sigma \mathbf{I}_{n_x}.
\end{aligned} \tag{24}$$

Remark 2. As suggested by (24), a larger value of γ tends to result in a larger matrix $\boldsymbol{\Sigma}_k$ in the Löwner order sense. The magnitude of γ reflects the strength of privacy protection. A larger γ corresponds to stronger privacy, which requires increasing the uncertainty of the added noise and consequently reduces the accuracy of state estimation. In another word, enhanced privacy protection is often achieved at the expense of state estimation accuracy. This trade-off is a common characteristic of privacy-preserving methods based on random perturbations. For example, the same principle applies to differential privacy: smaller ϵ and δ values correspond to stronger privacy and require greater noise uncertainty, ultimately at the expense of estimation accuracy (see, e.g., [1]). The specific choice of γ depends on the requirements of the practical problem.

However, there exist two substantial difficulties in solving (24). Firstly, calculating $\mathrm{CRLB}(\mathbf{d}_{k-1})$ directly suffers from a heavy computational burden since the computational complexity of $\mathrm{CRLB}(\mathbf{d}_{k-1})$ becomes greater over the time step k. More specifically, the computational complexity of the matrix $\mathbf{A}_k$ given by (22) increases over the time step k. Essentially, this is caused by the fact that calculating $\mathrm{CRLB}(\mathbf{d}_{k-1})$ requires the history of all the measurements up to time step k, i.e., $\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_k$. As such, the computational complexity of $\mathrm{CRLB}(\mathbf{d}_{k-1})$ tends to become more expensive over time step k. Secondly, (24) is a non-convex optimization problem since its first constraint is non-convex. Thus, an analytic solution of (24) is hard to obtain. As well known, computational efficiency is critically important for real-time state estimation. Thus, reducing the computational complexity of $\mathrm{CRLB}(\mathbf{d}_{k-1})$ and solving (24) efficiently are the two important issues that we will address sequentially in the following section.

4 Privacy-preserving state estimation with low complexity

In this section, we first reduce the computational complexity of $\mathrm{CRLB}(\mathbf{d}_{k-1})$. Then, we propose a relaxation approach for solving (24) and provide the privacy-preserving state estimation algorithm with low complexity. Finally, we show that the proposed algorithm also ensures (ϵ, δ)-differential privacy.

4.1 Privacy-preserving state estimation algorithm with low complexity

For calculating the $\mathrm{CRLB}(\mathbf{d}_{k-1})$ given by (21), the main computation lies in $\mathbf{A}_k$ given by (22). As the computational complexity of (22) mainly lies in the matrices $\hat{\mathbf{P}}_{k':k}^{\mathrm{umv}}$ given by (18) and $\mathbf{L}_k$ given by (13), we next provide low-complexity calculations for these matrices.

1) Recursive calculation for $\hat{\mathbf{P}}^{\rm umv}_{k':k}$***.*** To avoid directly calculating the matrix $\hat{\mathbf{P}}^{\rm umv}_{k':k}$ at each time step k, we design a recursive calculation for $\hat{\mathbf{P}}^{\rm umv}_{k':k}$ as follows.

Theorem 3. The following recursions hold:

$$\begin{aligned}
{\rm Cov}(\mathbf{x}_0, \mathbf{x}_0) &= \mathbf{P}_0, \\
{\rm Cov}(\mathbf{x}_0, \hat{\mathbf{x}}^{\rm umv}_0) &= \mathbf{P}_0\mathbf{H}^{\rm T}_0\mathbf{K}^{\rm T}_0, \qquad (25)\\
{\rm Cov}(\hat{\mathbf{x}}^{\rm umv}_0, \hat{\mathbf{x}}^{\rm umv}_0) &= \mathbf{K}_0\mathbf{H}_0\mathbf{P}_0\mathbf{H}^{\rm T}_0\mathbf{K}^{\rm T}_0 + \mathbf{K}_0\mathbf{R}_0\mathbf{K}^{\rm T}_0. \qquad (26)
\end{aligned}$$

For $k = 1, 2, \ldots,$

$$\begin{aligned}
{\rm Cov}(\mathbf{x}_k, \mathbf{x}_k) &= \mathbf{F}_{k-1}{\rm Cov}(\mathbf{x}_{k-1}, \mathbf{x}_{k-1})\mathbf{F}^{\rm T}_{k-1} + \mathbf{Q}_{k-1}, \qquad (27)\\
{\rm Cov}(\hat{\mathbf{x}}^{\rm umv}_k, \hat{\mathbf{x}}^{\rm umv}_k) &= \mathbf{D}_k{\rm Cov}(\hat{\mathbf{x}}^{\rm umv}_{k-1}, \hat{\mathbf{x}}^{\rm umv}_{k-1})\mathbf{D}^{\rm T}_k + \mathbf{D}_k{\rm Cov}(\hat{\mathbf{x}}^{\rm umv}_{k-1}, \mathbf{x}_{k-1})\mathbf{F}^{\rm T}_{k-1}\mathbf{H}^{\rm T}_k\mathbf{K}^{\rm T}_k \\
&\quad + \mathbf{K}_k\mathbf{H}_k\mathbf{F}_{k-1}{\rm Cov}(\mathbf{x}_{k-1}, \hat{\mathbf{x}}^{\rm umv}_{k-1})\mathbf{D}^{\rm T}_k + \mathbf{K}_k\mathbf{H}_k{\rm Cov}(\mathbf{x}_k, \mathbf{x}_k)\mathbf{H}^{\rm T}_k\mathbf{K}^{\rm T}_k + \mathbf{K}_k\mathbf{R}_k\mathbf{K}^{\rm T}_k, \qquad (28)\\
{\rm Cov}(\mathbf{x}_k, \hat{\mathbf{x}}^{\rm umv}_k) &= \mathbf{F}_{k-1}{\rm Cov}(\mathbf{x}_{k-1}, \hat{\mathbf{x}}^{\rm umv}_{k-1})\mathbf{D}^{\rm T}_k + {\rm Cov}(\mathbf{x}_k, \mathbf{x}_k)\mathbf{H}^{\rm T}_k\mathbf{K}^{\rm T}_k. \qquad (29)
\end{aligned}$$

For $j \in \mathbb{Z}^+$, we have

$$\begin{aligned}
{\rm Cov}(\mathbf{x}_{k+j}, \hat{\mathbf{x}}^{\rm umv}_k) &= \mathbf{F}_{k+j-1}{\rm Cov}(\mathbf{x}_{k+j-1}, \hat{\mathbf{x}}^{\rm umv}_k), \qquad (30)\\
{\rm Cov}(\hat{\mathbf{x}}^{\rm umv}_{k+j}, \hat{\mathbf{x}}^{\rm umv}_k) &= \mathbf{D}_{k+j}{\rm Cov}(\hat{\mathbf{x}}^{\rm umv}_{k+j-1}, \hat{\mathbf{x}}^{\rm umv}_k) + \mathbf{K}_{k+j}\mathbf{H}_{k+j}\mathbf{F}_{k+j-1}{\rm Cov}(\mathbf{x}_{k+j-1}, \hat{\mathbf{x}}^{\rm umv}_k). \qquad (31)
\end{aligned}$$

Proof. To complete this proof, we derive each of (25)–(31) as follows. For (25), we have

$$\mathrm{Cov}(\mathbf{x}_0, \hat{\mathbf{x}}^{\rm umv}_0) = \mathrm{Cov}\big(\mathbf{x}_0, \bar{\mathbf{x}}_0 + \mathbf{K}_0(\mathbf{y}_0 - \mathbf{H}_0\bar{\mathbf{x}}_0)\big) = \mathrm{Cov}(\mathbf{x}_0, \mathbf{K}_0\mathbf{y}_0) = \mathrm{Cov}(\mathbf{x}_0, \mathbf{K}_0\mathbf{H}_0\mathbf{x}_0) = \mathbf{P}_0\mathbf{H}^{\rm T}_0\mathbf{K}^{\rm T}_0.$$

For (26), we have

$$\begin{aligned}
{\rm Cov}(\hat{\mathbf{x}}^{\rm umv}_0, \hat{\mathbf{x}}^{\rm umv}_0) &= {\rm Cov}\big(\bar{\mathbf{x}}_0 + \mathbf{K}_0(\mathbf{y}_0 - \mathbf{H}_0\bar{\mathbf{x}}_0), \bar{\mathbf{x}}_0 + \mathbf{K}_0(\mathbf{y}_0 - \mathbf{H}_0\bar{\mathbf{x}}_0)\big)\\
&= {\rm Cov}\big(\mathbf{K}_0(\mathbf{y}_0 - \mathbf{H}_0\bar{\mathbf{x}}_0), \mathbf{K}_0(\mathbf{y}_0 - \mathbf{H}_0\bar{\mathbf{x}}_0)\big)\\
&= {\rm Cov}(\mathbf{K}_0\mathbf{y}_0, \mathbf{K}_0\mathbf{y}_0)\\
&= {\rm Cov}\big(\mathbf{K}_0(\mathbf{H}_0\mathbf{x}_0 + \mathbf{v}_0), \mathbf{K}_0(\mathbf{H}_0\mathbf{x}_0 + \mathbf{v}_0)\big)\\
&= \mathbf{K}_0\mathbf{H}_0\mathbf{P}_0\mathbf{H}^{\rm T}_0\mathbf{K}^{\rm T}_0 + \mathbf{K}_0\mathbf{R}_0\mathbf{K}^{\rm T}_0.
\end{aligned}$$

For (27), we have

$$\begin{aligned}
{\rm Cov}(\mathbf{x}_k, \mathbf{x}_k) &= {\rm Cov}(\mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{d}_{k-1} + \mathbf{w}_{k-1}, \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{d}_{k-1} + \mathbf{w}_{k-1})\\
&= \mathbf{F}_{k-1}{\rm Cov}(\mathbf{x}_{k-1}, \mathbf{x}_{k-1})\mathbf{F}^{\rm T}_{k-1} + \mathbf{Q}_{k-1}.
\end{aligned}$$

