Title

A Numerical Verification Framework for Differential Privacy in Estimation

Author

Han, Yunhai

Publication Date

2021

UNIVERSITY OF CALIFORNIA SAN DIEGO

A Numerical Verification Framework for Differential Privacy in Estimation

A Thesis submitted in partial satisfaction of the requirements
for the degree Master of Science

in

Engineering Sciences (Mechanical Engineering)

by

Yunhai Han

Committee in charge:

Professor Sonia Martínez, Chair
Professor Jorge Cortés
Professor Nick Gravish

2021

The Thesis of Yunhai Han is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2021

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

First of all, I want to express my great gratitude towards my advisor Prof. Sonia Martínez for giving me the opportunity of working on this project. This research is inspired by her previous work on designing a novel moving horizon estimator that also satisfies the requirement of differential privacy. Her continuous support and detailed guidance are indispensable for me to accomplish this work and write down all the components in the thesis.

Second, I would like to thank all my friends who gave me the company through the hard time. Without them, I would probably get stuck in some mental illness since everything changed drastically after the wide spread of Covid-19.

The last but not least, I am also indebted to my thesis committee members, Prof. Jorge Cortés and Prof. Nick Gravish for attending my thesis defense with valuable suggestions.

This work is also being prepared for a publication: Yunhai Han; Sonia Martínez. The thesis author will be the primary author of the paper.

VITA

2019	Bachelor of Engineering in Mechanical Engineering, Yanshan University
2019-2021	Graduate Teaching&Research Assistant, UC San Diego
2021	Master of Science in Mechanical Engineering, UC San Diego

PUBLICATIONS

Han. Y, Liu. F and M. C. YIP, ”A 2D Surgical Simulation Framework for Tool-Tissue Interaction”, *IROS Workshop*, 2020.

Han. Y, Liu. Y, Paz. D, and Christensen. I. H, ”Auto-calibration Method Using Stop Signs for Urban Autonomous Driving Applications”, *ICRA*, 2021.

Liu. F, Li. Z, Han. Y, J Lu, F Richter and M. C. YIP, ”*Real-to-Sim* Registration of Deformable Soft Tissue with Position-Based Dynamics for Surgical Robot Autonomy”, *ICRA*, 2021.

ABSTRACT OF THE THESIS

A Numerical Verification Framework for Differential Privacy in Estimation

by

Yunhai Han

Master of Science in Engineering Sciences (Mechanical Engineering)

University of California San Diego, 2021

Professor Sonia Martínez, Chair

This work proposes a verification framework for detecting violations of differential privacy for dynamic systems. Differential privacy aims to protect the privacy of the inputs of a mechanism so an adversary can not obtain relevant information about any of them by analyzing its outputs. The framework evaluates the differential privacy of a dynamic system mechanism. An event is defined as a subset of the state space. Considering the outputs of the mechanism (continuous-space) state estimates, the number of events required to perform the test is infinite. Thus, to obtain a tractable test, we limit events containing the outputs up to a given resolution. Further, to limit the effect of long-time horizons, we restrict events to those which will contain the outputs with high probability using a data-driven scenario approach. Finally, a statistical hypothesis test is employed to detect the violations of differential privacy. In order to find the event that is most likely to disclose the violations, one event is chosen based on the test values. Numerical simulation results of W_2-Moving-Horizon-Estimator and Extended Kalman Filter are performed and evaluated using this framework. The results demonstrate that the differential privacy is achieved at the cost of inaccuracy.

Chapter 1

Introduction

Recently, a rapidly growing number of emerging systems, such as smart grids or intelligent transportation systems, require data from particular sensors or users (smart sensors or vehicle GPS) to continuously estimate the current states of the target system. Commonly, the estimation results are more accurate in the benefit of the online dynamic data, which is good for serving the tasks of monitoring or control. For example, in a smart grid application, the customer could receive better rates if the utility company tracks his instantaneous power consumption data, because it helps to improve the demand forecast algorithm. However, for the privacy reasons, the data senders who benefit from these systems want to preserve their individual information as much as possible. In other words, only the strictly necessary information should be released. Moreover, even though the released data is anonymous, the privacy is not guaranteed due to the public side information. This has been already demonstrated in [1], in which the researchers found that although the released datasets provided by Netflix was shown to preserve customer privacy, it is still possible to identify individual users by using side information from other third-parties, such as public recommendation systems. Traffic monitoring system [2] that measures the person's position from their smartphones is another example where the person's position traces can be identified by the correlation between their locations and residences. Hence, several researchers have developed privacy preserving mechanisms to address these problems in order to promote the applications of emerging systems.

Precisely defining what constitutes the privacy in a mathematical notion is a tough work. *Differential Privacy* [3], which was initially used in the database literature, is a successful definition and has gradually become as a standard privacy specification [4]. The differentially private algorithm randomizes the system outputs in such a way that the distribution of the outputs is not too sensitive to the system inputs provided by any single senders. As a result, it is more difficult for the adversaries to make any inferences about individuals based on the system outputs and side information [5]. It has already been applied in the commercial products provided by technological giants, such as Apple or Google. They announce that their products are able to learn from a group as much as possible while keeping minimum knowledge of any members in it. In the recent decades, the notion of differential privacy has received increasing attention from the researchers in the field of control and system. They take advantage of the clear definition to design privacy-preserved estimators for various tasks, including control systems [6], network topology [7] and estimation and filtering [8]. The work [9] focuses on the development of differentially-private estimator, which introduces the concept of differential privacy into Kalman filter design. In [10], the researchers modifies the optimization-based estimators via a perturbed objective function for the desired level of differential privacy. The concise and broad overview of the differential privacy in control system can be found in [11].

However, the design of these algorithms is very subtle and error-prone. It has been proved in the database literature that a large number of proposed algorithms (published) are incorrect [12] [13], which means the claimed level of differential privacy can not be achieved. In other words, the individual private information is potentially released for the undesirable usage even though the data is processed using these algorithms. In [14], they find an approach to detect the violations of differential privacy in several sophisticated differentially private algorithms. They build a numerical method of evaluating these algorithms and verifying the correct ones in which the claimed level of differential privacy is satisfied. However, their work is only limited to the database applications. To the best of the authors' knowledge, currently, there is no published method that is developed for the same purpose but targets at the control tasks. It disagrees with the fact that a large number of differentially private estimators are being studied on.

The main contribution of this paper is to build a numerically verification framework that can detect the violations of differential privacy in several estimators. We redefine the procedures in [14] so that it can be applied in the online estimation systems. The statistical nature of the framework is similar as illustrated in [14]: it implements the candidate estimator many times and then uses a statistical test to evaluate whether or not the claimed level of differential privacy is satisfied. With the help of this framework, it can greatly reducerobotics the burden for the other researchers or engineers of selecting the appropriate differentially private estimators in a given task.

The rest of the thesis is organized as follows: Chapter 2 gives the problem motivation; Chapter 3 includes all the necessary background knowledge. Chapter 4 presents the technical details of this framework; Chapter 5 shows the simulation results of a linear oscillator with a nonlinear observation model; Chapter 6 includes the conclusions.

Chapter 2

Problem Motivation

In this chapter, we motivate and introduce the concept of differential privacy. In the next chapter, we will formulate it mathematically in the context of estimation.

Consider the problem of a sensor network performing a distributed estimation task. Sensors may have different owners, who wish to maintain their locations anonymous. Even if the communication network among sensors is secure, the estimation of the target may be widely available due to public interest. Thus, an adversary who have access to additional side information[1] may deduce valuable knowledge about a particular sensor, which should be secured for some purposes. Fig. 2.1 illustrates a situation when the adversary can estimate the location of a particular sensor (e.g. via a Bayes rule) by analyzing the released measurement data, target's trajectory and other sensors' location.

To prevent this from happening, differential privacy applied on estimation aims to make it hard to distinguish between sequences of adjacent, noise-free data $\mathrm{y}_{0:T} \in \mathbb{R}^{d\times T}$, provided by adjacent sensor locations. This will be made more precise in the following chapters.

By "perturbing" the outputs of an estimator, differential privacy guarantees no individual sensor location can be deduced up to a certain degree. It is not hard to imagine that the amount

[1] About side information: it can be anything that helps with the estimation; for example something about the target's true location, or a restricted region of space where the adversary knows where the sensor can be.

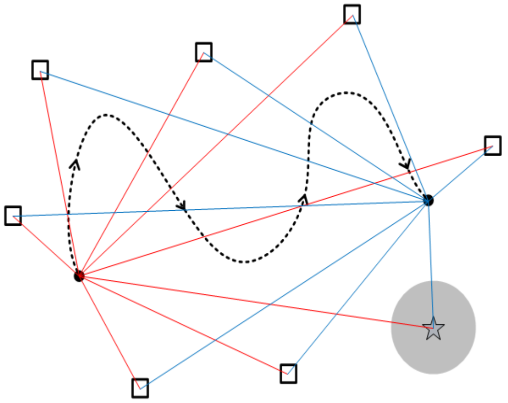


Figure 2.1: A demonstration of estimating a particular sensor location. The solid circle represents the moving target that is being estimated; the squares represent the location of known sensors (side information); the star represents the deduced location of the particular sensor, The actual location can be anywhere within the shaded circle (hypothesis). The diameter of the hypothesis depends on the level of differential privacy of the estimator. The dashed curve represents the target's trajectory and the arrows indicate the direction. The set of red/blue lines represent the output data released from sensors when the target is at the start/end point. In application, the adversary can probably achieve the sensor outputs at more than two time steps.

of perturbation influences both the accuracy and the level of privacy of the results, so there always exists a trade-off between them. In estimation, sensor output data may already be affected by natural system and sensor measurement noise. However, additional perturbations may be necessary to guarantee differential privacy. This concept was applied in [10] in the context of filtering and estimation, where theoretical sufficient conditions for differential privacy have been found. However, these conditions are conservative. Thus, in this work we aim to investigate the following questions:

1. Produce a numerical test procedure to evaluate the differential privacy of an estimation method for a dynamic system; while providing probability guarantees of its correctness.

2. Evaluate numerically the differential-privacy properties of the W_2 filter introduced in [10]; compare its performance with that of an extended Kalman filter.

3. Evaluate the differences in privacy/estimation when the perturbations are directly applied

to the measurement data before the filtering process is done.

Chapter 3

Background Knowledge

3.1 Differential Privacy in Database

Originally, differential privacy refers to a system for publicly sharing information about a dataset while withholding information about individuals in the dataset. The core behind differential privacy is that if the influence of making a single substitution in the database is small enough, the query result cannot be used to infer much about any single individual, thus providing privacy. An alternative way to describe differential privacy is applying a constraint on the algorithms used to publish aggregated information about a target database. This constraint helps to limit the disclosure of private information of records whose information is in the database. For instance, differentially private algorithms are used by some government offices to publish statistical information while ensuring confidentiality of each record, and by companies to collect information about user behavior while controlling what is visible even to internal analysts.

