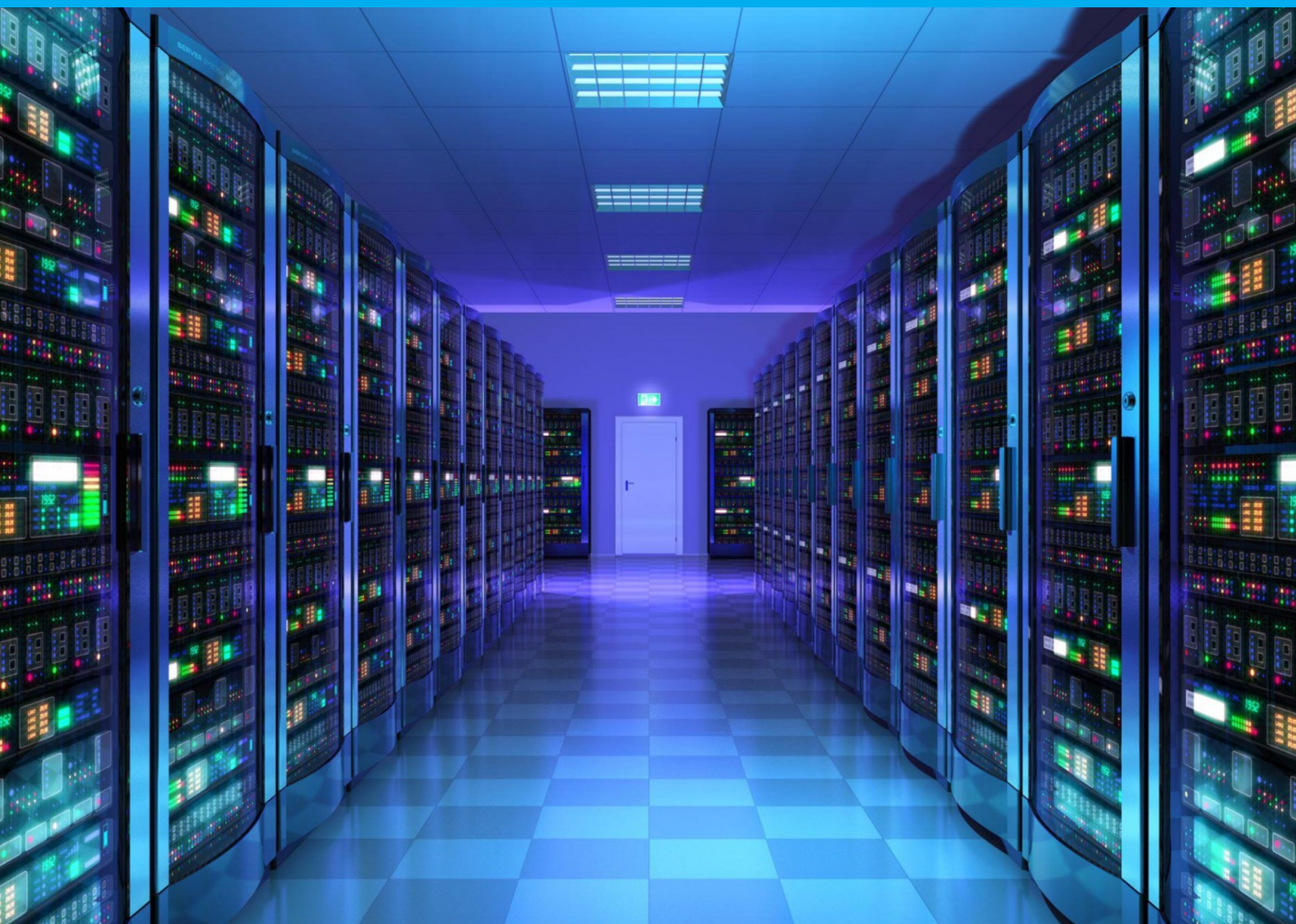
Defense Against Malicious Parameter Identification

System Immersion Coding and Hybrid Multiplicative Watermarking

Jiaxuan Zhang



Defense Against Malicious Parameter Identification

System Immersion Coding and Hybrid Multiplicative Watermarking

by

Jiaxuan Zhang

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday June 23, 2023 at 10:30 AM.

Student number:	5258162	
Project duration:	September 1, 2022 – June 23, 2023	
Thesis committee:	Associate Prof. Riccardo M. G. Ferrari,	TU Delft, supervisor
	Prof. Peter Palensky,	TU Delft
	Associate Prof. Peyman Mohajerin Esfahani	TU Delft
	Dr. Alexander J. Gallo,	TU Delft, daily supervisor

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Preface

The approximate-9-month thesis work is my first time tasting a research-oriented work. It has been a novelty experience for me. I would like to express my deepest gratitude to everyone for their invaluable support and contributions throughout the completion of this master's thesis.

I would like to thank my supervisor Prof. Riccardo Ferrari and my daily supervisor Dr Alex J. Gallo, for all the help and instruction. Their exceptional expertise, unwavering support, and invaluable guidance have been favourable in every stage of this research journey. They taught me how to think, solve, organize and write down research problems academically. Their patience, encouragement, and support sustained me, guiding me through each challenge and uplifting me when I encountered obstacles in my research. I am truly fortunate to have had the privilege of working under their mentorship. I am profoundly grateful to Prof. Riccardo Ferrari. His insightful feedback and ability to pose thought-provoking questions have significantly shaped the direction and quality of this thesis. Besides, Prof. Riccardo Ferrari can always provide great academic resources and opportunities, and these resources really help me to know more about academic research. I would also like to extend my heartfelt appreciation to Dr. Alex J. Gallo. I benefit a lot from the weekly meeting with Dr Alex J. Gallo. His extensive knowledge and expertise have been instrumental in providing additional perspectives and deepening the theoretical underpinnings of this study.

The 3-year master life in TU Delft is coming to a close. I have gained a lot from the experience. I have learned how to handle problems calmly, approach life positively, and accept myself with tranquillity. Special Thanks to my family (father Jiantang Zhang, mother Fengyun Liu, sister Yiying Zhang)for their unwavering love, and encouragement. Their unyielding support, sacrifices, and understanding have provided me with strength and motivation. I am truly fortunate to have them as my pillars of support. I would also like to express my gratitude to all my friends(Yiting Li, Li Xu, Ketong Huang, Zelin Xu, Xin Tan, Qingxin Liu, Hankai Yang, Chenxu Ma, Zhuoran Guo, Yuan Fu, Jingqi Zhuang, Chengkai Zhang, etc.). Their accompanying makes the journey meaningful, and their suggestions and shared experiences make me grow. Finally, I would like to thank my girlfriend, Yucong Zhang, for her love, unwavering support and understanding.