For (28), we have

$$\begin{aligned}
&{\rm Cov}(\hat{\mathbf{x}}^{\rm umv}_k, \hat{\mathbf{x}}^{\rm umv}_k)\\
&= {\rm Cov}(\mathbf{D}_k\hat{\mathbf{x}}^{\rm umv}_{k-1} + \mathbf{K}_k\mathbf{y}_k, \mathbf{D}_k\hat{\mathbf{x}}^{\rm umv}_{k-1} + \mathbf{K}_k\mathbf{y}_k)\\
&= {\rm Cov}(\mathbf{D}_k\hat{\mathbf{x}}^{\rm umv}_{k-1}, \mathbf{D}_k\hat{\mathbf{x}}^{\rm umv}_{k-1}) + {\rm Cov}(\mathbf{D}_k\hat{\mathbf{x}}^{\rm umv}_{k-1}, \mathbf{K}_k\mathbf{y}_k) + {\rm Cov}(\mathbf{K}_k\mathbf{y}_k, \mathbf{D}_k\hat{\mathbf{x}}^{\rm umv}_{k-1}) + {\rm Cov}(\mathbf{K}_k\mathbf{y}_k, \mathbf{K}_k\mathbf{y}_k)\\
&= \mathbf{D}_k{\rm Cov}(\hat{\mathbf{x}}^{\rm umv}_{k-1}, \hat{\mathbf{x}}^{\rm umv}_{k-1})\mathbf{D}^{\rm T}_k + {\rm Cov}\big(\mathbf{D}_k\hat{\mathbf{x}}^{\rm umv}_{k-1}, \mathbf{K}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k)\big) + {\rm Cov}\big(\mathbf{K}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k), \mathbf{D}_k\hat{\mathbf{x}}^{\rm umv}_{k-1}\big)\\
&\quad + {\rm Cov}\big(\mathbf{K}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k), \mathbf{K}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k)\big)\\
&= \mathbf{D}_k{\rm Cov}(\hat{\mathbf{x}}^{\rm umv}_{k-1}, \hat{\mathbf{x}}^{\rm umv}_{k-1})\mathbf{D}^{\rm T}_k + {\rm Cov}(\mathbf{D}_k\hat{\mathbf{x}}^{\rm umv}_{k-1}, \mathbf{K}_k\mathbf{H}_k\mathbf{x}_k) + {\rm Cov}(\mathbf{K}_k\mathbf{H}_k\mathbf{x}_k, \mathbf{D}_k\hat{\mathbf{x}}^{\rm umv}_{k-1})\\
&\quad + {\rm Cov}(\mathbf{K}_k\mathbf{H}_k\mathbf{x}_k, \mathbf{K}_k\mathbf{H}_k\mathbf{x}_k) + {\rm Cov}(\mathbf{K}_k\mathbf{v}_k, \mathbf{K}_k\mathbf{v}_k)\\
&= \mathbf{D}_k{\rm Cov}(\hat{\mathbf{x}}^{\rm umv}_{k-1}, \hat{\mathbf{x}}^{\rm umv}_{k-1})\mathbf{D}^{\rm T}_k + \mathbf{D}_k{\rm Cov}(\hat{\mathbf{x}}^{\rm umv}_{k-1}, \mathbf{x}_{k-1})\mathbf{F}^{\rm T}_{k-1}\mathbf{H}^{\rm T}_k\mathbf{K}^{\rm T}_k + \mathbf{K}_k\mathbf{H}_k\mathbf{F}_{k-1}{\rm Cov}(\mathbf{x}_{k-1}, \hat{\mathbf{x}}^{\rm umv}_{k-1})\mathbf{D}^{\rm T}_k\\
&\quad + \mathbf{K}_k\mathbf{H}_k{\rm Cov}(\mathbf{x}_k, \mathbf{x}_k)\mathbf{H}^{\rm T}_k\mathbf{K}^{\rm T}_k + \mathbf{K}_k\mathbf{R}_k\mathbf{K}^{\rm T}_k.
\end{aligned}$$

For (29), we have

$${\rm Cov}(\mathbf{x}_k, \hat{\mathbf{x}}^{\rm umv}_k) = {\rm Cov}(\mathbf{x}_k, \mathbf{D}_k\hat{\mathbf{x}}^{\rm umv}_{k-1} + \mathbf{K}_k\mathbf{y}_k)$$

$$
\begin{aligned}
&= \mathrm{Cov}(\mathbf{x}_k, \hat{\mathbf{x}}_{k-1}^{\mathrm{umv}})\mathbf{D}_k^{\mathrm{T}} + \mathrm{Cov}(\mathbf{x}_k, \mathbf{y}_k)\mathbf{K}_k^{\mathrm{T}} \\
&= \mathrm{Cov}(\mathbf{x}_k, \hat{\mathbf{x}}_{k-1}^{\mathrm{umv}})\mathbf{D}_k^{\mathrm{T}} + \mathrm{Cov}(\mathbf{x}_k, \mathbf{x}_k)\mathbf{H}_k^{\mathrm{T}}\mathbf{K}_k^{\mathrm{T}} \\
&= \mathbf{F}_{k-1}\mathrm{Cov}(\mathbf{x}_{k-1}, \hat{\mathbf{x}}_{k-1}^{\mathrm{umv}})\mathbf{D}_k^{\mathrm{T}} + \mathrm{Cov}(\mathbf{x}_k, \mathbf{x}_k)\mathbf{H}_k^{\mathrm{T}}\mathbf{K}_k^{\mathrm{T}}.
\end{aligned}
$$

For (30), we have

$$
\begin{aligned}
\mathrm{Cov}(\mathbf{x}_{k+j}, \hat{\mathbf{x}}_k^{\mathrm{umv}}) &= \mathrm{Cov}(\mathbf{F}_{k+j-1}\mathbf{x}_{k+j-1} + \mathbf{G}_{k+j-1}\mathbf{d}_{k+j-1} + \mathbf{w}_{k+j-1}, \hat{\mathbf{x}}_k^{\mathrm{umv}}) \\
&= \mathbf{F}_{k+j-1}\mathrm{Cov}(\mathbf{x}_{k+j-1}, \hat{\mathbf{x}}_k^{\mathrm{umv}}).
\end{aligned}
$$

For (31), we have

$$
\begin{aligned}
\mathrm{Cov}(\hat{\mathbf{x}}_{k+j}^{\mathrm{umv}}, \hat{\mathbf{x}}_k^{\mathrm{umv}}) &= \mathrm{Cov}(\mathbf{D}_{k+j}\hat{\mathbf{x}}_{k+j-1}^{\mathrm{umv}} + \mathbf{K}_{k+j}\mathbf{y}_{k+j}, \hat{\mathbf{x}}_k^{\mathrm{umv}}) \\
&= \mathbf{D}_{k+j}\mathrm{Cov}(\hat{\mathbf{x}}_{k+j-1}^{\mathrm{umv}}, \hat{\mathbf{x}}_k^{\mathrm{umv}}) + \mathbf{K}_{k+j}\mathrm{Cov}(\mathbf{y}_{k+j}, \hat{\mathbf{x}}_k^{\mathrm{umv}}) \\
&= \mathbf{D}_{k+j}\mathrm{Cov}(\hat{\mathbf{x}}_{k+j-1}^{\mathrm{umv}}, \hat{\mathbf{x}}_k^{\mathrm{umv}}) + \mathbf{K}_{k+j}\mathrm{Cov}(\mathbf{H}_{k+j}\mathbf{x}_{k+j} + \mathbf{H}_{k+j}\mathbf{v}_{k+j}, \hat{\mathbf{x}}_k^{\mathrm{umv}}) \\
&= \mathbf{D}_{k+j}\mathrm{Cov}(\hat{\mathbf{x}}_{k+j-1}^{\mathrm{umv}}, \hat{\mathbf{x}}_k^{\mathrm{umv}}) + \mathbf{K}_{k+j}\mathbf{H}_{k+j}\mathrm{Cov}(\mathbf{x}_{k+j}, \hat{\mathbf{x}}_k^{\mathrm{umv}}) \\
&= \mathbf{D}_{k+j}\mathrm{Cov}(\hat{\mathbf{x}}_{k+j-1}^{\mathrm{umv}}, \hat{\mathbf{x}}_k^{\mathrm{umv}}) + \mathbf{K}_{k+j}\mathbf{H}_{k+j}\mathbf{F}_{k+j-1}\mathrm{Cov}(\mathbf{x}_{k+j-1}, \hat{\mathbf{x}}_k^{\mathrm{umv}}).
\end{aligned}
$$

The proof is completed. □

Following Theorem 3, we realize recursive calculation for $\hat{\mathbf{P}}_{k':k}^{\mathrm{umv}}$ by computing each sub-block of $\hat{\mathbf{P}}_{k':k}^{\mathrm{umv}}$ recursively at each time step k.

2) Time-independent calculation for $\mathbf{L}_k$. We can see from (20) that the computational complexity of $\mathbf{L}_k$ is caused by $\mathbf{L}_{11}(k)$ and $\mathbf{L}_{21}(k)$ since the sizes of these two matrices are dependent on the time step k. To solve this problem, we introduce the following pseudo-CRLB (PCRLB) by replacing $\mathbf{L}_k$ with $\tilde{\mathbf{L}}_k$ in (19):

$$
\mathrm{PCRLB}(\mathbf{d}_{k-1}) \coloneqq \begin{pmatrix} 0 & \mathbf{I}_{n_d} \end{pmatrix} \left(\tilde{\mathbf{L}}_k^{\mathrm{T}}\hat{\mathbf{P}}_{k':k}^{-1}\tilde{\mathbf{L}}_k\right)^{-1} \begin{pmatrix} 0 & \mathbf{I}_{n_d} \end{pmatrix}^{\mathrm{T}}, \tag{32}
$$

where $\tilde{\mathbf{L}}_k = \mathbf{L}_k \begin{pmatrix} 0 & \mathbf{I}_{mn_d} \end{pmatrix}^{\mathrm{T}} \in \mathbb{R}^{mn_x \times mn_d}$. It is worth noting that the size of $\tilde{\mathbf{L}}_k$ is independent of the time step k. To calculate $\tilde{\mathbf{L}}_k$ (instead of calculating $\mathbf{L}_k$), denote

$$
\tilde{\mathbf{L}}_{\mathrm{DK},k} = \begin{pmatrix}
\mathbf{I}_{n_x} & & & \\
\mathbf{D}_{k'+1} & \mathbf{I}_{n_x} & & \\
\prod_{i=k'+1}^{k'+2}\mathbf{D}_i & \mathbf{D}_{k'+2} & \mathbf{I}_{n_x} & \\
\vdots & \vdots & \vdots & \ddots \\
\prod_{i=k'+1}^{k}\mathbf{D}_i & \prod_{i=k'+2}^{k}\mathbf{D}_i & \ldots & \ldots \ \mathbf{I}_{n_x}
\end{pmatrix} \text{block-diag}(\mathbf{K}_{k'}, \mathbf{K}_{k'+1}, \ldots, \mathbf{K}_k), \tag{33}
$$

$$
\tilde{\mathbf{L}}_{\mathrm{HF},k} = \text{block-diag}(\mathbf{H}_{k'}, \mathbf{H}_{k'+1}, \ldots, \mathbf{H}_k) \begin{pmatrix}
\mathbf{I}_{n_x} & & & & \\
\mathbf{F}_{k'} & \mathbf{I}_{n_x} & & & \\
\prod_{i=k'}^{k'+1}\mathbf{F}_i & \mathbf{F}_{k'+1} & \mathbf{I}_{n_x} & & \\
\prod_{i=k'}^{k'+2}\mathbf{F}_i & \prod_{i=k'+1}^{k'+2}\mathbf{F}_i & \mathbf{F}_{k'+2} & \mathbf{I}_{n_x} & \\
\vdots & \vdots & \vdots & \vdots & \ddots \\
\prod_{i=k'}^{k-1}\mathbf{F}_i & \prod_{i=k'+1}^{k-1}\mathbf{F}_i & \ldots & \ldots & \ldots \ \mathbf{I}_{n_x}
\end{pmatrix}. \tag{34}
$$

Then, the following proposition presents the calculation for $\tilde{\mathbf{L}}_k$.