Roughly speaking, an algorithm is differentially private if an observer seeing its output cannot decide if a particular individual's information was used in the computation. Differential privacy is usually discussed in identifying individuals whose information may be in a database. Although it does not directly refer to re-identification attacks, differentially private algorithms probably resist such attacks [15].

Differential privacy was developed by cryptographers and thus is often associated with the applications in database searching. Here, we first formulate it mathematically in the context of database.

We can view a database as a finite set of records generated from some domain. Differential privacy replies on the concept of adjacent databases. Usually, the two most common definitions of adjacency with respect to database are: 1). two databases D_1 and D_2 are adjacent if D_1 can be obtained from D_2 by removing or adding one record; 2). two databases D_1 and D_2 are adjacent if D_1 can be obtained from D_2 by modifying one record. We write $D_1 \sim D_2$ to indicate D_1 is adjacent to D_2. The term mechanism is referred to an algorithm $\mathcal{M}$ that aims to protect the differential privacy of its inputs. Suppose this mechanism satisfies the ε-differential privacy:

Definition 1 (ε-Differential Privacy [15])**.** *Let $\varepsilon > 0$, a mechanism is said to satisfy ε-differential privacy if for every pair of adjacent databases D_1 and D_2, and every subset $E \subseteq Range(\mathcal{M})$,*

$$\mathbb{P}(\mathcal{M}(D_1) \in E) \leq e^{\varepsilon} \cdot \mathbb{P}(\mathcal{M}(D_2) \in E)$$

Here, $\mathbb{P}(\mathcal{M}(D_1) \in E)$ and $\mathbb{P}(\mathcal{M}(D_2) \in E)$ are the probabilities that the numerical queries infer the same record when searching the adjacent databases . The value of ε controls the level of privacy and the smaller ε is, the better privacy can be expected.

One of the famous ε-differentially private algorithms is the Laplace mechanism, which is used to assign the property of differential privacy to any numerical queries. If D is the target database, a numerical query is a function $q : D \to \mathbb{R}^k$ (i.e. it outputs a k dimensional vector to identify the database). Laplace mechanism works by adding the Laplace noise to the query answers:

Definition 2 (Laplace Mechanism [15]). *For any query q, the Laplace mechanism outputs:*

$$\mathcal{M}(D,q,\varepsilon) = q(D) + (\eta_1, \cdots, \eta_k)$$

Where each η_i are independent random variables with relation to the value ε. In [16], it is proved to be ε-differential private.

Adopting the same ideas, we can then formulate differential privacy in the context of estimation.

3.2 Differential Privacy in Estimation

The importance of the differential privacy with respect to estimation has been shown in Chapter 2. Here, we formulate it in a mathematical manner.

In this work, the dynamics and the observation model that we consider are given as:

$$\Omega : \begin{cases} x_{k+1} = f\left(x_k, w_k\right), \\ y_k = h\left(x_k\right) + v_k. \end{cases} \tag{3.1}$$

where, $x_k \subset \mathbb{R}^{dx}, y_k \subset \mathbb{R}^{dy}$, $w_k \subset \mathbb{R}^{dw}$ and $v_k \subset \mathbb{R}^{dv}$. w_k and v_k represents the process noise and measurement noise at time step k, respectively. We make two assumptions:

1. (Lipschitz continuity). The functions f and h are Lipschitz continuous, with $\|f\left(x_1, w_1\right) - f\left(x_2, w_2\right)\| \leq c_f \|x_1 - x_2\| + c_n \|w_1 - w_2\|$ and $\|h\left(x_1\right) - h\left(x_2\right)\| \leq c_h \|x_1 - x_2\|$.

2. (Noise characteristics). The noise sequences $\{w_k\}_{k\in\mathbb{N}}$ and $\{v_k\}_{k\in\mathbb{N}}$ are independent samples from distributions ω and ν with $|w|$ and $|v|$ bounded. Moreover, we assume that $\mathbb{E}_{\omega}\left[w_k\right] = 0$ and $\mathbb{E}_{\nu}\left[v_k\right] = 0$. In order to drop the noise variables from the optimization (will be introduced later), we describe the autonomous system corresponding to Eqn. 3.1

$$\Sigma : \begin{cases} x_{k+1} = f\left(x_k, 0\right) = f_0\left(x_k\right), \\ y_k = h\left(x_k\right). \end{cases} \tag{3.2}$$

Let $\{0, \ldots, T\}$ be the time horizon, and denote by $\mathrm{y}_{0:T} = (y_0, \ldots, y_T)$ the sensor output data up to time T and by $\Sigma_T(x) = \left(h(x), h \circ f_0(x), \ldots, h \circ f_0^T(x)\right)$ the output mapping from system Σ.

We can now more formally define ε-δ differential privacy for a state estimator $\mathcal{M}$ up to time T of a system as in Eqn. 3.1. In doing this, we interpret the estimator or mechanism $\mathcal{M}$ as a mapping $\mathcal{M} : \mathbb{R}^{(T+1)d_Y} \to \mathbb{R}^{d_X}$, which assigns an output sensor data $y_{0:T}$ of the autonomous system in Eqn. 3.2 to a state estimate. This mapping is stochastic, with randomness induced by the noises that are present in Eqn. 3.1.

Definition 3 (**ε-δ Differential Privacy).** *Suppose that $\mathcal{M}$ is a state estimator of system in Eqn. 3.1. Given $\delta \in \mathbb{R}_{>0}$, the estimator $\mathcal{M}$ satisfies ε-δ differential privacy (for some $\varepsilon \in \mathbb{R}_{>0}$) up to time T if for any two δ-adjacent sensor output data $\mathrm{y}^1_{0:T}, \mathrm{y}^2_{0:T}$ ($d_{\mathrm{y}}\left(\mathrm{y}^1_{0:T}, \mathrm{y}^2_{0:T}\right) \leq \delta$)), we have*

$$\mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^1_{0:T}\right) \in E\right) \leq \mathrm{e}^{\varepsilon}\mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^2_{0:T}\right) \in E\right), \forall E \subset range(\mathcal{M}). \tag{3.3}$$

Here, $\mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^1_{0:T}\right) \in E\right)$ and $\mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^2_{0:T}\right) \in E\right)$ are the probabilities that the estimation result falls into the same event E by using adjacent sensor output data $\mathrm{y}^1_{0:T}$ and $\mathrm{y}^2_{0:T}$, respectively.

Note that $d_{\mathrm{y}}\left(\mathrm{y}^1_{0:T}, \mathrm{y}^2_{0:T}\right)$ is a general expression that evaluates the distance between two outputs of the estimator up to time T. Under the assumptions of Lipschitz continuity of h, adjacency in sensor measurements (in the sense of the 2-norm) will translate into adjecency of the corresponding output data in the sense of the 2-norm. We elaborate on this in Chapter 5, where we employ the L_2 norm in Euclidean space to compute it. Note also that, instead of focusing on a single time step, we require sequences of sensor output data as we are interested in evaluating the performance along a complete time horizon.

3.3 W_2-Moving-Horizon-Estimator (W_2MHE)

In this section, and for the sake of completeness, we provide the background of Wasserstein Moving Horizon Estimator (W_2-MHE) and refer readers to [10], for its asymptotic stability and robustness properties. The estimator can be further modified by adding an entropy operator to attain the differential privacy, which is used to be against data integrity attacks.

It is worth emphasizing that the proposed framework is not specific to any estimator: readers should be aware that the following theory can easily be adopted to other estimators, such as the differentially private Kalman filter in [9].

3.3.1 Moving-Horizon Estimator

The moving-horizon estimation method provides an estimation method for nonlinear systems which are stable under bounded disturbances [17], [18] [19]. It treats the state estimation as an optimization problem.

Using the notations in Eqn. 3.2, the full-horizon state estimation (FIE) problem aims at computing x_0 based on given $\mathrm{y}_{0:T}$ and $\Sigma_T(x_0)$, which is an inverse optimization problem. To formulate the inverse problem, we consider the function $J_T\left(\mathrm{y}_{0:T}, \Sigma_T(x_0)\right) = \sum_{k=0}^{T} \|y_k - h \circ f_0^k(x_0)\|^2$, such that $J_T\left(\mathrm{y}_{0:T}, \Sigma_T(x_0)\right) = 0$ if and only if $\Sigma_T(x_0) = \mathrm{y}_{0:T}$. Now, the estimation problem becomes:

$$x_0 \in \arg\min_{x \in \mathbb{R}^{d_X}} J_T\left(\mathrm{y}_{0:T}, \Sigma_T(x)\right).$$

In the above equation, $\mathrm{y}_{0:T}$ is given. Under some conditions, a local minimum of $J_T\left(\mathrm{y}_{0:T}, \Sigma_T(x_0)\right)$ is also global; see [10] and Theorem 1 in [20].

By means of a moving window of length N, one can additionally assimilate the new sensor outputs online. In this way, at any time step k, we use $\mathrm{y}_{k:k+N}, \Sigma_N(x_k)$, and the previous state estimation x_{k-1} to obtain the next state estimate x_k as follows:

$$x_k \in \arg\min_{x\in\mathbb{R}^{d_X}} \gamma\left(f_0\left(x_{k-1}\right),x\right) + G_k^N\left(x_k\right), \tag{3.4}$$

where, $\gamma(f_0(x_{k-1}),x)$ measures the difference between the estimation x and system evolution $f_0\left(x_{k-1}\right)$. This term can also be seen as a prediction by the filter. Here, $G_k^N\left(x_k\right) = J_N\left(\mathrm{y}_{k:k+N}, \Sigma_N\left(x_k\right)\right)$ is the correction within the moving window. In other words, Eqn. 3.4 indicates the core component of a prediction & correction online estimation algorithm.

Differential privacy can be easily addressed by lifting the estimation to the space of probability distributions. This is summarized in the following subsections.

3.3.2 Distributional Moving-Horizon Estimator

In the previous subsection, we presented a recursive moving-horizon estimator. We now lift the estimates to the space of probability distributions over $\mathbb{R}^{d_X}$. This allows for the consideration of more general noise distributions and can easily account for differential privacy. In this way, the expectation of the objective function in Eqn. 3.4 should be minimized. In the following, we use $\mathbb{X}$ to denote the sample space of the state variable. Thus, the estimation results can be encoded via probability measures over $\mathbb{X}$, a set we denote by $\mathcal{P}(\mathbb{X})$.