Jiaxuan Zhang
Delft, June 2023

Abstract

Cyber-physical systems are vulnerable to malicious attacks, which can lead to severe consequences. Active detection methods have emerged as a promising approach for identifying such attacks. However, existing active detection methods are susceptible to malicious parameter identification attacks, where attackers exploit eavesdropped data to identify and manipulate the active detection mechanisms. In this work, we propose two methods to address the issue of malicious parameter identification: the *system immersion coding* method and the *hybrid multiplicative watermarking* method. These approaches have a primal focus on disturbing the identification of attackers and defending against malicious parameter identification. Besides, as active detection methods, both of them are capable of detecting multiple attacks.

The system immersion coding method, derived from the privacy solution in federated learning, is adapted to enhance its capability to detect malicious attacks by merging the input information and defend malicious parameter identification by leveraging its privacy-preserving properties. This method involves mapping the plant output into a higher-dimensional space and introducing carefully defined noise, which can create arbitrarily large disturbances without compromising performance. The introduced disturbance disrupts the attacker's parameter estimation. Theoretical conditions are provided to discuss the detection performance of replay attacks, control-signal-injection zero-dynamics attacks, and sensor-signal-injection zero-dynamics attacks. However, we also identify that the system immersion coding method is vulnerable to *known-plaintext attacks*.

Watermarking is a promising active diagnosis technique for the detection of highly sophisticated attacks. Motivated by the computational hardness problems of cryptography analysis, we propose a hybrid multiplicative watermarking scheme as an active diagnosis technique. In this scheme, watermarking parameters are periodically updated based on the dynamics of unobservable states in specifically designed piecewise affine (PWA) hybrid systems. We conduct a theoretical analysis to assess the impact of this scheme on closed-loop performance, demonstrating its stability preservation. We also provide conditions to detect replay attacks and control-signal-injection zero-dynamics attacks. Furthermore, we demonstrate that the proposed approach makes it challenging for an eavesdropper to reconstruct watermarking parameters, considering both computational complexity and systems theoretic perspectives.

Notation

Throughout the thesis, the following notation is used. $\mathbb{Z}_+$ denotes the set of nonnegative integers. I_n represents the n-dimensional identity matrix, while $0_{n\times m} \in \mathbb{R}^{n\times m}$ is a matrix of zeros; whenever clear from context, the subscripts n and $n \times m$ are omitted. Given a matrix $X \in \mathbb{R}^{n\times n}$, $\sigma(X)$ denotes its spectrum, and $\rho(X)$ its spectral radius.A matrix $X \in \mathbb{R}^{n\times n}$ is said to be orthogonal, or orthonormal if it is invertible and $X^{-1} = X^\top$. The space of symmetric matrices in $\mathbb{R}^{n\times n}$ is defined as $\mathbb{S}^n \subset \mathbb{R}^{n\times n}$. For any two matrices X_1 and X_2, let $X = \mathrm{diag}(X_1, X_2)$ denote the block-diagonal matrix defined by X_1 and X_2. The space of symmetric matrices in $\mathbb{R}^{n\times n}$ is defined as $\mathbb{S}^n \subset \mathbb{R}^{n\times n}$. Notation $X \succ (\succeq) 0$ is used to state that a symmetric matrix $X \in \mathbb{S}^n$ is positive (semi)definite; similarly, a negative (semi)definite matrix is defined as $x \prec (\preceq) 0$.

Given a time-varying signal $x[k] \in \mathbb{R}^n$, $k \in \mathbb{Z}_+$, $x[k_1 : k_2]$ is the sequence of instances $x[k], k \in \{k_1, k_1 + 1, \ldots, k_2\} \subseteq \mathbb{Z}_+$. A polyhedron $\mathscr{X} \subset \mathbb{R}^{n\times n}$ is a convex set, defined as $\mathscr{X} = \{x \in \mathbb{R}^n : Hx \leq k\}$, where $H \in \mathbb{R}^{m\times n}$ and $k \in \mathbb{R}^m$. For any two sets $\mathscr{A}$ and $\mathscr{B}$, $\mathscr{A} \times \mathscr{B}$ denotes their Cartesian product.

Contents

1 Introduction 1
1.1 Cyber-Physical System(CPS) and Its Vulnerabilities . . . 1
1.2 Security Problem in CPSs . . . 1
1.3 Detection and Isolation Methods . . . 2
1.4 Research Questions and Contributions . . . 3
1.5 Outline . . . 4

2 Background: Cyber-Physical Systems(CPSs) 5
2.1 CPSs Model, A Control Perspective . . . 5
2.1.1 A Classical Linear Configuration . . . 5
2.2 Attacker Models . . . 6
2.2.1 Eavesdropping System Identification(ESI) Attack . . . 7
2.2.2 Denial-of-service Attack . . . 7
2.2.3 Replay Attack . . . 7
2.2.4 Zero-Dynamics Attack . . . 8
2.2.5 Covert Attack . . . 9
2.3 Passive Detection Methods . . . 9
2.3.1 Detector Design or Detector Optimization Methods . . . 10
2.3.2 Secure State Estimation Methods . . . 10
2.3.3 Machine-Learning-Based Methods . . . 10
2.4 Active Detection Methods . . . 11
2.4.1 Additive Watermarking (AW) . . . 12
2.4.2 Measurement-Coding Methods . . . 14
2.4.3 Moving Target Defense (MTD) Methods . . . 15
2.4.4 Multiplicative Watermarking (MW) . . . 18
2.4.5 Other Methods . . . 19
2.5 Privacy Problem in Networked Control Systems and Other Fields . . . 20
2.5.1 Privacy Solutions in Networked Control Systems . . . 21
2.5.2 Privacy Solution in Supply Chain and Federated Learning . . . 23
2.5.3 Conclusion for Privacy Problems . . . 24
2.6 Conclusion . . . 25

3 Problem Formulation 29
3.1 System Model . . . 29
3.2 Attacker Capabilities . . . 30
3.3 Active Detection Methods and Privacy Solutions Considered . . . 30
3.3.1 System Immersion Method in Federated Learning . . . 30
3.3.2 Measurement-Coding Method and Output-Coding Method . . . 31
3.3.3 Multiplicative Watermarking . . . 31
3.4 Problem Formulation . . . 32
3.4.1 Problem Formulation: System Immersion Coding Method . . . 32
3.4.2 Problem Formulation: Hybrid Multiplicative Watermarking Method . . . 32
3.5 Testbench Overview . . . 33
3.5.1 Testbench 1 . . . 33
3.5.2 Testbench 2 . . . 35