Proposition 1. $\tilde{\mathbf{L}}_k$ can be calculated as follows:

$$
\tilde{\mathbf{L}}_k = \tilde{\mathbf{L}}_{\mathrm{DK},k}\tilde{\mathbf{L}}_{\mathrm{HF},k}\text{block-diag}(\mathbf{G}_{k-m}, \ldots, \mathbf{G}_{k-1}). \tag{35}
$$

Proof. From (13) and $\tilde{\mathbf{L}}_k = \mathbf{L}_k \begin{pmatrix} 0 & \mathbf{I}_{mn_d} \end{pmatrix}^{\mathrm{T}}$, we have

$$
\tilde{\mathbf{L}}_k = \mathbf{L}_{\mathrm{DK},k}\mathbf{L}_{\mathrm{HF},k}\mathbf{\Lambda}_{\mathrm{G},k-1} \begin{pmatrix} 0 & \mathbf{I}_{mn_d} \end{pmatrix}^{\mathrm{T}}
$$

$$
\begin{aligned}
&= \mathbf{L}_{\mathrm{DK},k}\mathbf{L}_{\mathrm{HF},k}\begin{pmatrix} 0 \\ \text{block-diag}(\mathbf{G}_{k-m},\ldots,\mathbf{G}_{k-1})\end{pmatrix}\\
&= \mathbf{L}_{\mathrm{DK},k}\begin{pmatrix} 0 \\ \tilde{\mathbf{L}}_{\mathrm{HF},k}\text{block-diag}(\mathbf{G}_{k-m},\ldots,\mathbf{G}_{k-1})\end{pmatrix}\\
&= \tilde{\mathbf{L}}_{\mathrm{DK},k}\tilde{\mathbf{L}}_{\mathrm{HF},k}\text{block-diag}(\mathbf{G}_{k-m},\ldots,\mathbf{G}_{k-1}).
\end{aligned}
$$

This completes the proof. □

The following proposition provides an explicit expression for the PCRLB($\mathbf{d}_{k-1}$).

Proposition 2. An explicit expression for the PCRLB($\mathbf{d}_{k-1}$) with respect to $\mathbf{\Sigma}_k$ is given by

$$
\mathrm{PCRLB}(\mathbf{d}_{k-1}) = \big(\mathbf{G}_{k-1}^{\mathrm{T}}(\mathbf{\Sigma}_k + \tilde{\mathbf{A}}_k)^{-1}\mathbf{G}_{k-1}\big)^{-1}, \tag{36}
$$

where

$$
\begin{aligned}
\tilde{\mathbf{A}}_k = {}& \hat{\mathbf{P}}_k^{\mathrm{umv}} - \hat{\mathbf{P}}_{k,k':k-1}^{\mathrm{umv}}\hat{\mathbf{P}}_{k':k-1}^{-1}\hat{\mathbf{P}}_{k':k-1,k}^{\mathrm{umv}} + \big(\tilde{\mathbf{L}}_{21} - \hat{\mathbf{P}}_{k,k':k-1}^{\mathrm{umv}}\hat{\mathbf{P}}_{k':k-1}^{-1}\tilde{\mathbf{L}}_{11}\big)\\
&\cdot\big(\tilde{\mathbf{L}}_{11}^{\mathrm{T}}\hat{\mathbf{P}}_{k':k-1}^{-1}\tilde{\mathbf{L}}_{11}\big)^{-1}\big(\tilde{\mathbf{L}}_{21}^{\mathrm{T}} - \tilde{\mathbf{L}}_{11}^{\mathrm{T}}\hat{\mathbf{P}}_{k':k-1}^{-1}\hat{\mathbf{P}}_{k':k-1,k}^{\mathrm{umv}}\big),
\end{aligned} \tag{37}
$$

$$
\tilde{\mathbf{L}}_{11} = \begin{pmatrix}\mathbf{I}_{(m-1)n_x} & 0\end{pmatrix}\tilde{\mathbf{L}}_k\begin{pmatrix}\mathbf{I}_{(m-1)n_d} & 0\end{pmatrix}^{\mathrm{T}}, \tilde{\mathbf{L}}_{21} = \begin{pmatrix}0 & \mathbf{I}_{n_x}\end{pmatrix}\tilde{\mathbf{L}}_k\begin{pmatrix}\mathbf{I}_{(m-1)n_d} & 0\end{pmatrix}^{\mathrm{T}}. \tag{38}
$$

Proof. The proof is similar to that of Theorem 2, and hence, omitted here. □

Note that the sizes of $\tilde{\mathbf{L}}_{11}$ and $\tilde{\mathbf{L}}_{21}$ given by (38) are independent of the time step k. We next provide the relation between the PCRLB and the CRLB.

Proposition 3. For the PCRLB given by (32) and the CRLB given by (19), the following inequality holds:

$$
\mathrm{tr}\big(\mathrm{CRLB}(\mathbf{d}_{k-1})\big) \geqslant \mathrm{tr}\big(\mathrm{PCRLB}(\mathbf{d}_{k-1})\big). \tag{39}
$$

Proof. Denote $\mathbf{L}_k^{\mathrm{T}}\hat{\mathbf{P}}_{k':k}^{-1}\mathbf{L}_k = \begin{pmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}^{\mathrm{T}} & \mathbf{C}\end{pmatrix}$, where $\mathbf{A} \in \mathbb{R}^{(k-m)n_d\times(k-m)n_d}$, $\mathbf{B} \in \mathbb{R}^{(k-m)n_d\times mn_d}$, $\mathbf{C} \in \mathbb{R}^{mn_d\times mn_d}$. Then, we have

$$
\begin{aligned}
\mathrm{CRLB}(\mathbf{d}_{k-1}) &= \begin{pmatrix}0 & \mathbf{I}_{n_d}\end{pmatrix}\big(\mathbf{L}_k^{\mathrm{T}}\hat{\mathbf{P}}_{k':k}^{-1}\mathbf{L}_k\big)^{-1}\begin{pmatrix}0 & \mathbf{I}_{n_d}\end{pmatrix}^{\mathrm{T}}\\
&= \begin{pmatrix}0 & \mathbf{I}_{n_d}\end{pmatrix}\begin{pmatrix}0 & \mathbf{I}_{mn_d}\end{pmatrix}\big(\mathbf{L}_k^{\mathrm{T}}\hat{\mathbf{P}}_{k':k}^{-1}\mathbf{L}_k\big)^{-1}\begin{pmatrix}0 & \mathbf{I}_{mn_d}\end{pmatrix}^{\mathrm{T}}\begin{pmatrix}0 & \mathbf{I}_{n_d}\end{pmatrix}^{\mathrm{T}}\\
&= \begin{pmatrix}0 & \mathbf{I}_{n_d}\end{pmatrix}(\mathbf{C}^{-1} + \mathbf{C}^{-1}\mathbf{B}^{\mathrm{T}}(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^{\mathrm{T}})^{-1}\mathbf{B}\mathbf{C}^{-1})\begin{pmatrix}0 & \mathbf{I}_{n_d}\end{pmatrix}^{\mathrm{T}},
\end{aligned} \tag{40}
$$

$$
\mathrm{PCRLB}(\mathbf{d}_{k-1}) = \begin{pmatrix}0 & \mathbf{I}_{n_d}\end{pmatrix}\left(\begin{pmatrix}0 & \mathbf{I}_{mn_d}\end{pmatrix}\mathbf{L}_k^{\mathrm{T}}\hat{\mathbf{P}}_{k':k}^{-1}\mathbf{L}_k\begin{pmatrix}0 & \mathbf{I}_{mn_d}\end{pmatrix}^{\mathrm{T}}\right)^{-1}\begin{pmatrix}0 & \mathbf{I}_{n_d}\end{pmatrix}^{\mathrm{T}} = \begin{pmatrix}0 & \mathbf{I}_{n_d}\end{pmatrix}\mathbf{C}^{-1}\begin{pmatrix}0 & \mathbf{I}_{n_d}\end{pmatrix}^{\mathrm{T}}. \tag{41}
$$

Due to $\mathbf{C}^{-1}\mathbf{B}^{\mathrm{T}}(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^{\mathrm{T}})^{-1}\mathbf{B}\mathbf{C}^{-1} \geqslant 0$, we obtain CRLB($\mathbf{d}_{k-1}$) $\geqslant$ PCRLB($\mathbf{d}_{k-1}$), and thus, $\mathrm{tr}\big(\mathrm{CRLB}(\mathbf{d}_{k-1})\big) \geqslant \mathrm{tr}(\mathrm{PCRLB}(\mathbf{d}_{k-1}))$. □

Remark 3. From (40) and (41), we can directly derive the approximation error between PCRLB($\mathbf{d}_{k-1}$) and CRLB($\mathbf{d}_{k-1}$), denoted as

$$
e = \begin{pmatrix}0 & \mathbf{I}_{n_d}\end{pmatrix}(\mathbf{C}^{-1}\mathbf{B}^{\mathrm{T}}(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^{\mathrm{T}})^{-1}\mathbf{B}\mathbf{C}^{-1})\begin{pmatrix}0 & \mathbf{I}_{n_d}\end{pmatrix}^{\mathrm{T}}.
$$

Based on Propositions 2 and 3, the original optimization problem (24) can be relaxed as follows:

$$
\begin{aligned}
\min_{\mathbf{\Sigma}_k\in\mathbb{S}^+} \quad & \mathrm{tr}(\mathbf{\Sigma}_k)\\
\text{s.t.} \quad & \mathrm{tr}\Big(\big(\mathbf{G}_{k-1}^{\mathrm{T}}(\mathbf{\Sigma}_k + \tilde{\mathbf{A}}_k)^{-1}\mathbf{G}_{k-1}\big)^{-1}\Big) \geqslant \gamma,\ \mathbf{\Sigma}_k \geqslant \sigma\mathbf{I}_{n_x},
\end{aligned} \tag{42}
$$

It is not difficult to find that (42) has the same form as (24). This indicates that we reduce the computational complexity without increasing the difficulty of solving the problem (24).