The new optimization problem is described as

$$\mu_k \in \arg\min_{\mu\in\mathcal{P}(\mathbb{X})} D\left(\mu, f_{0\#}\mu_{k-1}\right) + \mathbb{E}_\mu\left[G_k^N\right],$$

where $D : \mathcal{P}(\mathbb{X}) \times \mathcal{P}(\mathbb{X}) \to \mathbb{R}_{\geq 0}$ is a metric or divergence providing a measure of distance on $\mathcal{P}(\mathbb{X})$. Different choices of D result in different estimators. In this work, we study Wasserstein metric W_2, which exactly yields the fast moving-horizon estimator [21] given as:

$$\mu_k \in \arg\min_{\mu\in\mathcal{P}(\mathrm{X})} \frac{1}{2} W_2^2\left(\mu, f_{0\#}\mu_{k-1}\right) + \mathbb{E}_\mu\left[G_k^N\right]. \tag{3.5}$$

An implementable version of this estimator is derived in [10], which uses Monte Carlo methods to sample from μ_k:

$$x_k \in \arg\min_x \frac{1}{2}\left\|x - f_0\left(x_{k-1}\right)\right\|^2 + G_k^N(x), \quad k > 0.$$

The initial condition x_0 can be sampled from the initial probability distribution $\mathcal{P}_0(\mathbb{X})$. The number of particles is adjustable and the final estimation result are the average of all the particles.

The asymptotic stability and the robustness of W_2-MHE are guaranteed under certain assumptions in [10]. In the ensuing subsection, an entropy term is added into the objective function for the purpose of differential privacy.

3.3.3 Differentially-private W_2-MHE

Compared with the regular W_2-MHE, differential privacy holds if the objective function in Eqn. 3.5 is modified with an entropy term as follows:

$$\begin{aligned}\mu_k \in \arg\min\nolimits_{\mu\in\mathcal{P}(\mathbb{X})} &\left[\tfrac{1}{2} s_k W_2^2\left(\mu, f_{0\#}\mu_{k-1}\right) + s_k \mathbb{E}_\mu\left[G_k^N\right]\right. \\ &\left. - \left(1 - s_k\right) S^{\mathcal{K}_k}(\mu)\right].\end{aligned} \tag{3.6}$$

Where $s_k \in [0,1]$ is a tunable time-variant parameter for all k. In applications, the values depend on the level of differential privacy that is expected to maintain. Moreover, $S^A(\mu) = \int_A \rho \log(\rho)\, \mathrm{dvol}$, where $A \subset \mathbb{X}$ and $d\mu = \rho$ dvol with ρ being the corresponding density function and vol being the Lebesgue measure. For n-dimensional Euclidean space, if $n = 1, 2$, or 3, Lebesgue measure coincides with the measure of length, area, or volume. In Eqn. 3.6, $\mathcal{K}_k$ is the support of $f_{0\#}\mu_{k-1}$ (with $\mathcal{K}_0$ being the support of μ_0). We note that when $s_k = 0$, the problem reduces to an entropy maximization problem, which yields a uniform distribution over $f_0(\mathcal{K}_{k-1})$ as the solution. The uniform distribution is insensitive to the sensor outputs, so it provides with maximum level of differential privacy, while being of no use for the estimation. Similarly, when $s_k = 1$, Eqn. 3.6 reduces to Eqn. 3.5, which maximizes the accuracy but maintain no differential privacy. Thus, it is important to find the best upper bound of s_k to guarantee the desired level of privacy. Also, the goal targets at full time horizon $\{0,\ldots,T\}$, so the the upper bound should be applied on a sequence of values $\{s_k\}_{k=1}^{T}$.

The optimization problem in Eqn. 3.6 can be rewritten as follows:

$$\begin{aligned} \mu_k \in \arg\min\nolimits_{\mu\in\mathcal{P}(\mathbb{X})} \Big[\tfrac{1}{2}W_2^2\left(\mu, f_{0\#}\mu_{k-1}\right) + \mathbb{E}_\mu\left[G_k^N\right] \\ -\left(\tfrac{1-s_k}{s_k}\right) S^{\mathcal{K}_k}(\mu)\Big]. \end{aligned} \tag{3.7}$$

With a theoretical justification provided in [10], the following upper bound condition on $\{s_k\}_{k=1}^{T}$ is sufficient for the ε-δ differential privacy of the modified W_2-MHE in Eqn. 3.7 over full time horizon:

$$\sum_{k=1}^{T}\left(\frac{s_k}{1-s_k}\right)c_f^k \leq \frac{\varepsilon}{l\delta\,\mathrm{diam}\left(\mathcal{K}_0\right)}. \tag{3.8}$$

Here, c_f^k denotes k-th power of Lipschitz constant for the discrete dynamics system $f_0(x)$; l denotes the l-smoothness constant of function G_k^N [1]; and $\mathcal{K}_0$ is the support of the initial distribution and $\mathrm{diam}\left(\mathcal{K}_0\right)$ can be approximated by the diameter of the distribution.

We note that for a given ε, the upper bound on $\{s_k\}_{k=1}^{T}$ decreases with δ. Hence, the same level of ε-δ differential privacy for more divergent adjacent sensor outputs (larger δ) requires an increase of the proportion of entropy term, resulting in less accuracy. With one more step, Eqn.

[1] It is assumed that for any $x,y\in\mathbb{R}$, we have $\|\nabla G_k^N(y) - \nabla G_k^N(x)\| \leq l\|y-x\|$.

3.8 can be turned into:

$$\varepsilon \geq l\delta \mathrm{diam}(K_0) \sum_{k=1}^{T} \frac{s_k}{1-s_k} c_f^k. \tag{3.9}$$

This shows that given $\{s_k\}_{k=1}^{T}$, what the expected level of privacy from the estimator is. In Chapter 5, we will correlate the simulation results with theoretical results from Eqn. 3.9. However, the conditions considered in [10] are only sufficient and may be restrictive. This is what we aim to test by means of hypothesis tests.

3.4 Hypothesis testing

The most important component of our proposed approach is based on a hypothesis testing procedure. Hence, we give a brief introduction of this procedure. Under some conditions, tests can guarantee that the probability of making a mistake with test rule is small with high probability. This probability can be calculated exactly for finite samples in some cases, or, with high-confidence as a function of the number of samples, with the help of Large Number theory. Readers can find more about hypothesis test in the chapter 8 of [22] and [23]. For more information, we provide a brief description of this next.

A statistical hypothesis procedure is a claim about a parameter of a distribution that generates data. The test provides a rule based on samples of the distribution to decide whether a null hypothesis (denoted by H_0) is accepted as true or rejected as false.

The decision rule of the test is given in terms of a test statistic, or function of the sampled data $W(X_1,\ldots,X_n)$, and a rejection region R. For example, this can be expressed as $W(X_1,\ldots,X_n) \in R$, which implies that the null hypothesis should be rejected. The evaluation of the hypothesis test is done via the probability of making a mistake when accepting or rejecting the null hypothesis. This leads to the so-called Type I and Type II errors. A Type I test error occurs when the test incorrectly rejects the null hypothesis when it is indeed true. If, on the other hand, the test accepts the null hypothesis while it should be rejected, then the Type II error occurs. The common practice is to control only the Type I error because it is the most important. Mathematically, the Type I error is defined as the probability $P(W(X_1,\dots,X_n) \in R|H_0)$. The Type I error also defines the so-called p-value, that is, $p = P(W(X_1,\dots,X_n) \in R|H_0)$. The significance level of the test is a constant α (typically 0.1, 0.05 or 0.01) which is used to decide whether to accept or reject H_0. If p is such that $p \leq \alpha$, then the probability that a mistake is made by rejecting the null hypothesis is very small. In other words, it implies the stronger the evidence that the null hypothesis should be rejected when our test statistic falls into the rejection region. On the other hand, if $p > \alpha$, the test should accept the null hypothesis. Usually the p-value can not be computed exactly, except for the so-called exact tests. Using the Large Number theory, one can approximate the p-values for some tests.

The Fisher's exact test [24] is an exact test on binomial distributions. Let c_1, c_2 be two samples from two binomial distributions B(n_1, p_1) and B(n_2, p_2), respectively. Here, p_1 and p_2 are two unknown parameters and let $s = c_1 + c_2$. The goal is to test the null hypothesis $H_0 : p_1 \leq p_2$, given the total $s = c_1 + c_2$ and knowing the parameters n_1, n_2. That is, one aims to evaluate whether or not the number of successes according to the first distribution happens in the same proportion as the number of successes according to the second distribution. For the special case $H_0 : p_1 = p_2$, and taking $s = c_1 + c_2$, n_1 and n_2 as given, the p-value of an extreme event is equal to 1 - Hypergeometric.cdf($c_1 - 1|n_1 + n_2, n_1, s$)[2] When $p_1 < p_2$, the p-value of an extreme event can not be exactly computed without knowing p_1 and p_2 [24], but it is still bounded by 1 - Hypergeometric.cdf($c_1 - 1|n_1 + n_2, n_1, s$). Thus, this quantity is used as a surrogate p-value. This value should be larger than the chosen level α to accept the null hypothesis. Note that this result applies for small sample sizes because the test is *exact*.

[2] This notation represents the cumulative distribution function of a hypergeometric distribution.

3.5 Reachable Set Analysis

In this section, we show how to employ reachable set analysis and their approximation to determine the set of events that will be considered in our test approach later. This will lead to the verification of ε-δ differential-privacy with high confidence.

In most cases, the computation of the exact reachable set $R_{[t_0,t_k]}$ for a dynamical system (see Def. 4) is intractable. Instead, one can attempt to obtain some approximation $\hat{R}_{[t_0,t_k]}$. We do this as in [25], which employs scenario optimization [26] to obtain approximations with probabilistic guarantees of correctness. In this way, the approximated reachable set can be guaranteed to contain a certain part of the actual one with high confidence.

We perform the approximation by finding a vector parameter $\boldsymbol{\theta}$, which represents a family of sets, and that can best capture the actual reachable set. For example, using ellipsoids, we consider:

$$\hat{R}_{[t_0,t_k]}(\boldsymbol{\theta}) = \{x : \|Ax-b\|_2 \leq 1\}. \tag{3.10}$$

Thus, the parameters reprensenting this are $\boldsymbol{\theta} = (A,b)$. It is not difficult to see how $\hat{R}_{[t_0,t_k]}(\boldsymbol{\theta})$ can provide either an over-approximation or under-approximation of $R_{[t_0,t_k]}$. Though the two kinds of approximations are useful to provide safety guarantees, an estimated reachable set from a set of samples does not in general correspond to either of these cases.

Consider a random variable $\mathcal{Z}$ with support on $R_{[t_0,t_k]}$. Then, its probability density function (pdf) will satisfy $p_{\mathcal{Z}}(x) = 0$ for x not within $R_{[t_0,t_k]}$ and $p_{\mathcal{Z}}(x) > 0$ for x within $R_{[t_0,t_k]}$. That is, $R_{[t_0,t_k]}$ is a set with probability one, and any other disjoint set defines an event with probability zero. Other sets that overlap with $R_{[t_0,t_k]}$ have a probability between zero and one. A higher probability indicates that it contains more parts of $R_{[t_0,t_k]}$. In this context, our goal is to approximate $R_{[t_0,t_k]}$ via a set that has a high probability under this distribution. In particular, we would like to obtain a β-accurate approximation, $\hat{R}_{[t_0,t_k]}(\boldsymbol{\theta})$ satisfying:

$$p_{\mathcal{Z}}(\hat{R}_{[t_0,t_k]}(\boldsymbol{\theta})) \geq 1-\beta.$$

Notice that it is still possible that a β-accurate approximation is quite conservative, which means in addition to the $1-\beta$ part of $R_{[t_0,t_k]}$, it can also contain a large portion of the state space outside. For most methods, the undesirable over-approximation can not be totally eliminated, but it can be minimized.