4 System Immersion Coding Method 39
4.1 System Immersion Coding for Active Detection Method . . . 39
4.1.1 Detectability Design of System Immersion Coding . . . 40

4.2 Identification Resistance of System Immersion Coding Method . . . 44
4.2.1 Identification of the System Immersion Parameter . . . 44
4.2.2 Problem of System Immersion Coding Method: Under the Known-plaintext Attack . . . 45
4.3 Simulation Study . . . 47
4.3.1 Detection Performance . . . 47
4.3.2 Theorem Verification . . . 48
4.3.3 Identification Resistance: Testbench 1 & Testbench 2 . . . 49
4.4 Conclusion . . . 51

5 Hybrid Multiplicative Watermarking: Theory 55
5.1 Background: Hybrid System Theory . . . 55
5.1.1 Hybrid System Introduction . . . 55
5.1.2 Stability of Discrete-time Switching Affine Systems . . . 59
5.1.3 Discrete-time Switching Affine Systems Identification . . . 61
5.2 Design of HMWM . . . 63
5.2.1 HMWM Structure . . . 64
5.2.2 Generator and Remover Stability . . . 64
5.2.3 Detectability Design . . . 66
5.2.4 Switching Rule Design . . . 69
5.3 Identification Resistance . . . 72
5.4 Conclusion . . . 73

6 Hybrid Multiplicative Watermarking: Simulation Study 75
6.1 Detection Performance . . . 75
6.1.1 Testbench 1 . . . 75
6.1.2 Testbench 2 . . . 75
6.1.3 Theorem Verification . . . 76
6.2 Switching Rule Analysis . . . 78
6.2.1 Testbench 1 . . . 78
6.2.2 Testbench 2 . . . 79
6.3 Conclusion . . . 80

7 Conclusion 83
7.1 Conclusion and Answer of the Research Questions . . . 83
7.2 Limitations and Future Work . . . 84

Bibliography 85

1 Introduction

1.1. Cyber-Physical System(CPS) and Its Vulnerabilities

Cyber-Physical Systems (CPSs) are integrations of computation, networking, and physical processes [1]. In CPSs, physical plants and remote operators exchange plant information and control signals through network communication technology (mostly wireless network communication). By using network communication technology, CPSs utilize remote computation resources, which can perform more complicated computations than local MCUs and makes it possible to optimally operate large-scale networked systems (e.g., power grid). The CPS can provide more economical, sustainable, and efficient solutions to our daily infrastructures. Nowadays, CPSs play an important role in critical fields, such as industrial manufacturing, medical healthcare, and power management systems.

Although being widely used in critical fields, the current design methods of CPSs suffer from security problems. One important reason is that networking communication is unsafe in CPSs. CPSs mainly use industrial control system (ICS) network protocols like DNP3 and Modbus [2], which are designed to handle basic communication requirements with limited computation resources rather than prioritizing the security requirement. These protocols are vulnerable to malicious attackers [3].

Because of such flaws, malicious programs such as Stuxnet [4] can use simple strategies to disturb the operation of the networked system and cause great damage without being detected. The Stuxnet worm program used a replay attack-based strategy to attack the Iranian nuclear facility, infected more than half of the computers in Iran and finally made Iran postpone their establishment of the nuclear station. However, such a dangerous worm kept stealthy for more than five years before being detected. Other examples, such as power blackouts in Brazil, are mentioned in [5].

Except for the replay attack used in the Stuxnet worm, more and more malicious attack strategies have been discovered [6], such as zero-dynamics attacks, etc. Like the Stuxnet, such attacks on CPS will cause substantial economic losses. Such attacks can even endanger human lives if the attacks target infrastructures like the power grid. Therefore, it is imperative to ensure the security of the CPSs and design securing CPSs.

1.2. Security Problem in CPSs

To design securing CPSs, the first problem is to specify the security goals of secure CPSs. Authors in [7], divide the security goals into two classes: *securing operational goals* and *securing non-operational goals*. *Securing operational goals* relates to the performance of the CPSs under malicious attack. *Securing non-operational goals* relates to security goals in computer science: integrity, availability, and confidentiality.

1. *Availability*: In CPSs, availability relates to the consistent availability of both data and system resources.
2. *Integrity*: In CPSs, integrity refers to preventing unauthorized users from modifying data, system state, and system resources.
3. *Confidentiality*: In CPSs, confidentially refers to preventing unauthorized users from viewing data, system state, and system resources.

How to design and implement securing CPSs is still an open question. Utilizing cryptography tools like RSA to improve network protocols can be a direct way. However, most CPSs require low feedback latency

and only have limited computation resources on sensors and actuators. Operating complicated cryptography tools on these sensors and actuators will introduce a considerable delay, which is terrible for CPS systems with real-time requirements. Besides, cryptography tools are designed to guarantee data security, which mainly corresponds to securing non-operational goals. For CPSs, guaranteeing the securing operational goals is also important. Securing operational goals is highly related to the dynamics of the plant and original controllers.

Because of the above reasons, the methodology to achieve resilient CPSs has drawn great attention in the control society. Researchers in [8] divide the methods into three categories:

1. *Prevention Methods*: Prevention methods are to postpone the onset of an attack. Two main trends of prevention methods are the homomorphic cryptographic method and the randomization-based method.
2. *Resilience Methods*: Prevention methods try to minimize and withstand the impact of malicious attacks. Classic examples of prevention methods include game-theoretic methods and event-triggered methods.
3. *Detection and Isolation Methods*: Detection and isolation methods aim to detect attack events and identify the attack as soon as possible. Such methods are prevalent topics nowadays. Classical methods include observer-based passive detection methods and watermarking-based active detection methods.

This thesis's main focus is the *detection and isolation methods*.

1.3. Detection and Isolation Methods

To detect and isolate malicious attacks, researchers have proposed different methods.

One popular research direction is *passive detection methods* [9–21]. Passive detection methods aim to detect malicious modifications or recover actual data using existing components in the systems without introducing extra structure to the plant. Popular passive detection methods include detector improvement methods [9–13], secure state estimation methods [14–17] and machine-learning-based methods [18–21]. Existing passive detection methods mainly have three main disadvantages:

D 1.1 Most of these methods only work under strict conditions, e.g. some secure state estimation methods are effective only if at least a certain number of sensors is safe.

D 1.2 Each of these methods is designed and effective for a limited range of attacks.

D 1.3 These methods cannot defend attackers with perfect system knowledge. For example, such an attacker can easily use malicious covert attacks to destroy the system while remaining invisible.