3) Computational complexity. Compared with the CRLB($\mathbf{d}_{k-1}$) given by (21), the computational complexity of the PCRLB($\mathbf{d}_{k-1}$) given by (36) is greatly reduced, as specified by the following theorem.

Theorem 4 (Computational complexity)**.** From (21) to (36), the computational complexity reduces from $\mathcal{O}(k^3)$ to $\mathcal{O}(1)$.

Proof. For calculating (21), the main computation lies in (22) and is specified as follows. Specifically, for calculating $\hat{\mathbf{P}}_k^{\text{umv}} - \hat{\mathbf{P}}_{k,k':k-1}^{\text{umv}}\hat{\mathbf{P}}_{k':k-1}^{-1}\hat{\mathbf{P}}_{k':k-1,k}^{\text{umv}}$, the computational complexity is $(n_x(m-1))^2 + n_x^4(m-1)^2 + n_x^2$, for calculating $\mathbf{L}_{21}(k) - \hat{\mathbf{P}}_{k,k':k-1}^{\text{umv}}\hat{\mathbf{P}}_{k':k-1}^{-1}\mathbf{L}_{11}(k)$, the computational complexity is $(n_x(m-1))^3 + n_x^3 n_d(m-1)^2(k-1) + n_x n_d(k-1)$, and for calculating $(\mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}_{k':k-1}^{-1}\mathbf{L}_{11}(k))^{-1}$, the computational complexity is $(n_x(m-1))^3 + n_x^2 n_d^2(m-1)^2(k-1)^2 + ((k-1)n_d)^3$. Additionally, for calculating $(\mathbf{L}_{21}(k) - \hat{\mathbf{P}}_{k,k':k-1}^{\text{umv}}\hat{\mathbf{P}}_{k':k-1}^{-1}\mathbf{L}_{11}(k))(\mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}_{k':k-1}^{-1}\mathbf{L}_{11}(k))^{-1}(\mathbf{L}_{21}(k)^{\mathrm{T}} - \mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}_{k':k-1}^{-1}\hat{\mathbf{P}}_{k':k-1,k}^{\text{umv}})$, the computational complexity is $n_x^2 n_d^2(k-1)^2$, and for calculating $\hat{\mathbf{P}}_k^{\text{umv}} - \hat{\mathbf{P}}_{k,k':k-1}^{\text{umv}}\hat{\mathbf{P}}_{k':k-1}^{-1}\hat{\mathbf{P}}_{k':k-1,k}^{\text{umv}} + (\mathbf{L}_{21}(k) - \hat{\mathbf{P}}_{k,k':k-1}^{\text{umv}}\hat{\mathbf{P}}_{k':k-1}^{-1}\mathbf{L}_{11}(k))(\mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}_{k':k-1}^{-1}\mathbf{L}_{11}(k))^{-1}(\mathbf{L}_{21}(k)^{\mathrm{T}} - \mathbf{L}_{11}(k)^{\mathrm{T}}\hat{\mathbf{P}}_{k':k-1}^{-1}\hat{\mathbf{P}}_{k':k-1,k}^{\text{umv}})$, the computational complexity is n_x^2. Hence, for calculating (22), the total computational complexity is

$$\begin{aligned} & n_d^3(k-1)^3 + n_x^2 n_d^2(1+(m-1)^2)(k-1)^2 + (n_x^3 n_d(m-1)^2 + n_x n_d)(k-1) \\ & + n_x^4(m-1)^2 + 3n_x^3(m-1)^3 + 2n_x^2. \end{aligned} \tag{43}$$

For calculating (36), the main computation lies in (37). By replacing k with m in (43), we can obtain the computational complexity as follows:

$$\begin{aligned} & n_x^2 n_d^2(m-1)^4 + n_x^3 n_d(m-1)^3 + n_d^3(m-1)^3 + 3n_x^3(m-1)^3 + n_x^2 n_d^2(m-1)^2 + n_x^4(m-1)^2 \\ & + n_x n_d(m-1) + 2n_x^2. \end{aligned} \tag{44}$$

From (43) to (44), the computational complexity reduces from $\mathcal{O}(k^3)$ to $\mathcal{O}(1)$. □

Theorem 4 indicates that from $\mathrm{CRLB}(\mathbf{d}_{k-1})$ to $\mathrm{PCRLB}(\mathbf{d}_{k-1})$, the computational complexity is reduced from $\mathcal{O}(k^3)$ to $\mathcal{O}(1)$. In another word, the computational complexity of CRLB is reduced to be time-independent.

4) Relaxed solution. Since the analytic solution of the problem (42) is hard to obtain, we next provide a relaxed solution of (42). Denote the singular value decomposition of $\mathbf{G}_{k-1}$ by

$$\mathbf{G}_{k-1} = \mathbf{U}_k \begin{pmatrix} \boldsymbol{\Upsilon}_k & 0 \end{pmatrix}^{\mathrm{T}} \mathbf{V}_k, \tag{45}$$

where $\mathbf{U}_k$ and $\mathbf{V}_k$ are orthogonal matrices, $\boldsymbol{\Upsilon}_k \in \mathbb{R}^{n_d \times n_d}$ is a diagonal matrix, and denote

$$\mathbf{M}_k = \mathbf{U}_k^{\mathrm{T}}(\tilde{\mathbf{A}}_k + \sigma\mathbf{I}_{n_x})\mathbf{U}_k = \begin{pmatrix} \tilde{\mathbf{A}}_{11} & \tilde{\mathbf{A}}_{12} \\ \tilde{\mathbf{A}}_{21} & \tilde{\mathbf{A}}_{22} \end{pmatrix}, \tag{46}$$

where $\tilde{\mathbf{A}}_{11} \in \mathbb{R}^{n_d \times n_d}$. Then, the following theorem provides a relaxed solution of (42).

Theorem 5. A relaxed solution of (42) is given by

$$\boldsymbol{\Sigma}_k = \mathbf{U}_k \begin{pmatrix} \tilde{\boldsymbol{\Sigma}}_{11}^* - \tilde{\mathbf{A}}_{11} + \sigma\mathbf{I}_{n_d} & 0 \\ 0 & \sigma\mathbf{I}_{n_x - n_d} \end{pmatrix} \mathbf{U}_k^{\mathrm{T}}, \tag{47}$$

where $\tilde{\boldsymbol{\Sigma}}_{11}^*$ is the minimizer of the following semi-definite programming problem:

$$\begin{aligned} \min_{\tilde{\boldsymbol{\Sigma}}_{11} \in \mathbb{S}^+} \quad & \mathrm{tr}(\tilde{\boldsymbol{\Sigma}}_{11}) \\ \text{s.t.} \quad & \mathrm{tr}\big(\boldsymbol{\Upsilon}_k^{-2}(\tilde{\boldsymbol{\Sigma}}_{11} - \tilde{\mathbf{A}}_{12}\tilde{\mathbf{A}}_{22}^{-1}\tilde{\mathbf{A}}_{21})\big) \geqslant \gamma, \ \tilde{\boldsymbol{\Sigma}}_{11} \geqslant \tilde{\mathbf{A}}_{11}. \end{aligned} \tag{48}$$

Proof. Problem (42) is equivalent to

$$\begin{aligned} \min_{\boldsymbol{\Sigma}_k \in \mathbb{S}^+} \quad & \mathrm{tr}(\boldsymbol{\Sigma}_k + \tilde{\mathbf{A}}_k) \\ \text{s.t.} \quad & \mathrm{tr}\Big(\big(\mathbf{G}_{k-1}^{\mathrm{T}}(\boldsymbol{\Sigma}_k + \tilde{\mathbf{A}}_k)^{-1}\mathbf{G}_{k-1}\big)^{-1}\Big) \geqslant \gamma, \ \boldsymbol{\Sigma}_k + \tilde{\mathbf{A}}_k \geqslant \tilde{\mathbf{A}}_k + \sigma\mathbf{I}_{n_x}. \end{aligned} \tag{49}$$

From (45), we have $\mathbf{G}_{k-1}^{\mathrm{T}}(\boldsymbol{\Sigma}_k+\tilde{\mathbf{A}}_k)^{-1}\mathbf{G}_{k-1} = \mathbf{V}_k^{\mathrm{T}}\begin{pmatrix}\boldsymbol{\Upsilon}_k & 0\end{pmatrix}\mathbf{U}_k^{\mathrm{T}}(\boldsymbol{\Sigma}_k+\tilde{\mathbf{A}}_k)^{-1}\mathbf{U}_k\begin{pmatrix}\boldsymbol{\Upsilon}_k & 0\end{pmatrix}^{\mathrm{T}}\mathbf{V}_k$. Denote $\tilde{\boldsymbol{\Sigma}}_k^{-1}=\mathbf{U}_k^{\mathrm{T}}(\boldsymbol{\Sigma}_k+\tilde{\mathbf{A}}_k)^{-1}\mathbf{U}_k=\begin{pmatrix}\tilde{\boldsymbol{\Sigma}}_{11} & \tilde{\boldsymbol{\Sigma}}_{12}\\ \tilde{\boldsymbol{\Sigma}}_{21} & \tilde{\boldsymbol{\Sigma}}_{22}\end{pmatrix}^{-1}$ with $\tilde{\boldsymbol{\Sigma}}_{11}\in\mathbb{R}^{n_d\times n_d}$. Then, the problem (49) is equivalent to

$$\begin{aligned}\min_{\tilde{\boldsymbol{\Sigma}}_k\in\mathbb{S}^+}\quad & \mathrm{tr}(\tilde{\boldsymbol{\Sigma}}_k)\\ \text{s.t.}\quad & \mathrm{tr}\left(\left(\mathbf{V}_k^{\mathrm{T}}\begin{pmatrix}\boldsymbol{\Upsilon}_k & 0\end{pmatrix}\tilde{\boldsymbol{\Sigma}}_k^{-1}\begin{pmatrix}\boldsymbol{\Upsilon}_k & 0\end{pmatrix}^{\mathrm{T}}\mathbf{V}_k\right)^{-1}\right)\geqslant\gamma,\ \tilde{\boldsymbol{\Sigma}}_k\geqslant\mathbf{U}_k^{\mathrm{T}}(\tilde{\mathbf{A}}_k+\sigma\mathbf{I}_{n_x})\mathbf{U}_k.\end{aligned}\tag{50}$$