With this in mind, we define the following chance-constrained optimization problem that 1). computes a β-accurate reachable set; 2). minimizes the volume the estimated set:

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{minimize}} \quad & \text{Vol}\left(\hat{R}_{[t_0,t_k]}(\boldsymbol{\theta})\right), \\ \text{subject to} \quad & P_{\mathcal{Z}}\left(\hat{R}_{[t_0,t_k]}(\boldsymbol{\theta})\right) \geq 1-\beta. \end{aligned} \tag{3.11}$$

In general, this optimization problem is intractable. The work [27] discusses how to solve this problem approximately as follows.

First, using sample data, $R_{[t_0,t_k]}$ is approximated via an L_2 norm ball, which is an ellipsoid. As defined in Eqn. 3.10, $x, b \in \mathbb{R}^d, A \in \mathbb{R}^{d\times d}$. In general cases, we can consider that A is any symmetric matrix, which allows us to use $-\log\det A$ as a proxy for the volume of $\hat{R}_{[t_0,t_k]}(\theta)$. It can be proved that the value of $-\log\det A$ is directly proportional to the volume of the L_2 norm ball [25]. Using the L_2 norm ball with the proxy, the β-accurate reachable set that contains the least state space can be computed by solving:

$$\begin{aligned} \underset{A,b}{\text{minimize}} \quad & -\log\det A \\ \text{subject to} \quad & P_{\mathcal{Z}}(\|A\mathcal{Z}-b\|_2 - 1 \leq 0) \geq 1-\beta \end{aligned} \tag{3.12}$$

Note that $-\log\det A$ and $\|A\mathcal{Z}-b\|_2 - 1$ are both convex with respect to A and b, so Eqn. 3.12 defines a convex chance-constrained optimization problem. Via scenario approach, one can obtain the problem:

$$\begin{aligned} \underset{A,b}{\text{minimize}} \quad & -\log\det A \\ \text{subject to} \quad & \|Az^{(i)} - b\|_2 - 1 \leq 0, \quad i = 1, \ldots, \Gamma \end{aligned} \tag{3.13}$$

Note that in Eqn. 3.13, A is a $d \times d$ matrix and b is a d-vector, so the degree of freedom (Dof) of the decision variables in that equation can be computed by $d(d+1)/2+d$. Here comes the following theorem.

Theorem 1 ([25].). *Given γ, if Γ in Eqn. 3.13 satisfies*

$$\Gamma \geq \frac{1}{\beta}(\frac{e}{e-1})(\log\frac{1}{\gamma} + \frac{d(d+1)}{2} + d),$$

The global minimizer of Eqn. 3.13 is also a feasible solution of Eqn. 3.12 with probability $\geq 1-\gamma$, which indicates that the approximated reachable set $\boldsymbol{\theta} = (A,b)$ contains at least $1-\beta$ of the actual one while has the minimum area of state space.

Chapter 4

Differential Privacy Test Framework

4.1 Overview of the differential-privacy test framework

The algorithmic approach to test for differential privacy is based on the following steps and modules:

1. The evaluation of ε-δ differential-privacy for an estimator requires checking Eqn. 3.3 for an infinite number of events. Since this is not possible, the algorithm first computes a finite list of events through an EventListGeneration function. This function is based on 1) obtaining a sufficiently fine partition of the state space, and 2) implementing a reachability set analysis to obtain the most-likely event.

2. Once a finite set of events is generated, a second step is to identify the worst case event that leads with higher probability of violation of differential privacy. This is done via a WorstEventSelection module in the algorithm which also includes a hypothesis testing step.

3. A final step performs a hypothesis test with respect to the worst event via the HypothesisTest module.

Based on the above, we will construct the main components of an algorithm to test for ε-δ

differential privacy. The overall algorithm description is summarized below.

Algorithm 1 Overview of the Test Framework

```
function TEST FRAMEWORK($\mathcal{M}$, ε, $y^1_{0:T}, y^2_{0:T}$)
    Input: Target estimator($\mathcal{M}$)
            Desired differential privacy(ε)
            δ-adjacent sensor outputs($y^1_{0:T}, y^2_{0:T}$)
    EventList = EventListGenerator($\mathcal{M}$, $y^1_{0:T}$)
    WorstEvent = WorstEventSelector($\mathcal{M}$, ε, $y^1_{0:T}, y^2_{0:T}$,
    EventList)
    $p^+, p_+$ = HypothesisTest($\mathcal{M}$, ε, $y^1_{0:T}, y^2_{0:T}$,
    WorstEvent)
    Return $p^+, p_+$
end function
```

4.1.1 EventListGenerator

Here, we start by discussing the methods for generating the candidate list of events via the EventListGenerator module in the algorithm. This is based on the concept of *reachable sets* for dynamic systems.

Definition 4 (**Reachable Set for a dynamic system).**

The general definition of a dynamic system can be considered as a mapping from initial state $x_0 \subset \mathbb{R}^{dx}$ to a unique final state at time k under the influence of the system dynamics, and the bounded disturbances. Therefore, reachable set can be defined as: Suppose we are given with an initial set $X_0 \subset \mathbb{R}^{dx}$, and a set of disturbances $\omega \subset \mathbb{R}^{dw}$, then the forward reachable set can be defined as:

$$R_{[t_0,t_k]} = f(x_0, \omega_{t_0} : \omega_{t_k}), x_0 \subset \mathbb{R}^{dx}, \omega \subset \mathbb{R}^{dw}.$$

This indicates all the reachable states where the dynamic system can reach at time step k if it starts from a state within $\mathbb{R}^{dw}$ with bounded disturbance at each time step.

Our approach to reduce the set of events that need to be checked to guarantee ε-δ differential privacy consists of the following: a) approximating the reachable set of the dynamic system, and b) dividing the state space with a grid of a certain resolution. See the schematics of the function in Algorithm 2.

Algorithm 2 Event List Generator

```
function EventListGenerator($\mathcal{M}$, $y^1_{0:T}$)
    Input: Target Estimator($\mathcal{M}$)
           Sensor Output($y^1_{0:T}$)
    ReachableSet ← Approximate of reachable set
    at each time step $k$
    EventList ← Each event is composed of grids that
    are inside ReachableSet along all time steps
    Return EventList
end function
```

Traditional approaches to approximate the reachable set of a dynamic system require a careful analysis of the system dynamical equations, which can be difficult or conservative for general nonlinear systems. Hence, we adopt a recently proposed method in [25], which does this in a data-driven fashion as shown in Alg. 3 with probabilistic guarantees of correctness. The theoretical justifications can be found in Section 3.5.

In Alg. 3, Γ defines the sampling number that is necessary for guaranteeing the estimated reachable set contains $1-\beta$ of the actual distribution with confidence $1-\gamma$. In Section 3.5, we will discuss on how it works. Thus, we run the estimator Γ times and record the results. Finally, a scenario optimization problem is solved for the matrix A^k and the vector b^k, which encompasses an ellipsoid (ellipse if dimension $n = 2$).

In the reachable set, several grids are then generated according to the given resolution r. For example, if $r = 2$, there are two intervals along both x, y directions. Thus, in total, four grids are obtained at each time step k.

Each event E represents a combination of grids at all time steps. If $r = 2, T = 4$, the number of total events is $4^4 = 256$. The next step is to select the worst event E^*.

Algorithm 3 Reachable Set Approximation

```
Input: Target Estimator($\mathcal{M}$) with dimension $d$
        Measurement data($\mathrm{y}^1_{0:T}$)
        Probabilistic guarantee parameters $\beta, \gamma$
Output: Matrix $A^k$ and vector $b^k$ representing an
         $\beta$-accurate reachable set at time step $k$
         $R_k(A^k, b^k) = \left\{x : \|A^k x + b^k\|_2 \leq 1\right\}$
         with confidence $1 - \eta$.
Set number of samples $\Gamma$ =
        $\left[\frac{1}{\beta}\frac{e}{e-1}\left(\log\frac{1}{\gamma} + d(d+1)/2 + d\right)\right]$
for $k \in \{0, \ldots, T\}$ do
    for $i \in \{0, \ldots, \Gamma\}$ do
        Simulator Initialization
        Record $z^k_i$ = $\mathcal{M}\left(\mathrm{y}^1_{0:k}\right)$
    end for
    Solve the convex problem
      $\begin{array}{ll} \arg\min_{A^k, b^k} & -\log\det A^k \\ \text{subject to} & \left\|A^k z^k_i - b^k\right\|_2 - 1 \leq 0,\, i = 0, \ldots, \Gamma \end{array}$
    return $A^k, b^k$
end for
```

4.1.2 WorstEventSelector

We now discuss how to select E^* that is most likely to show a violation of ε-δ differential privacy. The procedures are described in Alg. 4. They are designed to return one event E^* for the future use in Hypothesis Test in Alg. 1. First, **WorstEventSelector** receives an EventList from **EventListGenerator**. Then, for each event, it counts the number of estimation results falling in it and run **PVALUE** to compute p^*. The event that produces the minimal p^* is returned to Alg. 1 as E^* and **HypothesisTest** on E^* will be implemented to check ε-δ differential privacy.

4.1.3 HypothesisTest

We now discuss how to apply the Fisher's exact test to evaluate the differential privacy of our estimator $\mathcal{M}$. To do this, fix a event E^* and define $p_1 = \mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^1_{0:T}\right) \in E^*\right)$ and $p_2 = \mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^2_{0:T}\right) \in E^*\right)$. The random variables given by the history of sensor outputs $\mathrm{y}^i_{0:T} \sim Y^i$,

Algorithm 4 Worst Event Selector

```
function WORSTEVENTSELECTOR($n$, $\mathcal{M}$, $\varepsilon$, $y^1_{0:T}, y^2_{0:T}$, EventList)
    Input: Target Estimator($\mathcal{M}$)
            Desired differential privacy($\varepsilon$)
            $\delta$-adjacent measurement data($y^1_{0:T}, y^2_{0:T}$)
            EventList
    $O_1 \leftarrow$ Estimation results of running $\mathcal{M}(y^1_{0:T})$ for
            $n$ times
    $O_2 \leftarrow$ Estimation results of running $\mathcal{M}(y^2_{0:T})$ for
            $n$ times
    $pvalues \leftarrow [\,]$
    for $E \in$ EventList do
        $c_1 \leftarrow |\{i|O_1[i] \in E\}|$
        $c_2 \leftarrow |\{i|O_2[i] \in E\}|$
        $p^+, p_+ \leftarrow$ PVALUE $(c_1, c_2, n, \varepsilon)$
        $p^* \leftarrow \min(p^+, p_+)$
        $pvalues$.append($p^*$)
    end for
    WorstEvent ($E^*$) $\leftarrow$ EventList[argmin$\{pvalues\}$]
    Return WorstEvent ($E^*$)
end function
```

$i = 1,2$, starting from the initial condition x_0, defines a new random variable $Z^i = 1$ if $\mathcal{M}(Y^i) \in E^*$, $Z^i = 0$, otherwise. It is clear that Z^i are distributed as a Bernoulli with parameter p_i, $i = 1,2$. By implementing the estimator $\mathcal{M}$ using $y^1_{0:T}$ and $y^2_{0:T}$ for n_1 and n_2 times, we count the number of $\mathcal{M}\left(y^1_{0:T}\right) \in E^*$ as c_1 and $\mathcal{M}\left(y^2_{0:T}\right) \in E^*$ as c_2. In this sense, c_1 and c_2 can be seen as samples from two binomial distributions $\mathrm{B}(n_1, p_1)$ and $\mathrm{B}(n_2, p_2)$, which can be used in the Fisher's exact test.