Active detection methods is another promising trend of attack detection methods [22–27], which do not rely solely on the knowledge of the plant dynamics, but actively modify the plant inputs or outputs to enhance attack detectability. For instance, in [22] an additive random watermark is added to the plant inputs, while the inclusion of *measurement encoding matrices* is explored in [23]. Furthermore, [24] proposed a multiplicative watermarking scheme for sensor outputs, while randomly generated parallel auxiliary systems are employed in [25]. Often, these methods rely on matched mechanisms on both the plant and the controller side to generate, validate, and then possibly remove the signals added to the plant inputs or outputs. Existing active detection methods mainly have two drawbacks:

D 2.1 Although active detection methods have been shown to improve detection capabilities against malicious agents injecting false data on the communication network, they do so under the assumption that attackers do not adapt their behaviour in response to the defence strategies. Indeed, if the attacker successfully identifies the additional security measures put in place for defence, the injected data can be suitably adapted to evade detection. From a control perspective, system identification methods can identify structures and parameters of the additional security measure.

D 2.2 Most of these methods can only detect limited types of classical attack models. For example, the measurement-coding method [23] cannot detect replay attack, while the multiplicative watermarking method [24] cannot detect control-signal-injection zero-dynamics attack.

Different methods have been proposed as countermeasures to drawbacks **D 2.1**, e.g., in [28–30], the parameters of the active diagnosis scheme are switched over time, thus changing the parameters that must be identified by an attacker to remain stealthy. On the other hand, methods like [25] naturally resist identifica-

tion, as parameters are randomly generated at each time step. Of these methods, [25, 28] rely on pseudo-random number generators to produce new parameters, which must be synchronized at the plant and the controller sides. Furthermore, a switching mechanism is proposed in [30], relying on an event-triggered strategy to define when to update the parameters of the multiplicative watermarking systems. In [31] a method based on the elliptic curve cryptography is proposed further to improve security for multiplicative watermarking (MWM).

The problem of maintaining the structures and parameters secret to attackers is similar to the privacy problem in CPSs, in which researchers try to keep control system states and parameters*private* and protect from eavesdroppers. Solutions based on Differential Privacy (DP) [32, 33] are very popular for keeping control system states and parameters private. DP methods inject additional noise into the system to mask sensitive information. The limitation of DP methods is that the reliable receiver cannot recover the true data from noisy data either. So, in CPSs, the addition of privacy noise would act as a disturbance, thus affecting control performances and increasing the false alarm rate (FAR). Based on the author's knowledge, currently, there is no research working on the relationships between DP and FAR in active detection methods, but a similar work studying the relationships between DP and FAR in fault detection can be found in [34]. Another family of approaches to keep privacy involve the use of encrypted control [35], which nevertheless can lead to possibly unacceptable time overheads, which could impact stability margins.

1.4. Research Questions and Contributions

As mentioned in 1.3, most of the current active detection methods have drawbacks. This thesis work mainly focuses on the countermeasure of Drawback **D 2.1**. This thesis work will focus on updating existing or proposing a new active detection method that can defend against malicious parameter identification. Besides, as active detection methods, the proposed approaches should also be able to detect malicious attacks. Formally, this thesis work will focus on the following question:

Problem 1. *Upgrade existing active detection methods or propose a new active detection method, such that:*

1. *It can defend against malicious parameter identification.*
2. *Multiple attacks can be detected with suitably-defined methods, including the replay attack, the control-signal-injection zero-dynamics attack and the sensor-signal-injection zero-dynamics attack. If possible, make sure the detection probability of attacks is closer to 1.* ◁

Based on the system immersion method [36], the output-coding method [26], and the multiplicative watermarking method [30], this work will address the following two specific questions.

Problem 2. *Can we propose a new active detection method based on the system immersion method [36] and the output-coding method [26] to address Problem 1?* ◁

Problem 3. *Can we upgrade the multiplicative watermarking method [30] to address Problem 1?* ◁

The main contribution of this thesis is our answer to Problem 2 and Problem 3. The contribution can be summarized as follows:

1. We propose a new system-immersion coding active detection method based on the system immersion method [36] and the output-coding method [26].
   (a) The proposed method can detect the replay attack, the control-signal-injection zero-dynamics attack, and the sensor-signal-injection zero-dynamics attack under some conditions.
   (b) The proposed method effectively disrupts the attacker's accuracy in estimating parameters, even under the known-plaintext attack scenario.
   (c) However, it is important to note that although the attacker's estimation is disturbed, if the attacker can execute the known-plaintext attack, the attacker with estimated parameters can still inject a stealthy malicious attack sequence into the system.
2. Motivated by the computational hardness problem in cryptography analysis, we propose a hybrid multiplicative watermarking (HMWM) based on the multiplicative watermarking method [30]. The proposed hybrid multiplicative watermarking is designed to be a piecewise affine (PWA) system with unobservable states. The watermarking parameters are periodically updated, following the dynamics of the unobservable states of the designed piecewise affine (PWA) hybrid systems.

(a) The method enhances the original multiplicative watermarking method's capability to detect control-signal-injection zero-dynamics attacks and replay attack under some conditions.

(b) We provide a theoretical analysis of the effects of this scheme on the closed-loop performance and prove that stability properties are preserved.

(c) Furthermore, we demonstrate that the proposed approach makes it difficult for an eavesdropper to reconstruct the watermarking parameters, both in terms of the associated computational complexity and from a systems theoretic perspective.

1.5. Outline

A portion of this thesis work is based on the paper titled "Hybrid Design of Multiplicative Watermarking for Defense Against Malicious Parameter Identification," which has been submitted to the 62nd IEEE Conference on Decision and Control, 2023.

The remainder of this thesis is organized as follows:

Chapter 2 reviews the current research progress on the security and privacy of cyber-physical systems, encompassing classical attacker models, passive detection methods, active detection methods, and privacy solutions for cyber-physical systems.

Chapter 3 presents the configuration of the cyber-physical system under consideration, the attacker model, and the testbench used. It also formulates the sub-questions that will be explored in subsequent chapters, building upon the main research questions 2 and 3.

Chapter 4 introduces our first proposed method, the *system immersion coding method*, and demonstrates its performance through numerical simulation results.

Chapter 5 introduces the second proposed method, the *hybrid multiplicative watermarking method*, along with the necessary background knowledge of hybrid systems.

Chapter 6 uses numerical simulation results to show the performance of the hybrid multiplicative watermarking method.

Chapter 7 concludes the thesis, summarises the findings, and outlines potential avenues for future research.