Here, we use the fact that $\mathrm{tr}(\boldsymbol{\Sigma}_k+\tilde{\mathbf{A}}_k)=\mathrm{tr}\big(\mathbf{U}_k^{\mathrm{T}}(\boldsymbol{\Sigma}_k+\tilde{\mathbf{A}}_k)\mathbf{U}_k\big)=\mathrm{tr}(\tilde{\boldsymbol{\Sigma}}_k)$. Denote $\tilde{\boldsymbol{\Sigma}}_k^{-1}=\begin{pmatrix}\mathbf{S}_{11} & \mathbf{S}_{12}\\ \mathbf{S}_{21} & \mathbf{S}_{22}\end{pmatrix}$. Then, we have $\mathbf{S}_{11}=(\tilde{\boldsymbol{\Sigma}}_{11}-\tilde{\boldsymbol{\Sigma}}_{12}\tilde{\boldsymbol{\Sigma}}_{22}^{-1}\tilde{\boldsymbol{\Sigma}}_{21})^{-1}\in\mathbb{R}^{n_d\times n_d}$, and

$$\mathrm{tr}\left(\left(\mathbf{V}_k^{\mathrm{T}}\begin{pmatrix}\boldsymbol{\Upsilon}_k & 0\end{pmatrix}\tilde{\boldsymbol{\Sigma}}_k^{-1}\begin{pmatrix}\boldsymbol{\Upsilon}_k & 0\end{pmatrix}^{\mathrm{T}}\mathbf{V}_k\right)^{-1}\right)=\mathrm{tr}\big((\boldsymbol{\Upsilon}_k\mathbf{S}_{11}\boldsymbol{\Upsilon}_k)^{-1}\big)=\mathrm{tr}\big(\boldsymbol{\Upsilon}_k^{-2}(\tilde{\boldsymbol{\Sigma}}_{11}-\tilde{\boldsymbol{\Sigma}}_{12}\tilde{\boldsymbol{\Sigma}}_{22}^{-1}\tilde{\boldsymbol{\Sigma}}_{21})\big).$$

Thus, the problem (50) is equivalent to

$$\begin{aligned}\min_{\tilde{\boldsymbol{\Sigma}}_{11},\tilde{\boldsymbol{\Sigma}}_{22}\in\mathbb{S}^+}\quad & \mathrm{tr}(\tilde{\boldsymbol{\Sigma}}_{11})+\mathrm{tr}(\tilde{\boldsymbol{\Sigma}}_{22})\\ \text{s.t.}\quad & \mathrm{tr}\big(\boldsymbol{\Upsilon}_k^{-2}(\tilde{\boldsymbol{\Sigma}}_{11}-\tilde{\boldsymbol{\Sigma}}_{12}\tilde{\boldsymbol{\Sigma}}_{22}^{-1}\tilde{\boldsymbol{\Sigma}}_{21})\big)\geqslant\gamma,\ \begin{pmatrix}\tilde{\boldsymbol{\Sigma}}_{11} & \tilde{\boldsymbol{\Sigma}}_{12}\\ \tilde{\boldsymbol{\Sigma}}_{21} & \tilde{\boldsymbol{\Sigma}}_{22}\end{pmatrix}\geqslant\begin{pmatrix}\tilde{\mathbf{A}}_{11} & \tilde{\mathbf{A}}_{12}\\ \tilde{\mathbf{A}}_{21} & \tilde{\mathbf{A}}_{22}\end{pmatrix}.\end{aligned}\tag{51}$$

By letting $\tilde{\boldsymbol{\Sigma}}_{22}=\tilde{\mathbf{A}}_{22}$, the problem (51) can be relaxed to

$$\begin{aligned}\min_{\tilde{\boldsymbol{\Sigma}}_{11}\in\mathbb{S}^+}\quad & \mathrm{tr}(\tilde{\boldsymbol{\Sigma}}_{11})\\ \text{s.t.}\quad & \mathrm{tr}\big(\boldsymbol{\Upsilon}_k^{-2}(\tilde{\boldsymbol{\Sigma}}_{11}-\tilde{\boldsymbol{\Sigma}}_{12}\tilde{\mathbf{A}}_{22}^{-1}\tilde{\boldsymbol{\Sigma}}_{21})\big)\geqslant\gamma,\ \begin{pmatrix}\tilde{\boldsymbol{\Sigma}}_{11} & \tilde{\boldsymbol{\Sigma}}_{12}\\ \tilde{\boldsymbol{\Sigma}}_{21} & \tilde{\mathbf{A}}_{22}\end{pmatrix}\geqslant\begin{pmatrix}\tilde{\mathbf{A}}_{11} & \tilde{\mathbf{A}}_{12}\\ \tilde{\mathbf{A}}_{21} & \tilde{\mathbf{A}}_{22}\end{pmatrix}.\end{aligned}\tag{52}$$

We adopt this relaxation because $\begin{pmatrix}\tilde{\boldsymbol{\Sigma}}_{11} & \tilde{\boldsymbol{\Sigma}}_{12}\\ \tilde{\boldsymbol{\Sigma}}_{21} & \tilde{\boldsymbol{\Sigma}}_{22}\end{pmatrix}\geqslant\begin{pmatrix}\tilde{\mathbf{A}}_{11} & \tilde{\mathbf{A}}_{12}\\ \tilde{\mathbf{A}}_{21} & \tilde{\mathbf{A}}_{22}\end{pmatrix}\geqslant 0$ implies $\tilde{\boldsymbol{\Sigma}}_{11}\geqslant\tilde{\mathbf{A}}_{11}$ and $\tilde{\boldsymbol{\Sigma}}_{22}\geqslant\tilde{\mathbf{A}}_{22}$. From the sufficient and necessary condition for the positive semi-definiteness of a matrix in terms of a generalized Schur complement [43], we have

$$\begin{pmatrix}\tilde{\boldsymbol{\Sigma}}_{11}-\tilde{\mathbf{A}}_{11} & \tilde{\boldsymbol{\Sigma}}_{12}-\tilde{\mathbf{A}}_{12}\\ \tilde{\boldsymbol{\Sigma}}_{21}-\tilde{\mathbf{A}}_{21} & 0\end{pmatrix}\geqslant 0 \iff \tilde{\boldsymbol{\Sigma}}_{11}-\tilde{\mathbf{A}}_{11}\geqslant 0,\ \tilde{\boldsymbol{\Sigma}}_{21}-\tilde{\mathbf{A}}_{21}=\tilde{\boldsymbol{\Sigma}}_{12}-\tilde{\mathbf{A}}_{12}=0.$$

Thus, the problem (52) is equivalent to the problem (48). Further, let $\boldsymbol{\Sigma}_{11}^{*}$ be the solution of (48). Then,

$$\begin{aligned}\boldsymbol{\Sigma}_k &= \mathbf{U}_k\tilde{\boldsymbol{\Sigma}}_k\mathbf{U}_k^{\mathrm{T}}-\tilde{\mathbf{A}}_k\\ &= \mathbf{U}_k\begin{pmatrix}\tilde{\boldsymbol{\Sigma}}_{11}^{*} & \tilde{\mathbf{A}}_{12}\\ \tilde{\mathbf{A}}_{21} & \tilde{\mathbf{A}}_{22}\end{pmatrix}\mathbf{U}_k^{\mathrm{T}}-\tilde{\mathbf{A}}_k\\ &= \mathbf{U}_k\begin{pmatrix}\tilde{\mathbf{A}}_{11} & \tilde{\mathbf{A}}_{12}\\ \tilde{\mathbf{A}}_{21} & \tilde{\mathbf{A}}_{22}\end{pmatrix}\mathbf{U}_k^{\mathrm{T}}-\tilde{\mathbf{A}}_k+\mathbf{U}_k\begin{pmatrix}\tilde{\boldsymbol{\Sigma}}_{11}^{*}-\tilde{\mathbf{A}}_{11} & 0\\ 0 & 0\end{pmatrix}\mathbf{U}_k^{\mathrm{T}}\\ &= \sigma\mathbf{I}_{n_x}+\mathbf{U}_k\begin{pmatrix}\tilde{\boldsymbol{\Sigma}}_{11}^{*}-\tilde{\mathbf{A}}_{11} & 0\\ 0 & 0\end{pmatrix}\mathbf{U}_k^{\mathrm{T}}\\ &= \mathbf{U}_k\begin{pmatrix}\tilde{\boldsymbol{\Sigma}}_{11}^{*}-\tilde{\mathbf{A}}_{11}+\sigma\mathbf{I}_{n_d} & 0\\ 0 & \sigma\mathbf{I}_{n_x-n_d}\end{pmatrix}\mathbf{U}_k^{\mathrm{T}}.\end{aligned}$$

This completes the proof. □

Theorem 5 indicates that the problem of determining the covariance matrix $\mathbf{\Sigma}_k$ of the perturbed noise is finally converted to the problem of solving the semi-definite programming problem given by (48). Instead of solving the original optimization problem (24) directly, we relax it to the semi-definite programming problem (48), which has the following benefits: i) the computational complexity is greatly reduced; ii) the semi-definite programming problem can be solved efficiently; iii) the privacy level remains no less than γ.

5) Algorithm. The proposed privacy-preserving state estimation algorithm with low complexity is summarized in Algorithm 1. The semi-definite programming problem (48) therein can be efficiently solved by, e.g., the CVX package. For more details about the CVX package, the readers are referred to [44].