Instead of evaluating $p_1 \leq p_2$, we are interested in testing the null hypothesis is $p_1 \leq \mathrm{e}^{\varepsilon} p_2$, with the additional e^{ε}. To handle the effect of e^{ε}, we adapt the following procedure. Let's consider sampling $\bar{c_1}$ from a $\mathrm{B}(c_1, 1/e^{\varepsilon})$ distribution. The sampling criteria satisfies the following Lemma:

Lemma 1 ([14].). *Let $Y \sim B(n, p_1)$, and Z be sampled from $B(Y, 1/e^{\varepsilon})$, the unconditional distribution of Z is $B(n, p_1/e^{\varepsilon})$.*

Proof. Suppose Z is sampled from a Binomial $(Y, 1/e^{\varepsilon})$ and Y is distributed according to a Binomial (n, p_1), which has two possible outputs:$Y_i = 0$ or $Y_i = 1$. Hence, the unconditional distribution of Z follows:

$$\begin{aligned} P(Z_i = 1) &= P(Z_i = 1|Y_i = 1)P(Y_i = 1) + P(Z_i = 1|Y_i = 0)P(Y_i = 0) \\ &= (1/e^{\varepsilon}) \cdot p_1 + 0 \cdot (1 - p_1) = p1/e^{\varepsilon} \end{aligned}$$

This implies that the unconditional distribution of Z subjects to the $\mathrm{B}(n, p1/e^{\varepsilon})$. □

Thus, the facts follow immediately from the lemma:

- if $p_1 > e^{\varepsilon} p_2$, then the distribution of $\bar{c_1}$ ($\mathrm{B}(n_1, \bar{p_1})$) has a larger Binomial parameter than c_2 ($\mathrm{B}n_2, p_2)$) with $\bar{p_1} = p_1/e^{\varepsilon} > p_2$. In this case, the test results are expected to be able to reject the null hypothesis.

- if $p_1 = e^{\varepsilon} p_2$, then the distribution of $\bar{c_1}$ ($\mathrm{B}(n_1, \bar{p_1})$) has the same Binomial parameter as c_2 ($\mathrm{B}(n_2, p_2)$) with $\bar{p_1} = p_1/e^{\varepsilon} = p_2$. In this case, the rest results are not expected to be able to reject the null hypothesis.

- if $p_1 < e^{\varepsilon} p_2$, then the distribution of $\bar{c_1}$ ($\mathrm{B}(n_1, \bar{p_1})$) has a smaller Binomial parameter than c_2 ($\mathrm{B}(n_2, p_2)$) with $\bar{p_1} = p_1/e^{\varepsilon} < p_2$. In this case, the rest results are not expected to be able to reject the null hypothesis.

Hence, the problem of testing the null hypothesis $H_0 : p_1 \leq e^{\varepsilon} p_2$ is reduced to the problem of testing $\bar{p_1} \leq p_2$ on the basis of $\bar{c_1}$, c_2 instead of c_1, c_2. Checking whether or not $\bar{c_1}$, c_2 are generated from the same distribution can be done using the Fisher's exact test with p-value being equal to 1 - Hypergeometric.cdf($\bar{c_1} - 1 | n_1 + n_2, n_1, \bar{c_1} + c_2$).

The test procedures can be described as follows in our work: Suppose we have an estimator $\mathcal{M}$, δ-adjacent sensor outputs $(\mathrm{y}^1_{0:T}, \mathrm{y}^2_{0:T})$ and an Event E^*, we wish to check if $\mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^1_{0:T}\right) \in E^*\right) \leq \mathrm{e}^{\varepsilon} \mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^2_{0:T}\right) \in E^*\right)$ or $\mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^2_{0:T}\right) \in E^*\right) \leq \mathrm{e}^{\varepsilon} \mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^1_{0:T}\right) \in E^*\right)$. The first case is addressed, as the other one is symmetric. The complete procedures are shown in Alg. 5 and we explain why it works before.

For this purpose, we do:

Algorithm 5 Hypothesis Test

```
function PVALUE($c_1, c_2, n, \varepsilon$)
    $\bar{c_1} \leftarrow \mathrm{B}(c_1, 1/e^{\varepsilon})$
    $s \leftarrow \bar{c_1} + c_2$
    $p^+ \leftarrow 1$ - Hypergeom.cdf($\bar{c_1} - 1|2n, n, s$)
    $\bar{c_2} \leftarrow \mathrm{B}(c_2, 1/e^{\varepsilon})$
    $s \leftarrow \bar{c_2} + c_1$
    $p_+ \leftarrow 1$ - Hypergeom.cdf($\bar{c_2} - 1|2n, n, s$)
    return $p^+, p_+$
end function
function HYPOTHESISTEST($n, \mathcal{M}$, $\varepsilon$, $\mathrm{y}^1_{0:T}, \mathrm{y}^2_{0:T}$, $E^*$)
    Input: Target Estimator($\mathcal{M}$)
           Desired differential privacy($\varepsilon$)
           $\delta$-adjacent sensor outputs($\mathrm{y}^1_{0:T}, \mathrm{y}^2_{0:T}$)
           $E^*$ (WorstEvent)
    $O_1 \leftarrow$ Estimation results of running $\mathcal{M}(\mathrm{y}^1_{0:T})$ for
           $n$ times
    $O_2 \leftarrow$ Estimation results of running $\mathcal{M}(\mathrm{y}^2_{0:T})$ for
           $n$ times
    $c_1 \leftarrow |\{i|O_1[i] \in E^*\}|$
    $c_2 \leftarrow |\{i|O_2[i] \in E^*\}|$
    $p^+, p_+ \leftarrow$ PVALUE ($c_1, c_2, n, \varepsilon$)
    Return $p^+, p_+$
end function
```

- Define $p_1 = \mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^1_{0:T}\right) \in E^*\right)$ and $p_2 = \mathbb{P}(\mathcal{M}\ \left(\mathrm{y}^2_{0:T}\right) \in E^*)$

- Create the null hypothesis as $H_0 : p_1 \leq e^{\varepsilon} \cdot p_2$

- Run $\mathcal{M}$ with δ-adjacent sensor outputs($\mathrm{y}^1_{0:T}, \mathrm{y}^2_{0:T}$) n times independently and results are recorded as O_1, O_2

- Count the number of times that the estimation results fall into E^*. Let $c_1 = |\{i|O_1[i] \in E^*\}|$ and $c_2 = |\{i|O_2[i] \in E^*\}|$. We note that c_1 and c_2 are equivalent to samples from $\mathrm{B}(n, p_1)$ and $\mathrm{B}(n, p_2)$, respectively. By intuition, $c_1 \gg e^{\varepsilon} c_2$ provides firm evidence against the null hypothesis

- Compute the p-value based on c_1, c_2 to quantify how unlikely the null hypothesis is

To summarize, given c_1 and c_2, $\bar{c_1}$ is first generated from the $\mathrm{B}(c_1, 1/e^{\varepsilon})$ distribution and then p-value is computed. We should note that generating $\bar{c_1}$ is a random process, so we reduce its variation by multiple sampling and averaging the p-values. In other words, we can run the function **PVALUE** multiple times with the same inputs and average the p-values. The main limitation is that a great amount of time is required for collecting the estimates. Only by this, we can expect the reasonable results from the statistical tests. In this paper, T (time horizon) is set as a small number for less computation.

4.2 Algorithm analysis

In this section, we will consider some theoretical results of the test framework in this paper. However, to do so, we will first restrict our notion of differential privacy as follows:

Definition 5 (Differential privacy wrt a space partition). *Let $\mathcal{P} = \{E_1, \dots, E_n\}$ be a space partition[1] and $\mathcal{M}$ be an estimator of System 3.1. We say that $\mathcal{M}$ is ε-δ differential private (up to time T) wrt $\mathcal{P}$ if the definition of ε-δ differential privacy holds for $\mathcal{M}$ with respect to each event E_k, $k = 1, \dots, n$, in the partition.*

The following results explain the relationship between differential privacy wrt partitions of different resolutions.

Lemma 2. *In the context of the previous definition, suppose we have a partition $\mathcal{P}_1 = \{E_1, \dots, E_n\}$, which is finer than another partition $\mathcal{P}_2 = \{F_1, \dots, F_n\}$. That is, each F_i can be represented by the disjoint union $F_i = \cup_{s=1}^{m_i} E_{l_s}$. Then, if ε-δ differental privacy holds on $\mathcal{P}_1$, then it also holds on $\mathcal{P}_2$.*

Proof. Suppose ε-δ differential privacy holds with respect to each event E_i in $\mathcal{P}_1$, then for all $i = 1, \dots, n$, $\mathbb{P}\left(\mathcal{M}\left(\mathrm{y}_{0:T}^1\right) \in E_i\right) \leq \mathrm{e}^{\varepsilon}\mathbb{P}\left(\mathcal{M}\left(\mathrm{y}_{0:T}^2\right) \in E_i\right)$ and $\mathbb{P}\left(\mathcal{M}\left(\mathrm{y}_{0:T}^2\right) \in E_i\right) \leq \mathrm{e}^{\varepsilon}\mathbb{P}\left(\mathcal{M}\left(\mathrm{y}_{0:T}^1\right) \in E_i\right)$ for $\mathrm{y}_{0:T}^1, \mathrm{y}_{0:T}^2$. In the following, we address the first inequality since their treatments are analogous.

[1] A partition of the space $\mathbb{R}^{(T+1)d_X}$ for Eqn. 3.1, if estimation is considered up to time T.