2

Background: Cyber-Physical Systems(CPSs)

This chapter will give basic knowledge about how to analyze security problems in cyber-physical systems (CPSs) from a control system perspective. This chapter will first introduce the model of CPSs and attackers. Then it will go through the current research progress of attack detection methods and stress the problem of malicious parameter identification for active detection methods. Finally, this chapter will relate the parameter identification problem to privacy problems in control systems and review several classical privacy solutions in control systems, supply chains and federated learning.

2.1. CPSs Model, A Control Perspective

From a control perspective, cyber-physical systems are composed of a physical plant $\mathscr{P}$ and a controller $\mathscr{C}$ [7]. The information between the controller and plant is exchanged over a communication network, thus exposing the CPS to attackers ($\mathscr{A}$). The controller estimates plant state, generates control signals, and detects faults and malicious attacks. Figure 2.1 shows the considered CPS structure. We call the network channel from the plant to controller the *plant-to-controller* channel, while the one from the controller to plant the *controller-to-plant* channel.

In the figure, y_p is the plant's output, and u is the controller's output. The signal $y'_p[k]$ indicates the sensor signal the controller receives from the plant via a communication network. The signal $u'[k]$ indicates the control signal the actuator receives from the controller. The signal $r[k]$ indicates the residual value to detect a fault or malicious attack. In the model, different attackers may have different access to the communication network. The discussion about different attackers will be posed in Chapter 2.2. In Figure 2.1, we use an attacker who has full access to $u[k]$, $y_p[k]$ and can modify these signals. We model the attacker's modification on the signal as additive modification $\Delta u^a[k], \Delta y^a[k]$. The control signal received at the plant side, and the sensor signal received at the controller side can be written as:

$$u'[k] = u[k] + \Delta u^a[k]; \quad y'_p[k] = y_p[k] + \Delta y^a[k] \tag{2.1}$$

2.1.1. A Classical Linear Configuration

For the sake of simplicity, we consider a linear system as follows: plant $\mathscr{P}$ is a linear-time-invariant (LTI) plant, while controller$\mathscr{C}$ consists of an observer ($\mathscr{O}$) with observer gain L, a static state feedback controller ($\mathscr{C}_f$) with feedback gain K and an anomaly detector ($\mathscr{R}$). Besides, we use $\hat{x}_p$ to denote the observer's estimation of the system state, and r denotes the residual value calculated by the anomaly detector. The dynamics of $\mathscr{P}$ and $\mathscr{C}$ are as follows:

$$\begin{aligned}
\mathscr{P}: &\begin{cases} x_p[k+1] = A_p x_p[k] + B_p u[k] + w_p[k]; \\ y_p[k] = C_p x_p[k] + v_p[k] \end{cases} \\
\mathscr{C}: &\begin{cases} \hat{x}_p[k+1] = A_p \hat{x}_p[k] + B_p u[k] + L(y_p[k] - C_p \hat{x}_p[k]) \\ u[k] = -K\hat{x}_p[k] \\ r[k] = y_p[k] - C_p \hat{x}_p[k] \end{cases}
\end{aligned} \tag{2.2}$$

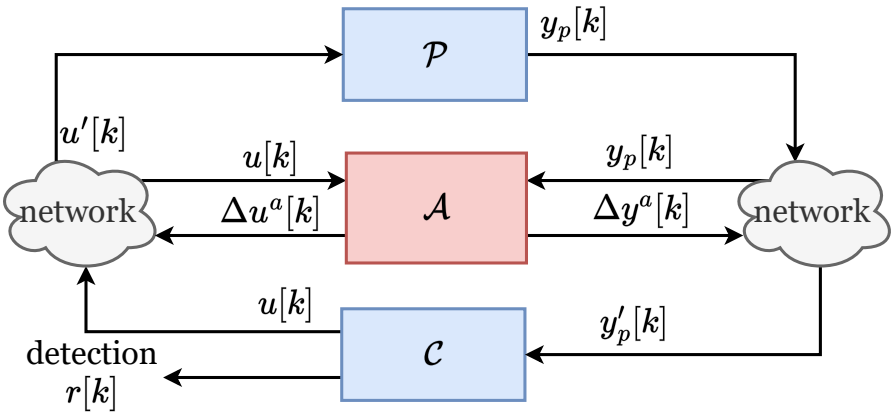


Figure 2.1: CPSs Model, A Control Perspective

where $x_p \in \mathbb{R}^{n_x}, y_p \in \mathbb{R}^{n_y}$ are the plant's state and measurement output, and $u[k] \in \mathbb{R}^{n_u}$ is the control input. The signals $w_p \in \mathbb{R}^{n_x}$ and $v_p \in \mathbb{R}^{n_y}$ represent process and measurement noise, assumed to be realizations of identically and independently distributed zero-mean Gaussian processes with covariances $\Sigma_w \succ 0, \Sigma_v \succ 0$. The $\hat{x}_p \in \mathbb{R}^{n_x}$ denotes the observer's estimation of the system state, and r denotes the residual value calculated by the anomaly detector.

This chapter will use this classical configuration to discuss attacker models and attack detection methods.

2.2. Attacker Models

Different researchers have studied different attack behaviours, and they name the attacker behaviours they studied differently. This thesis mainly adopts the framework proposed in [6]. The framework proposed in [6] models an attacker from three dimensions: *model knowledge, disclosure resources, disruption resources*. Except for these three dimensions, researchers also describe attackers from other perspectives. We summarise that *attack stealthiness, attack effect, attack objectives*, and *attack policy* are also important to model an attacker. Therefore, we will introduce attacker models through the following seven perspectives:

- *Attack Objectives*: The attacker's objective by executing the attack.
- *Model Knowledge*: The amount of prior knowledge the attacker knows about the plant, controller, observer, and anomaly detector.
- *Disclosure Resources*: Disclosure resources model the attacker's passive eavesdropping access to different sensors and actuators.
- *Disruption Resources*: Disruption resources model the attacker's access to modify the data from different sensors and actuators actively.
- *Attack Policy*: The attack policy relates to the mathematical method to execute the attack and inject the malicious sequence.
- *Attack Stealthiness*: Attack Stealthiness describes how stealthy the attacker is.
- *Attack Effect*: The attack effect describes the harmfulness of the attack on the original system.

We use the conception of ϵ*-stealthy* [37] to model the stealthiness of a malicious attack. The conception of ϵ*-stealthy* considers the attack's effect on the residual value of a system without noise and is defined as follows.

Definition 2.2.1. *(ϵ-stealthy) [37]* Suppose that the closed-loop system is at equilibrium such that the residual value $r[-1] = 0$ and that there are no unknown disturbances, i.e., $w[k] = 0$ and $v[k] = 0$ for all k. An anomaly occurring at $k = k_a \geq 0$ is said to be ϵ-stealthy if $\|r\|_{[k,k+N_r)} \leq \varepsilon$ for all $k \geq k_a$.