Algorithm 1 Privacy-preserving state estimation algorithm with low complexity

```
Input: $\hat{\mathbf{x}}^{\text{umv}}_{k-1}$, $\hat{\mathbf{S}}^{\text{umv}}_{k-1}$, $m$, $\sigma$, $\gamma$
Prediction:
Calculate $\hat{\mathbf{x}}^{-}_{k}$ and $\hat{\mathbf{S}}^{-}_{k}$ using (2) and (3).
Update:
Calculate $\hat{\mathbf{x}}^{\text{umv}}_{k}$ and $\hat{\mathbf{S}}^{\text{umv}}_{k}$ based on $\hat{\mathbf{x}}^{\text{umv}}_{k-1}$ and $\hat{\mathbf{S}}^{\text{umv}}_{k-1}$ using (4) and (5).
Calculate the covariance matrix of the perturbed noise:
Set $k' = k - m + 1$.
Calculate each sub-block of $\hat{\mathbf{P}}^{\text{umv}}_{k':k}$ recursively using (27)–(31), and then obtain $\hat{\mathbf{P}}^{\text{umv}}_{k,k':k-1}$ and $\hat{\mathbf{P}}^{\text{umv}}_{k':k-1,k}$
  using (20).
Calculate $\hat{\mathbf{P}}^{-1}_{k':k-1}$ using (11).
Calculate $\tilde{\mathbf{L}}_k$ using (35), and then obtain $\tilde{\mathbf{L}}_{11}$ and $\tilde{\mathbf{L}}_{21}$ using (38).
Calculate $\tilde{\mathbf{A}}_k$ using (37).
Perform singular value decomposition on $\mathbf{G}_{k-1}$ to obtain $\boldsymbol{\Upsilon}_k$ and $\mathbf{U}_k$ using (45).
Calculate $\mathbf{M}_k$ using (46) with the pre-set $\sigma$ to obtain $\tilde{\mathbf{A}}_{11}$, $\tilde{\mathbf{A}}_{12}$, $\tilde{\mathbf{A}}_{21}$, and $\tilde{\mathbf{A}}_{22}$.
Solve the semi-definite programming problem (48) to obtain $\tilde{\mathbf{\Sigma}}^{*}_{11}$, with $\gamma$ being the pre-set privacy
  level.
Calculate $\mathbf{\Sigma}_k$ using (47).
Privacy-preserving state estimation:
Generate $\boldsymbol{\alpha}_k \sim \mathcal{N}(0, \mathbf{\Sigma}_k)$.
Set $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}^{\text{umv}}_{k} + \boldsymbol{\alpha}_k$.
Set $\hat{\mathbf{S}}_k = \hat{\mathbf{S}}^{\text{umv}}_{k} + \mathbf{\Sigma}_k$.
Output: $\hat{\mathbf{x}}_k$, $\hat{\mathbf{S}}_k$
```

4.2 Relation to differential privacy

This subsection shows that Algorithm 1 also ensures (ϵ, δ)-differential privacy. To this end, we first define the sensitivity of Algorithm 1 as follows, which determines how much perturbed noise should be added.

Definition 1 (Sensitivity). Suppose that $\mathbb{R}^{n_d}$ is equipped with an adjacency relation Adj. The sensitivity of a query $q : \mathbb{R}^{n_d} \to \mathbb{R}^{n}$ is defined as

$$\Delta_{\mathbf{A}} q := \sup_{\mathbf{d}_k, \mathbf{d}'_k : \text{Adj}(\mathbf{d}_k, \mathbf{d}'_k)} \|q(\mathbf{d}_k) - q(\mathbf{d}'_k)\|_{\mathbf{A}}, \ \mathbf{A} > 0.$$

At time step k, the change in $\mathbf{d}_{k-1}$ only affects $\hat{\mathbf{x}}_k$ rather than $\hat{\mathbf{x}}_{k':k-1}$, and thus, the Gaussian mechanism $\mathcal{M}_q : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ is defined by $\mathcal{M}_q(\mathbf{d}_{k-1}) = \hat{\mathbf{x}}_k = q(\mathbf{d}_{k-1}) + \omega$, where $\omega \sim \mathcal{N}(0, \hat{\mathbf{P}}_k)$, $\hat{\mathbf{P}}_k = \hat{\mathbf{P}}^{\text{umv}}_{k} + \mathbf{\Sigma}_k$ is the covariance matrix of $\hat{\mathbf{x}}_k$, and $q(\mathbf{d}_k) = \mathbb{E}[\hat{\mathbf{x}}_k] = \mathbf{K}_k \mathbf{H}_k \mathbf{G}_{k-1} \mathbf{d}_{k-1} + \mathbf{c}_k$ with $\mathbf{c}_k$ being a constant vector. Based on these analyses, we can show exactly what level of differential privacy Algorithm 1 can ensure in the following theorem.

Theorem 6 (Differential privacy). For any $\epsilon \geqslant 0$, Algorithm 1 is (ϵ, δ)-differentially private with $\delta = \mathcal{Q}(\xi) = \frac{1}{\sqrt{2\pi}} \int_{\xi}^{\infty} \exp\{-\frac{z^2}{2}\} \mathrm{d}z$ and

$$\xi = -\frac{\Delta_{\hat{\mathbf{P}}_k^{-1}} q}{2} + \frac{\epsilon}{\Delta_{\hat{\mathbf{P}}_k^{-1}} q}. \tag{53}$$

Proof. Let $\mathbf{d}_{k-1}$, $\mathbf{d}_{k'-1}$ be two adjacent elements in $\mathbb{R}^{n_d}$, and denote $\mathbf{v} := \mathbf{K}_k\mathbf{H}_k\mathbf{G}_{k-1}(\mathbf{d}_{k-1} - \mathbf{d}_{k'-1})$. Then, for any Borel set $S \in \mathbb{R}^{n_x}$, we have

$$\begin{aligned}
&P(\mathcal{M}_q(\mathbf{d}_{k-1}) \in S)\\
&= \int_S \mathcal{N}(\mathbf{u}; q(\mathbf{d}_{k-1}), \hat{\mathbf{P}}_k)\mathrm{d}\mathbf{u}\\
&= \int_S (2\pi)^{-\frac{n_x}{2}} \det(\hat{\mathbf{P}}_k)^{-\frac{1}{2}} \exp\bigg\{ -\frac{1}{2}\big\|\mathbf{u} - q(\mathbf{d}_{k-1})\big\|^2_{\hat{\mathbf{P}}_k^{-1}}\bigg\}\mathrm{d}\mathbf{u}\\
&= \int_S (2\pi)^{-\frac{n_x}{2}} \det(\hat{\mathbf{P}}_k)^{-\frac{1}{2}} \exp\bigg\{ -\frac{1}{2}\big\|\mathbf{u} - q(\mathbf{d}_{k'-1})\big\|^2_{\hat{\mathbf{P}}_k^{-1}}\bigg\} \exp\bigg\{\big(\mathbf{u} - q(\mathbf{d}_{k'-1})\big)^{\mathrm{T}}\hat{\mathbf{P}}_k^{-1}\mathbf{v} - \frac{1}{2}\|\mathbf{v}\|^2_{\hat{\mathbf{P}}_k^{-1}}\bigg\}\mathrm{d}\mathbf{u}.
\end{aligned}$$

Let $f(\mathbf{u}) = (\mathbf{u} - q(\mathbf{d}_{k'-1}))^{\mathrm{T}}\hat{\mathbf{P}}_k^{-1}\mathbf{v} - \frac{1}{2}\|\mathbf{v}\|^2_{\hat{\mathbf{P}}_k^{-1}}$, $A = \{\mathbf{u}|f(\mathbf{u}) \leqslant \epsilon\}$. Then,

$$\begin{aligned}
&P(\mathcal{M}_q(\mathbf{d}_{k-1}) \in S)\\
&= \int_{S\cap A} (2\pi)^{-\frac{n_x}{2}} \det(\hat{\mathbf{P}}_k)^{-\frac{1}{2}} \exp\bigg\{ -\frac{1}{2}\big\|\mathbf{u} - q(\mathbf{d}_{k'-1})\big\|^2_{\hat{\mathbf{P}}_k^{-1}}\bigg\} \exp\{f(\mathbf{u})\}\mathrm{d}\mathbf{u} + \int_{S\cap A^c} \mathcal{N}(\mathbf{u}; q(\mathbf{d}_{k-1}), \hat{\mathbf{P}}_k)\mathrm{d}\mathbf{u}\\
&\leqslant e^{\epsilon}P(\mathcal{M}_q(\mathbf{d}_{k'-1}) \in S) + \int_S \mathcal{N}(\mathbf{u}; q(\mathbf{d}_{k-1}), \hat{\mathbf{P}}_k)\mathcal{I}_{[f(\mathbf{u})>\epsilon]}\mathrm{d}\mathbf{u},
\end{aligned}$$

where A^c is the complement set to A, and $\mathcal{I}_{[f(\mathbf{u})>\epsilon]}$ is an indicative function defined as $\mathcal{I}_{[f(\mathbf{u})>\epsilon]} = \begin{cases} 1 & f(\mathbf{u}) > \epsilon \\ 0 & f(\mathbf{u}) \leqslant \epsilon \end{cases}$. Let $\mathbf{y} = \hat{\mathbf{P}}_k^{-\frac{1}{2}}(\mathbf{u} - q(\mathbf{d}_{k-1}))$. Then, we have

$$\begin{aligned}
P(\mathcal{M}_q(\mathbf{d}_{k-1}) \in S) &\leqslant e^{\epsilon}P(\mathcal{M}_q(\mathbf{d}_{k'-1}) \in S) + \int_S \mathcal{N}(\mathbf{y}; 0, \mathbf{I}_{n_x})\mathcal{I}_{[\mathbf{v}^{\mathrm{T}}\hat{\mathbf{P}}_k^{-\frac{1}{2}}\mathbf{y} > -\frac{1}{2}\|\mathbf{v}\|^2_{\hat{\mathbf{P}}_k^{-1}} + \epsilon]}\mathrm{d}\mathbf{y}\\
&\leqslant e^{\epsilon}P(\mathcal{M}_q(\mathbf{d}_{k'-1}) \in S) + \mathcal{Q}\bigg(-\frac{1}{2}\|\mathbf{v}\|_{\hat{\mathbf{P}}_k^{-1}} + \frac{\epsilon}{\|\mathbf{v}\|_{\hat{\mathbf{P}}_k^{-1}}}\bigg)\\
&\leqslant e^{\epsilon}P(\mathcal{M}_q(\mathbf{d}_{k'-1}) \in S) + \mathcal{Q}\bigg(-\frac{\Delta_{\hat{\mathbf{P}}_k^{-1}}q}{2} + \frac{\epsilon}{\Delta_{\hat{\mathbf{P}}_k^{-1}}q}\bigg).
\end{aligned}$$

Thus, for any $\epsilon \geqslant 0$, $\mathcal{M}_q$ is $(\epsilon, \mathcal{Q}(\xi))$-differentially private, where ξ is given by (53). □

Remark 4. As established by Theorem 6, for any fixed $\epsilon \geqslant 0$, a larger value γ leads to a smaller value of δ. This relationship can be further detailed: increasing γ results in a larger matrix $\hat{\mathbf{P}}_k$. Furthermore, it follows from (53) that a larger $\hat{\mathbf{P}}_k$ implies a greater ξ, which consequently leads to a smaller δ. This relationship is further illustrated in Fig. 6.