Take $F_i = E_{l_1} \cup \ldots \cup E_{l_{m_i}}$, then we can obtain:

$$\begin{aligned}
\mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^1_{0:T}\right) \in F_i\right) &= \mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^1_{0:T}\right) \in E_{l_1} \cup \ldots \cup E_{l_{m_i}}\right) \\
&= \sum_{s=1}^{m_i} \mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^1_{0:T}\right) \in E_{l_s}\right) \\
&\leq e^{\varepsilon} \sum_{s=1}^{m_i} \mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^2_{0:T}\right) \in E_{l_s}\right) \\
&= e^{\varepsilon} \mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^2_{0:T}\right) \in F_i\right)
\end{aligned}$$

This holds true since there is no overlap between each E_{l_s}. Thus, we can conclude that if ε-δ differential privacy holds for $\mathcal{P}_1$, it also holds for $\mathcal{P}_2$. □

Corollary 1. *In the context of this section, an estimator $\mathcal{M}$ is ε-δ differentially private (up to time T) if and only if it is ε-δ differentially private for infinitesimally small partitions.*

From the above, it is easy to see that if a mechanism is ε-δ differentially private wrt to all the points in the state-space, then it will be differentially private. Since it is impossible to handle infinitesimally small event sets, it is of interest to find a level of resolution in a partition that is enough to guarantee that ε-δ differential privacy wrt this partition. However, this does not seem to be possible. Differential privacy depends on the relative mass that the mechanism assigns to an event under two adjacent sensor outputs. This fraction is independent (and can be arbitrarily much larger than) the volume of the event set itself. Thus, even if a mechanism is differentially private wrt a partition composed of sets with very small volume, differential privacy wrt to a smaller event may not hold. Still, if the Jacobian of $\mathcal{M}$ is full row rank, one can view differential-privacy wrt a partition whose elements have a volume of ε to be a type of differential-privacy in "high-probability" $\geq 1-\zeta$, for some ζ a function of ε.

Now we turn our attention to the conditions that guarantee differential privacy wrt a partition with high confidence.

Recall that the rule to decide whether to accept the null hypothesis ($H_0 : p_1 \leq e^{\varepsilon} \cdot p_2$) based on the sample results on the worst event in a partition is described in Section 4.1. We now talk about the probability guarantee of the test.

Lemma 3. *Let α be a significance level, $\mathcal{M}$ be an estimator of System 3.1, and $\mathcal{P} = \{E_1, \dots, E_n\}$ be a partition that is used to test for differential privacy. Suppose that E^* is the worst event in the partition, in the sense that it provides with the minimum p-value. If this p-value is $p \leq \alpha$, the probability of making a Type I error to verify differential privacy wrt E^* is exactly less or equal than α. Thus, ε-δ differential privacy of $\mathcal{M}$ holds wrt to the partition $\mathcal{P}$ with probability $(1-\alpha)$.*

Proof. By looking at the worst-case event as indicated above, we guarantee that, if the test is passed, then it will also pass wrt all other elements in the partition. Given Lemma 1 and the exactness provided by the Fisher's test provided in Chapter 3.4, we can say that, for each event, the probability of making a Type I error is thus bounded by α exactly and it is also independent of the number of trials taken. Collectively, this means that differential privacy (or the probability of making a Type I error for all events) holds with probability $(1-\alpha)$. □

Finally, we can put a theorem together on the correctness of our algorithm to verify differential privacy using hypothesis test and reachable set approximation.

Theorem 2. *Consider an estimator $\mathcal{M}$ of the System 3.1, a time horizon T, and parameters $\varepsilon, \delta, \beta$ and γ. Consider a partition $\mathcal{P}$ of the subset of $R \subseteq \mathbb{R}^{(T+1)d_X}$, given by $R = \hat{R}_{[t_0,t_1]} \times \hat{R}_{[t_0,t_2]} \times \cdots \times \hat{R}_{[t_0,T]}$. Then, if Γ is taken such that*

$$\Gamma \geq \frac{1}{\beta}(\frac{e}{e-1})(\log\frac{1}{\gamma} + \frac{d(d+1)}{2} + d),$$

and the estimator passes the test given by Alg. 1, then, $\mathcal{M}$ is ε-δ differentially private wrt $\mathcal{P} \cup \{x \,|\, x \in \mathbb{R}^{(T+1)d_X} \setminus R\}$ with probability $(1-\alpha)(1-\gamma)$.

Chapter 5

Experiments & Results

In this chapter, we present the simulation results using the framework proposed in this paper. The simulations are performed in MATLAB(R2020a).

5.1 Example description

5.1.1 Dynamical system.

In what follows, we consider a non-isotropic oscillator in $\mathbb{R}^2$ with potential function:

$$V\left(x^1,x^2\right)=\frac{1}{2}\left((x^1)^2+4(x^2)^2\right).$$

Thus, the corresponding oscillator particle with position $\mathbf{x}_k=(x_k^1,x_k^2)\in\mathbb{R}^2$ moves from initial conditions $x_0^1=5, x_0^2=0, \dot{x}_0^1=0, \dot{x}_0^2=2.5$ under the force $-\Delta V$. The update equations of our autonomous dynamic system become:

$$\mathbf{x}_{k+1}=f(\mathbf{x}_k)=A\mathbf{x}_k, \quad k\geq 0.$$

where, A is a constant matrix. Thus, this takes the form of System 3.2, and Condition 1 (**Lipschitz continuity**), it holds with a Lipschitz constant for the dynamic given by $c_f = \|A\|$ as $|f(\mathbf{x}_1) - f(\mathbf{x}_2)| \leq \|A\| \|\mathbf{x}_1 - \mathbf{x}_2\|$. It should be noted that at each time step k, the system dynamical noise accords with a uniform distribution between $[-0.001, 0.001]$ for each dimension. We will assume that the distribution of initial conditions is given by a truncated Gaussian distribution with mean vector (5, 0, 0, 2.5) and bounded by 0.1 measured in the L_2 distance, which means $\mathcal{K}_0$ in Eqn. 3.8 is equal to 0.1.

5.1.2 Sensor network and observation model.

We will assume that a sensor network consisting of p nodes is placed on a circle with the center point located at (0,0) and radius $R = 10\sqrt{2}$, surrounding the oscillator. In addition, suppose that the sensors are homogeneous with the following observation model:

$$\begin{aligned} \mathbf{y}_{i,k} &= h(\mathbf{x}_k, \mathbf{q}_i) + \mathbf{v}_{i,k} \\ &= 100 \tanh\left(0.1\left(\mathbf{x}_k - \mathbf{q}_i\right)\right) + \mathbf{v}_{i,k}, \quad i = 1, \ldots, p, \end{aligned}$$

where $\mathbf{q}_i \in \mathbb{R}^2$ is the position of sensor i on the circle, and $\mathbf{x}_k = \left(x_k^1, x_k^2\right)$ is the position of the particle at time step k. Here, the hyperbolic tangent function tanh is applied in a element-wise way. Velocities are not observed by the sensors. The vector $\mathbf{v}_{i,k} \in \mathbb{R}^2$ represents the observation noise of each sensor, which is generated from the same truncated mixed Gaussian distribution at each time step k in simulation, indicating that the volume of random noise has a limit. tanh function introduces nonlinearity into the observation model. All these observations are stacked together as sensor outputs $\mathbf{y}_k = (\mathbf{y}_{1,k}^\top, \ldots, \mathbf{y}_{p,k}^\top)^\top \in \mathbb{R}^{2p}$ in W_2-MHE.

Given a fixed sensor position, $\mathbf{q}_i$, the (stacked noiseless) measurement function of the collective sensor network is:

$$h(\mathbf{x}_k) = \left(h\left(\mathbf{x}_k, \mathbf{q}_i\right)^\top = 100 \tanh\left(0.1\left(\mathbf{x}_k - \mathbf{q}_i\right)\right)^\top\right)^\top \in \mathbb{R}^{2p}.$$

Since tanh function has a Lipschitz constant of 1, We have that

$$\|h(\mathbf{x}_1) - h(\mathbf{x}_2)\| \leq \sum_{i=1}^{p} \|h\left(\mathbf{x}_1, \mathbf{q}_i\right) - h\left(\mathbf{x}_2, \mathbf{q}_i\right)\| \leq 10 \sum_{i=1}^{p} \|\mathbf{x}_1 - \mathbf{x}_2\| = 10p\|\mathbf{x}_1 - \mathbf{x}_2\| \tag{5.1}$$

. Thus, in the Lipschitz continuity condition 1. in System 3.1, we have $c_h = 10p$.

5.1.3 Estimation horizon and other filter parameters.

In order to implement the W_2 filter, we consider a certain time horizon $T = 8$ and a moving horizon $N = 5 \leq T$. We have:

$$\begin{aligned} G_k^N(\mathbf{x}_k) = & \; J_N\left(\mathbf{y}_{k:k+N}, \Sigma_N(\mathbf{x}_k)\right), \\ = & \; \textstyle\sum_{i=0}^{N} \|\mathbf{y}_{k+i} - h \circ f^i(\mathbf{x}_k)\|^2. \end{aligned}$$

Taking gradients, we have $\|\nabla G_k^N(\mathbf{x}_1) - \nabla G_k^N(\mathbf{x}_2)\| \leq 2(N+1)c_h \max_l \|A\|^l \|\mathbf{x}_1 - \mathbf{x}_2\|$. So l in Eqn. 3.8 will be approximated as $l = 2(N+1)10p\|A\|$.

5.1.4 δ-adjacent Sensor Outputs.

To check for differential privacy, we will consider adjacent sensor outputs given by adjacent sensors. To generate these, we first obtain a set of p sensor positions, denoted by S^1. Then, with $p - m$ $(0 < m < p)$ sensors fixed, we replace m of them by adjacent m sensors on the same circle and obtain a new set of p sensors, denoted by S^2. Let us use θ^1 and $\theta^2 \in \mathbb{R}^m$ to represents the angle of m sensors on the circle before and after replacement, respectively. For the two sets of sensor outputs $\mathbf{y_k^1}, \mathbf{y_k^2}$ obtained using S^1 and S^2, together with the fact that tanh function in obervation modelhas a Lipschitz constant of 1, at time step k, we have:

$$\begin{aligned}
\left|\mathbf{y}_k^1 - \mathbf{y}_k^2\right| = & \left|h(\mathbf{x}, S^1) - h(\mathbf{x}, S^2)\right|, \\
\leq & 10m\|(\cos\theta^1, \sin\theta^1) - (\cos\theta^2, \sin\theta^2)\|, \\
\leq & 10m\sqrt{2}\|\theta^1 - \theta^2\|.
\end{aligned}$$

where, $\Delta\theta = \|\theta^1 - \theta^2\|$, and it be measured as the L_2 norm between the two vectors θ^1 and θ^2. The distance between two sensor outputs along a full time horizon T is referred as $d_{\mathrm{y}}\left(\mathbf{y}_{0:T}^1, \mathbf{y}_{0:T}^2\right)$ in Def. 3, which is less than:

$$d_{\mathrm{y}}\left(\mathbf{y}_{0:T}^1, \mathbf{y}_{0:T}^2\right) \leq 10m\sqrt{2(T-N+1)}\Delta\theta = 20\sqrt{2}m\Delta\theta.$$

Therefore, in order to generate δ adjacent measurement data, we take

$$\Delta\theta \leq \frac{\delta}{20\sqrt{2}m}. \tag{5.2}$$

In our simulations, given δ, we take $m = 1$, so that one of p sensors is moved to a new position that is $\frac{\delta}{20\sqrt{2}}$ close on the circle.