For *disruption resources* and *disclosure resources*, this thesis will only focus on the minimal amount of resources needed to carry out the attack. Except for attack models introduced later, other attack models exist,

such as bounded residual attacks [38], etc. Without the pretence of providing an exhaustive list, we here only introduce classical attackers that are widely discussed.

2.2.1. Eavesdropping System Identification(ESI) Attack

The eavesdropping system identification (ESI) attack is discussed in [39]. The ESI attack aims to collect input-output data and use different system identification methods to identify plant models or parameters. The ESI attack is the first-stage attack of all attack strategies that need the system model.

- *Attack Objectives*: Collect input-output data, identify plant models or parameters
- *Model Knowledge*: The ESI attack has no requirement on the prior knowledge of the system.
- *Disclosure Resources*: The ESI attack needs to disclose both controller-to-plant channel and plant-to-controller-channel.
- *Disruption Resources*: No disruption resources are needed for the ESI attack. But disruption resources can be helpful for ESI attackers.
- *Attack Policy*: System identification methods can be used as the mathematical policy of the ESI attack
- *Attack Stealthiness*: The ESI attack is 0-stealthy because $\Delta u^a[k] = \Delta y^a[k] = 0$.
- *Attack Effect*: The ESI attack does not harm the original system. But the knowledge obtained by the ESI attack can be used to design harmful attacks.

In Section 3.2, we further consider the ESI attacker that tries to identify the parameters of the active detection methods instead of identifying the system model.

2.2.2. Denial-of-service Attack

The Denial-of-service (DoS) attack prevents the actuator and sensor data from reaching their respective destinations [6]. The DoS attack can be easily detected if the networked control system uses a reliable protocol (like TCP). But the defender can be hard to distinguish a DoS attack from a poor network quality when the networked system uses an unreliable protocol (like UDP).

- *Attack Objectives*: Try to prevent the actuator and sensor data from reaching their respective destinations to disturb the system's normal operation.
- *Model Knowledge*: The DoS attack does not require knowledge of the system model.
- *Disclosure Resources*: No disclosure resources are needed for the DoS attack.
- *Disruption Resources*: The DoS attack needs the disruption resources to the corresponding controller-to-plant or plant-to-controller channels.
- *Attack Policy*: A DoS attacker can arbitrarily disrupt the sensor or controller signals. There is no strict rule or policy to design a DoS attack.
- *Attack Stealthiness*: The stealthiness of the DoS attack is related to the network protocol. For example, in the TCP network, packet loss can directly indicate a DoS attack, while with the UDP protocol, it is hard to distinguish between the DoS attack and the poor network condition.
- *Attack Effect*: The DoS attack can disturb the observer's estimate of the system state and influence control performance.

2.2.3. Replay Attack

A replay attacker first eavesdrops and records sensor data and then replays the recorded data until the end of the attack. The replay attack model is used widely, for example, in [6, 24, 40].

- *Attack Objectives*: Try to disturb the system's normal operation.
- *Model Knowledge*: The replay attacker needs no prior information on the system model.
- *Disclosure Resources*: The attacker needs disclosure access to the plant-to-controller channel to execute the replay attack.
- *Disruption Resources*: The attacker at least needs to disrupt the plant-to-controller channel. If a replay attacker can disrupt the controller-to-plant channel, it can execute a more harmful attack.
- *Attack Policy*: Assume the attacker starts recording at timestamp k_r and starts replaying at time k_a, then $\Delta y^a[k] = y_p[k - k_a + k_r] - y_p[k]$.

- *Attack Stealthiness*: The replay attack's stealthiness depends on the original system's property. In [40], the authors show the conditions for linear systems are related to specific system parameters of the plant and the controller.
- *Attack Effect*: The replay attack will disturb the observer's estimate of the system states. The effect of the replay attack depends on the property of the plant.

The replay attacker needs no prior information about the system model. But because the data are from actual system data, it looks like it is generated from a simulated system. Suppose a replay attacker can disrupt the controller-to-plant channel. In that case, it can inject harmful input $\Delta u^a[k]$ into the system while replaying recorded sensor data [27], which is similar to the covert attack in Section 2.2.5.

2.2.4. Zero-Dynamics Attack

There are mainly two types of zero-dynamics attack, the control-signal-injection zero-dynamics attack [6, 27] and the sensor-signal-injection zero-dynamics attack [37].

In the control-signal-injection zero-dynamics attack, the attacker tries injecting an extra $\Delta u^a[k]$, changing the system state without affecting the sensor output.

- *Attack Objectives*: Try to disturb the system's normal operation.
- *Model Knowledge*: The attacker needs perfect knowledge of the plant model.
- *Disclosure Resources*: No disclosure resources is needed.
- *Disruption Resources*: The attack requires to disrupt controller-to-plant channel.
- *Attack Policy*: The injected control signal is related to the invariant zeros of the system. The attack signal $\Delta u^a[k]$ can be calculated by the following equations, where x_0 corresponds to the initial state that the attack is stealthy.

$$\begin{bmatrix} vI - A_p & -B_p \\ C_p & D_p \end{bmatrix} \begin{bmatrix} x_0 \\ g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$\Delta u^a[k] = v^k g \tag{2.3}$$

- *Attack Stealthiness*: If the state of the plan is equal to x_0 when the attack starts, the attack will be 0-stealthy because the attack does not affect the sensor output. If they are not equal, an impulsive extra residual will be triggered, and the stealthiness depends on the magnitude of the extra residual.
- *Attack Effect*: The effect of the controller-data-injection zero-dynamics attack depends on the system property. If all zeros are stable, the attack will have a limited effect on the plant. If there are unstable zeros, the unstable zeros will grow geometrically and damage the plant [6].

In the sensor-signal-injection zero-dynamics attack, the attacker tries injecting an extra $\Delta y_p[k]$, pulling the sensor output to zero. The controller will be misled by the fake sensor output and keep driving the system states to the reference point even when the system already reaches the reference point, and this will finally diverge the system states.