5 Examples

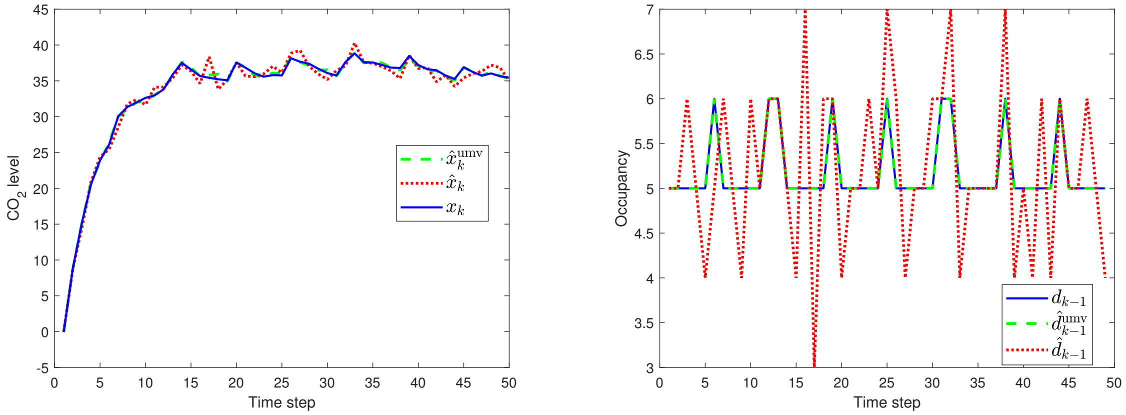
In this section, we demonstrate the effectiveness of the proposed algorithm through two examples.

5.1 Practical case: Building occupancy

Consider Example 1 and set $\sigma = 10^{-4}$, $m = 2$ and $\gamma = 0.5$ in Algorithm 1. Figs. 3(a) and 3(b) highlight that the estimated trajectory of CO_2 level by the proposed algorithm is very close to the real one. In contrast, the estimated trajectory of the occupancy is not identical with the real one almost every time step. This demonstrates that the proposed algorithm performs well in estimating CO_2 level and protecting occupancy, ensuring privacy and utility simultaneously.

5.2 Numerical case: A two-dimensional model

Consider the two-dimensional dynamic system $\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}d_k + \mathbf{w}_k$, where $\mathbf{x}_k \in \mathbb{R}^2$, $d_k \in \mathbb{R}$, $\mathbf{F} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, $\mathbf{G} = \begin{pmatrix} 0.5 & 0.5 \end{pmatrix}^{\mathrm{T}}$, $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$ with $\mathbf{Q}_k = 2\mathbf{I}_2$. The initial state $\mathbf{x}_0$ was drawn from

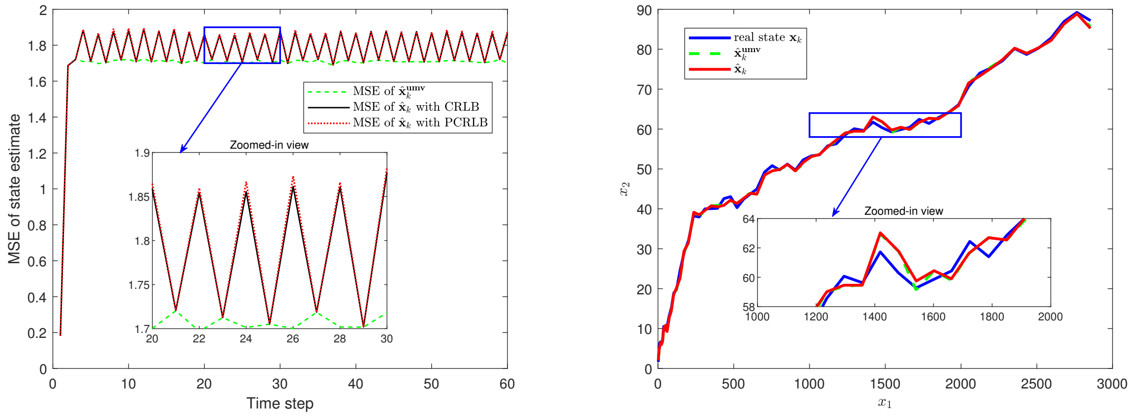


(a) CO$_2$ level and its estimates by unbiased minimum-variance and privacy-preserving state estimates.

(b) Occupancy and its estimates by the adversary using unbiased minimum-variance and privacy-preserving state estimates.

Figure 3 CO$_2$ level, occupancy and their estimates.

the Gaussian distribution $\mathcal{N}(\bar{\mathbf{x}}_0, \mathbf{P}_0)$ with $\bar{\mathbf{x}}_0 = \begin{pmatrix} 2 & 2 \end{pmatrix}^{\mathrm{T}}$ and $\mathbf{P}_0 = 0.1\mathbf{I}_2$. The exogenous input d_k is generated independently and identically distributed from a uniform distribution over the interval $[0, 5]$. The measurement equation is $\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k$, where $\mathbf{y}_k \in \mathbb{R}^2$, $\mathbf{H} = \mathbf{I}_2$, $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)$ with $\mathbf{R}_k = \mathbf{I}_2$. We set $\gamma = 11$, $m = 3$ and $\sigma = 10^{-4}$ in Algorithm 1. In addition to the PCRLB derived in (36), we also employ the CRLB presented in (21) for comparative analysis.



(a) MSEs of unbiased minimum-variance and privacy-preserving state estimates.

(b) Trajectories of real states, unbiased minimum-variance state estimates, and privacy-preserving state estimates.

Figure 4 Comparison of unbiased minimum-variance and privacy-preserving state estimates.

Fig. 4(a) reports the MSEs of the proposed algorithm over 50 time steps and 500 Monte Carlo runs (i.e., the black solid line and the red dotted line), where the unbiased minimum-variance state estimate serves as a benchmark (i.e., the green dashed line). In Fig. 4(a), both the black solid line and the red dotted line lie above the green dashed line, which revels that the MSEs of the proposed algorithm are greater than those of the unbiased minimum-variance state estimate. This is reasonable due to the perturbed noise. Besides, by noting that the red dotted line and the black solid line exhibit highly similar estimation performance, we can infer that the approximation errors of PCRLB and CRLB are negligible in this example, which is specifically reflected in their nearly identical accuracy in state estimation. Fig. 4(b) depicts the real state trajectory and the estimated trajectories by the unbiased minimum-variance state estimator and the proposed algorithm. This figure reveals that the proposed algorithm has good estimation performance similar to the that of the unbiased minimum-variance estimator.

Fig. 5 highlights that the MSEs of the adversary's estimates for d_k with either unbiased minimum-variance (green dashed line) or privacy-preserving state estimates (black solid and red dotted lines). We have the following two observations: i) MSEs of $\hat{d}_k$ with privacy-preserving state estimates are greater than those of $\hat{d}_k^{\mathrm{umv}}$ with unbiased minimum-variance state estimates. This is resulted from the perturbed noise strategy. ii) MSEs of $\hat{d}_k$ with privacy-preserving state estimates are greater than the threshold γ, while MSEs of $\hat{d}_k^{\mathrm{umv}}$ with unbiased minimum-variance state estimates are smaller than the threshold γ. This phenomenon indicates that the proposed privacy-preserving state estimates do protect the exogenous input such that the MSEs of the adversary's estimates for d_k are not less than γ.

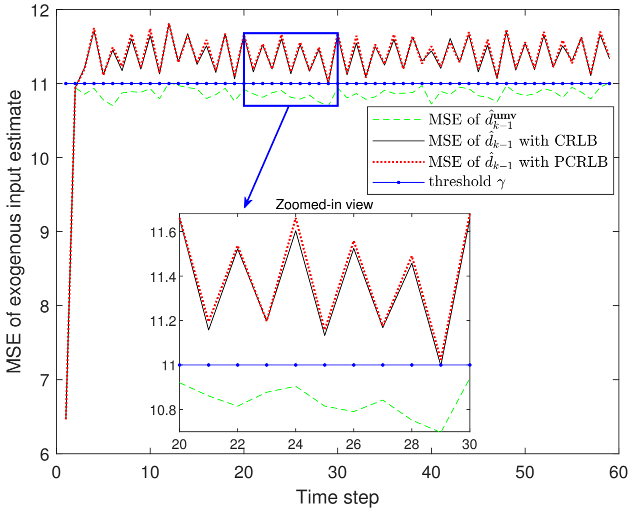


Figure 5 MSEs of the adversary's estimates for d_k with either unbiased minimum-variance or privacy-preserving state estimates.

Fig. 6 quantitatively displays the correlation between the proposed CRLB-based method and differential privacy. We can see that as the parameter γ measuring the privacy level increases, the curves approach the origin more closely, indicating that the level of differential privacy is higher.