For each time step k, s_k is assigned with same value and we want to verify the correctness of Eqn. 3.9.

5.2 Numerical Verification Results of W_2-MHE

In this section, we present our simulation & verification results. As described in Alg. 3, we take $d = 2$ and for the guarantee, we take $\beta = 0.05, \gamma = 10^{-9}$. The number of samples we take is $\Gamma = 814$ and each of them are initially generated from the truncated Gaussian distribution specified before. This allows us to produce an ellipsoid that contains at least 0.95 of the actual reachable set distribution with probability $\geq 1 - 10^{-9}$ (see Theorem 1). An example of the estimated reachable set at $t = 1$ is shown below.

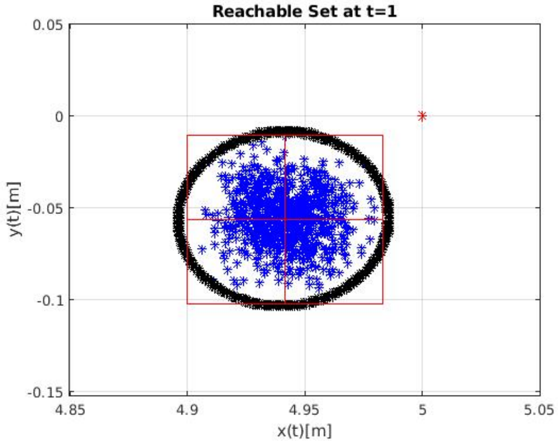


Figure 5.1: Estimated reachable set at $t = 1$

In Fig. 5.1, the red point represents the ground-truth position, that is, the oscillator's state evolving under the noiseless dynamics as described in Subsection 5.1.1 and Γ blue points represent the estimated positions using differentially private W_2-MHE. It can be seen how accuracy is sacrificed for the purpose of privacy. The black points form the ellipsoid that demonstrates the reachable set of the system subject to noise. The red lines divide the set into a gridded region with 4 subsets, plus a complementary set, which define the partition wrt we check for differential privacy as in Section 4.1.1.

The goal is to explore the relation between the accuracy and the privacy ε. By doing so, the system designers are able to verify the minimum requirements so that the proposed estimators can achieve ε level of differential privacy but also obtain the highest accuracy. At each time step k, (4 in total since $T = 8, N = 5$), the same operation is applied. Thus, the **EventList** is obtained by recording all the possible combinations of this partition, with the length of $4^4 = 256$. Followed by this, we re-run W_2-MHE enough times using both sets of sensor outputs, respectively. During the computation, as shown in Alg. 4, c_1, c_2 of each event are recorded and finally the event with minimum p-value is returned as **WorstEvent**. Again, we record c_1, c_2 with respect to this event by another set of runs. Then, the p-value is computed under a different level of differential privacy ε. We then compare this with the significance parameter α, we can tell whether or not ε-δ differential privacy is satisfied.

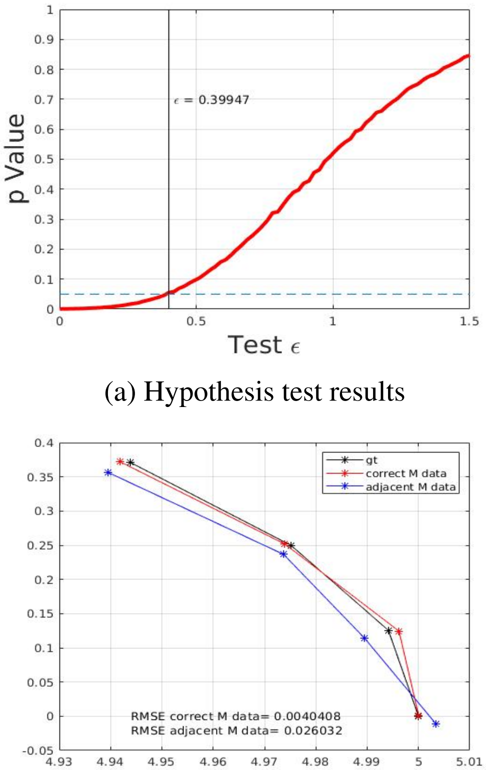
We first set $s_k = 0.8$ in Eqn. 3.7 and $\delta = 10$ in Eqn. 5.2 (for all simulations in the thesis) and the simulation results are shown in Fig. 5.2. In the top figure, we can see that around $\varepsilon = 0.39947$, the p-value grows larger than 0.05. From the definition of differential privacy, this means that $(\mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^1_{0:T}\right) \in E^*\right) \leq \mathrm{e}^{\varepsilon}\mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^2_{0:T}\right) \in E^*\right)$ or $\mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^2_{0:T}\right) \in E^*\right) \leq \mathrm{e}^{\varepsilon}\mathbb{P}\left(\mathcal{M}\left(\mathrm{y}^1_{0:T}\right) \in E^*\right))$ holds, if $\varepsilon \geq 0.39947$ in the worst case event E^*. This ε provides a lower bound to ε-δ differential privacy with high confidence. The bottom figure shows a plot of the ground-truth states as well as the estimates using both sets of sensor outputs. The root mean squared error (RMSE) for the estimated are found to be $E_{\text{correct}} = 0.0040408$ and $E_{\text{adjacent}} = 0.026032$ for the estimates using correct and adjacent sensor outputs, respectively. Recall that $s_k = 0.8$ implies a relatively low noise injection level. Note that here, E_{adjacent} is larger than E_{correct} because the adjacent sensor outputs are in fact generated from another set of sensor positions.

Then, we decrease s_k to be 0.7, enlarging the entropy term in the objective function. As shown in Fig. 5.3, the critical ε is smaller ($\varepsilon = 0.11485$), confirming that a higher level of differential privacy is satisfied wrt the considered partition and at a high confidence. However, the increase of privacy leads to a decrease in accuracy, which can be seen from $E_{\text{correct}} = 0.00599996$.

Therefore, the tests reflect the expected trade-off between differential privacy and accuracy. In order to choose between two given estimation methods, a designer can either (i) first set a bound on what is the tolerable estimation error, then compare two methods based on the differential privacy level they guarantee based on the given test, or (ii) given a desired level of differential privacy, choose the estimation method that results into the smallest estimation error.

5.2.1 Correctness of the sufficient Condition

Eqn. 3.9 provides with a theoretical formula to calculate ε that guarantees ε-δ differential privacy. However, since this condition is derived using several assumptions and upper bounds,



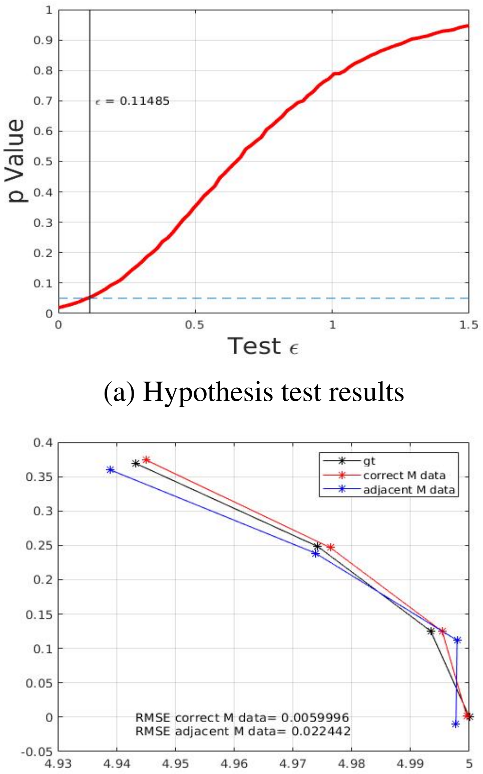
(a) Hypothesis test results

(b) Estimation accuracy

Figure 5.2: W_2-MHE: State estimation & differential privacy test results ($s_k = 0.8$)

the answer is in general expected to be conservative.

In order to make comparisons, we take $s_k = 0.8$ and the value of other parameters are: $T = 8, c_f = 1.0777, c_h = 100, l = 1293.2 (N = 5), \text{diam}(\mathcal{K}_0) = 0.1, \delta = 10$. Plug these values into Eqn. 3.9, we can obtain $\varepsilon \geq 6977.2$. Upon inspection, it is clear that the theoretical answer is much more conservative than the approximated one based on Fig. 5.2a, which indicates that if $\varepsilon \geq 0.39947$, the differential privacy is satisfied with high confidence wrt the given space partition.



(a) Hypothesis test results

(b) Estimation accuracy

Figure 5.3: W_2-MHE: State estimation & differential privacy test results ($s_k = 0.7$)

5.3 Input Perturbation

In [9] two mechanisms were defined that lead to two different approaches to produce differentially-private estimators. The first one generally adds noise at the output of the perfect estimators (and W_2-MHE discussed above can be considered in this category of estimators). The second one directly perturbs each input signal, such as the sensor output measurements. These perturbations are then passed through the estimators, leading to noisy outputs. Since only the input signals are perturbed, no changes of the estimators are necessary. An advantage of this approach is that the users do not need to rely on a trusted server to maintain their privacy since they themselves can release noisy signals.

Taking benefit of our work, it is then of interest to numerically compare the performance of these two mechanisms in the W_2-MHE estimator method. By selecting $s_k = 1$ (remove the entropy term in Eqn. 3.7) and adding Gaussian noise directly to both sets of sensor outputs, we can run the estimator enough times and find the trade-off between accuracy and privacy. The Gaussian noise has zero mean and the covariance matrix Q is defined as:

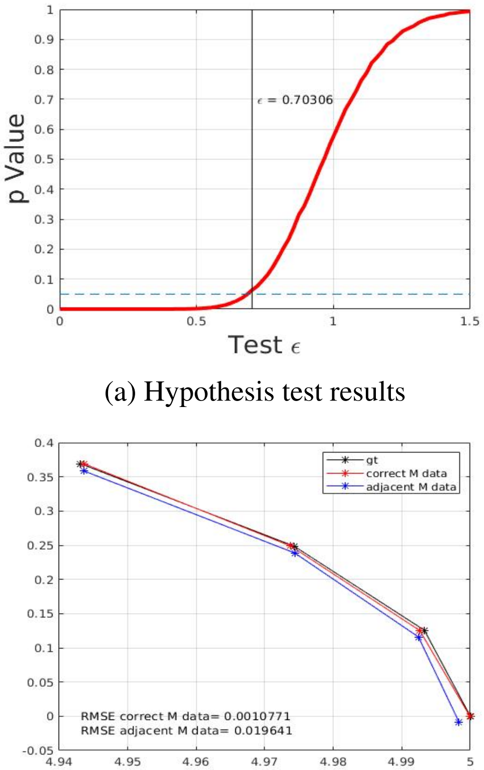
$$Q = (1-s)(I + \frac{R+R'}{2}). \tag{5.3}$$

Where, I is an Identity matrix and R is a matrix of random numbers over the interval $(0,1)$ (agree with uniform distribution). The value s is a scalar, $s \in \mathbb{R}$ which decides the magnitude of covariance matrix. In simulation, we change the value of s and compare the results.