- *Attack Objectives*: Try to disturb the system's regular operation.
- *Model Knowledge*: To successfully carry out the sensor-data-injection zero-dynamics attack, the attacker should at least have the perfect knowledge of the plant's A_p, C_p matrices.
- *Disclosure Resources*: No disclosure data is needed.
- *Disruption Resources*: The attack requires to disrupt plant-to-controller channel.
- *Attack Policy*: The injected control signal is related to the invariant zeros of the following system:

$$\begin{aligned} x_p[k+1] &= A_p x[k] + B_p u[k] + w[k] \\ y_p[k] &= C_p x[k] + \Delta y^a[k] + v[k] \end{aligned} \tag{2.4}$$

Similar to the controller-data-injection zero-dynamics attack, the $\Delta y_p[k]$ signal can be calculated as follows, where x_0 corresponds to the initial state that the attack is stealthy.

$$\begin{bmatrix} vI - A_p & 0 \\ C_p & I_{n_y} \end{bmatrix} \begin{bmatrix} x_0 \\ g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$\Delta y^a[k] = v^k g \tag{2.5}$$

- *Attack Stealthiness*: Generally, the combined dynamics of the state observer and the anomaly detector can be modelled as an LTI system as following [37]:

$$\begin{aligned} x_r[k+1] &= A_r x_r[k] + B_r u[k] + K_r y_p^{'}[k] \\ y_r[k] &= C_r x_r[k] + D_r u[k] + E_r y_p^{'}[k] \end{aligned} \tag{2.6}$$

  Suppose that $x_r[k] = 0$ at the beginning of the attack. If the received sensor signal is zero ($y_p^{'}[k] = 0$), then the residual value $y_r[k]$ will keep unchanged during the attack, and the attack will be 0-stealthy. When a plant operates at the normal operation point, the residual value should be very small, which means $x_r[k]$ and $y_r[k]$ are very close to 0.
- *Attack Effect*: Suppose the operational reference point is not zero; the controller will try to drive the system's states to the reference point. This procedure will not stop even when the system reaches the reference point because the fake sensor output is still zero. The states of the system will finally grow unbounded if A_p has at least one unstable eigenvalue [37].

2.2.5. Covert Attack

The covert attack is another widely-discussed attack strategy, such as in [41–43]. The covert attack tries to inject malicious input $\Delta u^a[k]$ and delete its effect from sensor output by injecting $\Delta y_p[k]$. In [42], the authors discuss covert attack from a transfer function model perspective and propose covert attack design for linear and nonlinear cases. In [41], the researchers divide covert attacks into measurement-dependent covert attacks and measurement-independent attacks. In [43], the authors propose a finite-time covert attack that will keep stealthy even after attack termination. This review will focus on the measurement-independent covert attack in [41] and the finite-time convert attack in [43].

The measurement-independent covert attack injects harmful input $\Delta u^a[k]$ and tries to directly delete the effect of the harmful input from sensor output to keep it stealthy.

- *Attack Objectives*: Try to disturb the system's normal operation.
- *Model Knowledge*: The measurement-independent covert attack requires a perfect plant model. In [42], the authors discuss the relationship between the quality of the model and the attack performance.
- *Disclosure Resources*: The attack doesn't require any disclosure resources of the system plant. As long as the plant is an LTI system, the attacker can delete its effect without the information of the real-time sensor data and control data.
- *Disruption Resources*: The attack need to disrupt both the controller-to-plant channel and the plant-to-controller channeattacksem *Attack Policy*: The attackser can inject arbitrary malicious controller signal $\Delta u^a[k]$ and delete its effect by add malicious sensor signal $\Delta y_p[k]$ to the original sensor output $y[k]$. For LTI plant, the $\Delta y_p[k]$ should meet the following condition:

$$\Delta y^a[k] = -\sum_{s=1}^{k} C_p A_p^{s-1} B_p u[k-s] \tag{2.7}$$

- *Attack Stealthiness*: The measurement-independent covert attack is 0-stealthy during the attack execution period. However, it can be easily detected after the attack termination.
- *Attack Effect*: The attacker can arbitrarily drive the system plant and then damage the system.

2.3. Passive Detection Methods

To detect the malicious attack, researchers have proposed different *passive detection methods*. These methods aim to detect malicious modifications or recover actual data using existing components in the systems without introducing extra structure to the plant. The *passive detection methods* can -be divided into three categories:

1. *Detector Design and Optimization Methods*: Methods in this category try to improve anomaly detectors to detect malicious attacks. Methods in [9–13] belongs to this category.
2. *Secure State Estimation Methods*: Methods in this category try to recover actual state value based on the corrupted sensor measurement. Research in [14, 15, 19] is related to this category.
3. *AI-base Methods*: Methods in this category try to use machine learning, deep learning, or other artificial intelligence methods to detect malicious attacks. Methods in [16, 18, 20, 21] belongs to this category.

2.3.1. Detector Design or Detector Optimization Methods

In [11, 12], the authors propose innovative anomaly detectors to detect stealthy attacks which will cause infinite estimation errors. The authors in [11] propose a summation(SUM) detector. Compared to the nominal χ^2 detector, the SUM detector utilizes current and historical data to expose the attacks. The authors prove that the SUM detector can detect data-deception attacks that will cause infinite estimation errors. In [12], the authors use a Euclidean-based detector to detect the data-deception attack. The authors argue that the Euclidean-based detector is more sensitive than the χ^2 detector and shows the detection performance under the data-deception attack that will cause infinite estimation error.

In [9, 10], the authors optimize the detector structure/parameter to improve the detection performance. In [10], the authors try to fuse the data from distributed local sensors and estimators to detect false data injection attacks and estimate the attack vector with an optimal alarming speed. Different estimators have different best estimation frequencies. The authors propose a convex optimization to optimize the parameter of each estimator and the weight of each estimator. In [9], the authors claim that the defender can use a bank of dissipativity-based fault detectors to detect malicious attacks. The stealthy attack vector needs to lie in a given vector space to be stealthy for a dissipativity-based fault detector. Instead of using a single detector, the defender can design multiple OR-relations detectors with different attack spaces. If an attack vector wants to be stealthy for all the detectors, it should lie in the joint space of these detectors' attack spaces. Otherwise, it will be detected. The authors then provide an optimization method to reduce the design complexity of detectors for systems whose state locates in a given neighbourhood when an attack happens.

The detector design and optimization methods have two drawbacks:

D 3.1 Most of them are designed for a single type of attack.

D 3.2 A covert attacker can easily break them with full knowledge.

2.3.2. Secure State Estimation Methods

The authors in [14] discuss how to estimate system states under sensor-data data deception attacks exactly. The authors introduce the s-sparse observability conditions and prove that if a defender wants to estimate the exact system states under the assumption that the attacker can corrupt arbitrary but constant k sensors from the n sensors, a sufficient and necessary condition is that the system should be θ-observable with $\theta + 2k \leq n$.

In [15], the researchers propose an estimation method for systems that meets the θ-obervable condition. In this method, the defender runs a bank of Kalman filters in parallel. These Kalman filters mutually use $n-k$ sensor measurement among n sensor measurements. Each filter generates a state estimation at each timestamp and calculates the residual value. The defender will use the estimation from the Kalman filter whose residual value meets the residual bound.