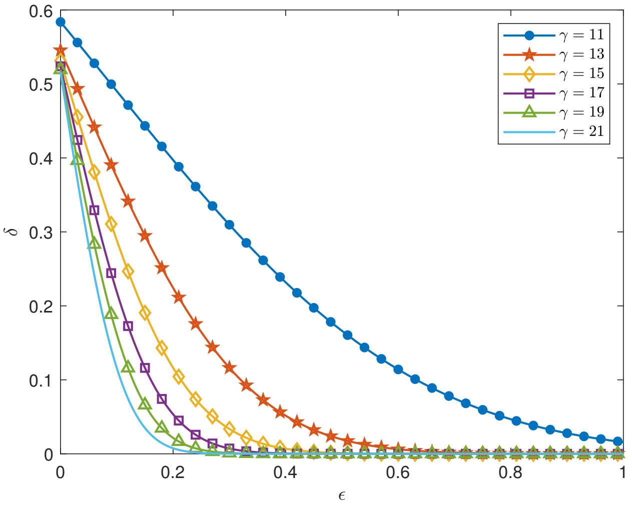


Figure 6 Correlation between the proposed CRLB-based method and differential privacy.

Table 1 reports the MSEs of $\hat{\mathbf{x}}$ and $\hat{d}$ averaged over 50 time steps and 500 Monte Carlo runs under different privacy level γ. It is observed that as γ increases, both the MSE of $\hat{\mathbf{x}}$ and the MSE of the adversary's estimate for d_k increase. This is because, as the privacy level increases, the accuracy of state estimation decreases while the level of exogenous input protection strengthens, demonstrating the trade-off between the state estimation accuracy and the privacy level. Additionally, it is noteworthy that

Table 1 Averaged MSEs of $\hat{\mathbf{x}}$ and $\hat{d}$ under different privacy level γ.

γ	9	10	11	12	13
MSE of $\hat{\mathbf{x}}$	1.6852	1.6852	1.7664	1.9998	2.2488
MSE of $\hat{d}$	10.7997	10.7997	11.3009	12.8472	14.4323

when $\gamma = 9$ and $\gamma = 10$, the MSE values are identical. This phenomenon stems from the fact that, under the framework of unbiased minimum variance state estimation, the MSE associated with the adversary's optimal unbiased estimate of d exceeds 10, thereby eliminating the necessity for additional noise injection.

6 Conclusion

We have developed a CRLB-based privacy-preserving state estimation algorithm with low complexity, which also ensures (ϵ, δ)-differential privacy. Specifically, by perturbing the unbiased minimum-variance state estimate with a zero-mean Gaussian noise, we have designed a noisy state estimate that prevents the adversary from inferring the exogenous inputs. Adopting the CRLB allows constraining the MSE of the adversary's estimate for the exogenous inputs. By minimizing the MSE of the noisy state estimate subject to a certain privacy level measured by CRLB, we have ensured privacy and utility by solving a non-convex constrained optimization. Additionally, we have provided explicit and low-complexity calculations for CRLB, significantly reducing the computational complexity from $\mathcal{O}(k^3)$ to $\mathcal{O}(1)$. Furthermore, we have solved the constrained optimization efficiently by providing a relaxed solution. Finally, we have demonstrated the effectiveness of the proposed algorithm through two examples, including a practical scenario for protecting building occupancy.

In practice, the measurements are usually collected by some sensors in a network structure, so our future work includes studying the privacy-preserving state estimation problem for multi-sensor systems.

Acknowledgements The work was supported by National Natural Science Foundation of China (Grant Nos. 62025306, 62433020, 62203045, T2293770), CAS Project for Young Scientists in Basic Research (Grant No. YSBR-008), CAS Special Research Assistant Project (Grant No. E4559305), Postdoctoral Fellowship Program of CPSF (Grant No. GZB20230813) and China Postdoctoral Science Foundation (Grant No. 2024M763479).

References

1 Le Ny J, Pappas G J. Differentially private filtering. IEEE Transactions on Automatic Control, 2014, 59: 341–354

2 Li X, Meng M, Hong Y, et al. A survey of decision making in adversarial games. Science China Information Sciences, 2024, 67, Article 141201

3 Zhang X, Yuan Z, Xu S, et al. Secure perception-driven control of mobile robots using chaotic encryption. IEEE Transactions on Automatic Control, 2024, 69: 2429–2436

4 Lu Y, Zhu M. On privacy preserving data release of linear dynamic networks. Automatica, 2020, 115, Article 108839

5 Wang Y. Privacy-preserving average consensus via state decomposition. IEEE Transactions on Automatic Control, 2019, 64: 4711–4716

6 Kawano Y, Cao M. Design of privacy-preserving dynamic controllers. IEEE Transactions on Automatic Control, 2020, 65: 3863–3878

7 Gao C, Wang Z, He X, et al. Sampled-data-based fault-tolerant consensus control for multi-agent systems: A data privacy preserving scheme. Automatica, 2021, 133, Article 109847

8 Tan J, Wang J, Zhang J F. Cooperative secure parameter identification of multi-participant ARX systems — a threshold paillier cryptosystem-based least-squares identification algorithm. SCIENTIA SINICA Informationis, 2023, 53: 2472–2492

9 Tong Y, Feng Q, Luo M, et al. Multi-party privacy-preserving decision tree training with a privileged party. Science China Information Sciences, 2024, 67, Article 182303

10 Wei K, Li J, Ma C, et al. Gradient sparsification for efficient wireless federated learning with differential privacy. Science China Information Sciences, 2024, 67, Article 142303

11 Miao Y, Kuang D, Li X, et al. Efficient privacy-preserving federated learning under dishonest-majority setting. Science China Information Sciences, 2024, 67: 1–2

12 Ye M, Hu G, Xie L, et al. Differentially private distributed nash equilibrium seeking for aggregative games. IEEE Transactions on Automatic Control, 2022, 67: 2451–2458

13 Wang Y, Başar T. Ensuring both almost sure convergence and differential privacy in Nash equilibrium seeking on directed graphs. IEEE Transactions on Automatic Control, 2024, 69: 5478–5485

14 Wang Y, Nedić A. Tailoring gradient methods for differentially private distributed optimization. IEEE Transactions on Automatic Control, 2024, 69: 872–887

15 Lu Y, Zhu M. Privacy preserving distributed optimization using homomorphic encryption. Automatica, 2018, 96: 314–325

16 Moradi A, Venkategowda N K D, Talebi S P, et al. Privacy-preserving distributed Kalman filtering. IEEE Transactions on Signal Processing, 2022, 70: 3074–3089

17 Weng C, Nekouei E, Johansson K H. Optimal privacy-aware dynamic estimation. IEEE Transactions on Automatic Control, DOI:10.1109/TAC.2025.3565931, 2025

18 Nekouei E, Sandberg H, Skoglund M, et al. A model randomization approach to statistical parameter privacy. IEEE Transactions on Automatic Control, 2023, 68: 839–850

19 Feng Z, Nekouei E. A privacy-preserving framework for cloud-based HVAC control. IEEE Transactions on Control Systems Technology, doi: 10.1109/TCST.2024.3487019, 2024: 1–15

20 Shang J, Chen T. Linear encryption against eavesdropping on remote state estimation. IEEE Transactions on Automatic Control, 2023, 68: 4413–4419

21 Chen W, Liu L, Liu G P. Privacy-preserving distributed economic dispatch of microgrids: A dynamic quantization-based consensus scheme with homomorphic encryption. IEEE Transactions on Smart Grid, 2023, 14: 701–713

22 Farokhi F, Sandberg H. Fisher information as a measure of privacy: Preserving privacy of households with smart meters using batteries. IEEE Transactions on Smart Grid, 2018, 9: 4726–4734

23 Dwork C, Roth A. The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science, 2013, 9: 211–407

24 Zhang J F, Tan J, Wang J. Privacy security in control systems. Science China Information Sciences, 2021, 64, Article 176201

25 Wang J, Zhang J F, He X. Differentially private distributed algorithms for stochastic aggregative games. Automatica, 2022, 142, Article 110440

26 Liu C, Johansson K H, Shi Y. Distributed empirical risk minimization with differential privacy. Automatica, 2024, 162, Article 111514

27 Le Ny J. Differential Privacy for Dynamic Data. Cham, Switzerland: Springer, 2020

28 Le Ny J, Mohammady M. Differentially private MIMO filtering for event streams. IEEE Transactions on Automatic Control, 2018, 63: 145–157

29 Degue K H, Le Ny J. Differentially private Kalman filtering with signal aggregation. IEEE Transactions on Automatic Control, 2023, 68: 6240–6246

30 Liu Y H, Lee S H, Khisti A. Information-theoretic privacy in smart metering systems using cascaded rechargeable batteries. IEEE Signal Processing Letters, 2017, 24: 314–318

31 Issa I, Wagner A B, Kamath S. An operational approach to information leakage. IEEE Transactions on Information Theory, 2020, 66: 1625–1657

32 Farokhi F, Sandberg H. Ensuring privacy with constrained additive noise by minimizing Fisher information. Automatica, 2019, 99: 275–288

33 He J, Cai L, Guan X. Preserving data-privacy with added noises: Optimal estimation and privacy analysis. IEEE Transactions on Information Theory, 2018, 64: 5677–5690

34 Farokhi F. Privacy-preserving constrained quadratic optimization with Fisher information. IEEE Signal Processing Letters, 2020, 27: 545–549

35 Ziemann I, Sandberg H. Parameter privacy versus control performance: Fisher information regularized control. In: American Control Conference, 2020, Denver, CO, USA. 1259–1265

36 Kitanidis P K. Unbiased minimum-variance linear state estimation. Automatica, 1987, 23: 775–778

37 Darouach M, Zasadzinski M. Unbiased minimum variance estimation for systems with unknown exogenous inputs. Automatica, 1997, 33: 717–719

38 Nekouei E, Sandberg H, Skoglund M, et al. Optimal privacy-aware estimation. IEEE Transactions on Automatic Control, 2022, 67: 2253–2266

39 Li S, Khisti A, Mahajan A. Information-theoretic privacy for smart metering systems with a rechargeable battery. IEEE Transactions on Information Theory, 2018, 64: 3679–3695

40 Shao J. Mathematical Statistics. 2 edition. New York, NY, USA: Springer, 2003

41 Malagò L, Pistone G. Information geometry of the Gaussian distribution in view of stochastic optimization. In: Proceedings of the ACM Conference on Foundations of Genetic Algorithms XIII, 2015, Aberystwyth, United Kingdom. 150–162

42 Higham N J. Accuracy and Stability of Numerical Algorithms. 2 edition. Philadelphia, PA, USA: SIAM, 2002

43 Zhang F. The Schur Complement and Its Applications. New York, NY, USA: Springer, 2005

44 Grant M, Boyd S. CVX: Matlab software for disciplined convex programming, version 2.1. https://cvxr.com/cvx, 2014