We first set $s = 0.944$ in Eqn. 5.3 and the simulation results are shown in Fig. 5.4. Then, we increase the magnitude of the noise by setting $s = 0.894$ and the simulation results are shown in Fig. 5.5. They also show that the higher level of differential privacy is achieved at the loss of accuracy. From Fig. 5.2 and Fig. 5.5, we find that although the level of differential privacy is close to each other, the RMSE in Fig. 5.5 is only $1/3$ of that in Fig. 5.2. Thus, the second mechanism (adding noise directly at the mechanism input) seems to indicate that can lead to better accuracy while maintaining the same ε-δ differential privacy guarantee. However, this is a preliminary result applied on a particular example, which requires further investigation to confirm the results.

5.3.1 Differentially private EKF

The framework can be easily extended with other differentially private estimators. Here, instead of W_2-MHE, a modified extended Kalman filter is tested on the same experiment settings.



(a) Hypothesis test results

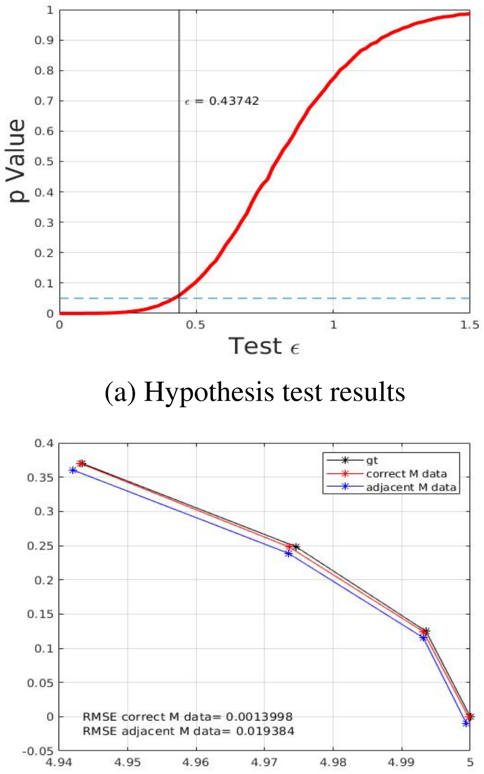
(b) Estimation accuracy

Figure 5.4: Input Perturbation: State estimation & differential privacy test results ($s = 0.944$)

The prediction and update formula are described in Eqn. 5.4.

$$\begin{aligned}
\bar{\mu}_k &= A\mu_{k-1}, \\
\bar{\Sigma}_k &= A\Sigma_{k-1}A^T + R_k, \\
M_k &= \bar{\Sigma}_k H_k^T \left(H_k\bar{\Sigma}_k H_k^T + Q_k\right)^{-1}, \\
\mu_k &= \bar{\mu}_k + M_k\left(z_k - h\left(\bar{\mu}_k\right)\right) - \tfrac{1-s_k}{s_k}w, \\
\Sigma_k &= \left(I - K_k H_k\right)\bar{\Sigma}_k.
\end{aligned} \tag{5.4}$$

In Eqn. 5.4, R_k, Q_k represent the covariance matrix of the update and measurement noise, respectively; H_k represents gradient matrix of the nonlinear measurement function $h(\mathbf{x})$ at $x = \bar{\mu}_k$ Compared with the regular EKF, uniformly distributed random noise w is added to the filter



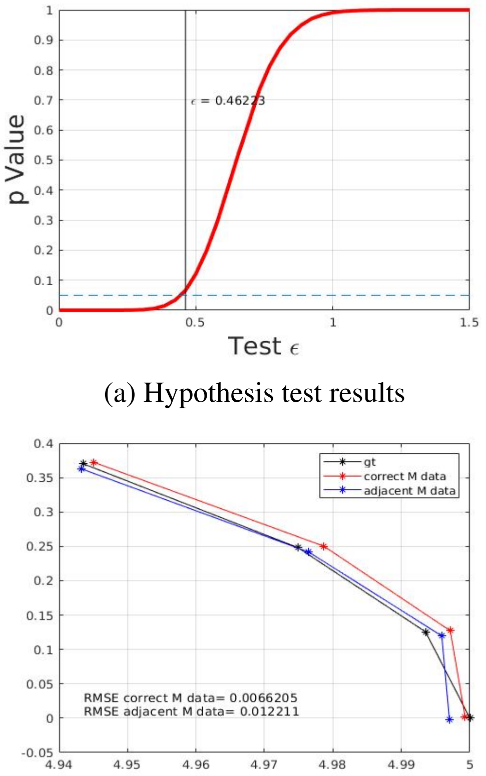
(a) Hypothesis test results

(b) Estimation accuracy

Figure 5.5: Input Perturbation: State estimation & differential privacy test results ($s = 0.894$)

output at the update step, which makes the estimator differentially private. The initial guess μ_0 is generated from the same distribution used in W_2-MHE.

In simulation, we set $s_k = 0.96$ and fine-tune the noise covariance matrices. The results are shown in Fig. 5.6. From that, the critical $\varepsilon = 0.46223$ with the RMSE $E_{\text{correct}} = 0.0066205$. Therefore, compared with the results in Fig. 5.2, the performance of the EKF is worse than that of W_2-MHE with respect to both privacy level and RMSE. Besides, for W_2-MHE, no extra efforts of tuning parameters (R_k and Q_k in Eqn. 5.4) or assumption of Gaussian noise distribution are required, which makes it more applicable in complex environments.



(a) Hypothesis test results

(b) Estimation accuracy

Figure 5.6: Diff-private EKF: State estimation & differential privacy test results ($s_k = 0.96$)

Chapter 6

Conclusions

In this work, we propose a novel numerical verification framework of differential privacy for estimators. We first introduce the differentially private mechanisms in the context of estimation by describing a moving horizon estimator that guarantees ε-differential privacy. We then clearly establish each components of the framework, including the approximation of reachable set by a data-driven method. In experiments, we demonstrate the estimator's performance and built the connections between the theoretical solution and the numerical solution. Also, we discuss on the other approaches and compare the simulation results. It is obvious that the framework can be applied to different estimators and this can benefit all researchers or designers working on this field.

This work is also being prepared for a publication: Yunhai Han; Sonia Martínez. The thesis author will be the primary author of the paper.

Bibliography

[1] A. Narayanan and V. Shmatikov. How to break anonymity of the netflix prize dataset. *preprint arxiv:cs/0610105*, 2006.

[2] B. Hoh, T. Iwuchukwu, Q. Jacobson, D. Work, A. M. Bayen, R. Herring, J. Herrera, M. Gruteser, M. Annavaram, and J. Ban. Enhancing privacy and accuracy in probe vehicle-based traffic monitoring via virtual trip lines. *IEEE Transactions on Mobile Computing*, pages 849–864, 2012.

[3] C. Dwork, M. Frank, N. Kobbi, and S. Adam. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, pages 265–284, 2006.

[4] C. Dwork. Differential privacy. In *Int. Colloquium on Automata, Languages, and Programming*, pages 1–12, 2006.

[5] S. Prasad and K. A. Smith. A note on differential privacy: Defining resistance to arbitrary side information, 2009.

[6] Y. Wang, Z. Huang, S. Mitra, and G. E. Dullerud. Entropy-minimizing mechanism for differential privacy of discrete-time linear feedback systems. In *IEEE Int. Conf. on Decision and Control*, pages 2130–2135, 2014.

[7] V. Katewa, A. Chakrabortty, and V. Gupta. Protecting privacy of topology in consensus networks. In *American Control Conference*, pages 2476–2481, 2015.

[8] J. L. Ny. Privacy-preserving filtering for event streams. *preprint arXiv: 1407.5553*, 2014.

[9] J. L. Ny and G. J. Pappas. Differentially private filtering. *IEEE Transactions on Automatic Control*, pages 341–354, 2014.

[10] V. Krishnan and S. Martínez. A probabilistic framework for moving-horizon estimation: Stability and privacy guarantees. *IEEE Transactions on Automatic Control*, pages 1–1, 06 2020.

[11] J. Cortes, G. E. Dullerud, S. Han, J. L. Ny, S. Mitra, and G. J. Pappas. Differential privacy in control and network systems. In *IEEE Int. Conf. on Decision and Control*, pages 4252–4272, 2016.

[12] Y. Chen and A. Machanavajjhala. On the privacy properties of variants on the sparse vector technique, 2015.

[13] M. Lyu, D. Su, and N. Li. Understanding the sparse vector technique for differential privacy, 2016.

[14] Z. Y. Ding, Y. X. Wang, G. H. Wang, D. F. Zhang, and D. Kifer. Detecting violations of differential privacy. *Proc.s of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018.

[15] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[16] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[17] L. Ji, J. Rawlings, W. Hu, A. Wynn, and M. Diehl. Robust stability of moving horizon estimation under bounded disturbances. *IEEE Transactions on Automatic Control*, 61(11):3509–3514, 2016.

[18] A. Angelo and B. Giorgio. Moving horizon estimation: Open problems, theoretical progress, and new application perspectives. *International Journal of Adaptive Control and Signal Processing*, pages 703–705, 2020.

[19] A. Alessandri, M. Baglietto, and G. Battistelli. Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes. *Automatica*, 44(7):1753–1765, 2008.

[20] I. Zang and M. Avriel. On functions whose local minima are global. *Journal of Optimization Theory & Applications*, 16(3-4):183–190, 1975.

[21] A. Alessandri and M. Gaggero. Fast moving horizon state estimation for discrete-time systems using single and multi iteration descent methods. *IEEE Transactions on Automatic Control*, 62(9):4499–4511, 2017.

[22] George Casella and Roger L. Berger. *Statistical Inference*. Cengage Learning, 2021.

[23] David J. Oliver. *Statistical Theory and Inference*. Springer, 2014.

[24] R. A. Fisher. *The design of experiments*. 1935.

[25] A. Devonport and M. Arcak. Estimating reachable sets with scenario optimization. In *Annual Learning for Dynamics & Control Conference*, 2020.

[26] Ron S. Dembo. Scenario optimization. *Ann. Oper. Res.*, 30(1):63–80, 1991.

[27] R. TempoFabrizio and D. Dabbene. Randomized algorithms for analysis and control of uncertain systems. *Springer*, 2007.