The authors in [19] propose a framework to estimate the system state under linear deception attacks from data both from safety sensors and corrupted sensors. The framework consists of three steps: update the estimation based on the data from the safety sensors, modify data from the unsafe sensors, and update the state estimation from the modified data. The simulation shows that the state estimation accuracy of this framework is better than the accuracy only based on data from the safe sensor while worse than the result when all sensors are safe.

Compared to other methods, the secure state estimation method focuses on detecting malicious attacks and recovering actual state estimation under attack. However, the secure state estimation methods have two drawbacks:

D 4.1 The secure state estimation method works when a certain proportion of sensors are safe. However, it is hard to guarantee this condition in real-life scenarios.

D 4.2 A direct way to meet the condition is adding extra safe-guaranteed sensors, which may need extra cost.

2.3.3. Machine-Learning-Based Methods

In [20], the authors test the performance of the classical supervised learning method. The researchers treat the attack detection problem as a binary classification problem, with the -1 label presenting the attacked class and the $+1$ label presenting the normal class. The research tests three supervised learning classification methods: support vector machine (SVM), k-nearest neighbour (KNN), and extended nearest neighbour (ENN) method. The simulation shows that these methods perform well under the article's assumption.

The authors in [18, 21] use deep learning-based methods(such as Aritificial-Neural-Network (ANN)) to

detect malicious attacks. Unlike the previous machine-learning-based methods that try to directly replace the original observer or detector, the following methods add an auxiliary detector in parallel with the χ^2 detector to detect the attack.

In [18], the authors try to detect sensor-data deception attacks using a hybrid architecture. The hybrid architecture consists of a static detector like the χ^2 detector and a dynamic detector based on the neural network. The dynamic part accepts two inputs: previous measurements and package information. The dynamic part first uses two convolutional neural networks (CNN) to read, equalize these two inputs and extract their features. The extracted features will then be sent to a recurrent neural network (RNN) to detect whether an attack happens. RNN network is widely used in application scenarios in which the dynamical behaviour of the scenario is important and is suitable for utilizing dynamic information in a control system.

In [21], the authors use a more complex neural network structure to detect malicious attacks. This article proposes a two-step detection method. An anomaly detector like χ^2 detector is first used. If the anomaly detector reports no anomaly, a modified conditional deep belief network (CDBN) is used for further analysis. In this article, this method performs better than other state-of-art ANN-based and SVM-based methods.

These machine-learning-based methods utilize advanced methods from the computer science society to detect malicious attacks. Most of them perform well under their assumptions. However, these methods still suffer from several drawbacks.

D 5.1 Some of the assumptions they used are not realistic. For example, in [20], the author assumes that the injected attack vector is sufficiently greater than the system's noise and that the mean of the injected attack vector is sufficiently greater than its variance, which is unrealistic.

D 5.2 Most of these methods are supervised learning methods, which means these methods need sufficiently large data sizes from both the normal operating condition and the attacked scenarios. However, the data from attacker scenarios are rare and hard to obtain.

D 5.3 These methods have no guarantee of detection performance. These articles do not provide theoretical proof or mathematical conditions for the detection performance.

2.4. Active Detection Methods

From Section 2.3, passive detection methods are ineffective if the attacker has complete knowledge of the system and full access to all sensors and actuators of the system. Figure 2.2 shows an example attack, where $\mathscr{P}_s$ is a simulated plant. The attacker can build a simulated system model, update the state and sensor output based on the received control input signal, and send the simulated output to the control operator while injecting an arbitrary input signal to the plant. The simulated model's signal will easily keep stealthy under passive detection method monitoring. The attack in figure 2.2 is very similar to the covert attack defined in Section 2.2. Such a scenario motivates the development of *active detection methods*. Unlike the passive detection method, *active detection methods* add some private signal/structure and use the property of these private signal/structures to detect attacks.

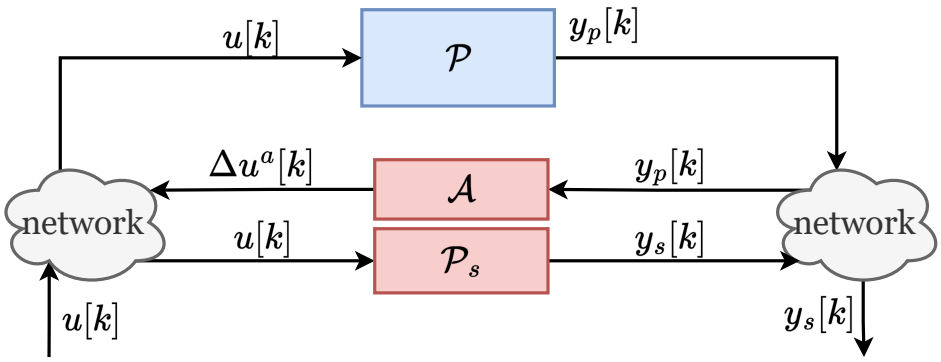


Figure 2.2: An example method that is stealthy under passive detection method

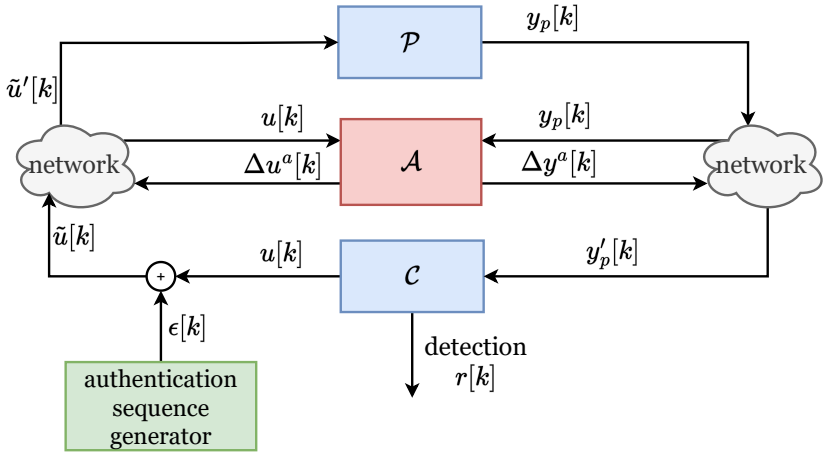


Figure 2.3: Additive Watermarking Framework

2.4.1. Additive Watermarking (AW)

Additive watermarking (AW) is first proposed to detect replay attacks [40] and has attracted much research attention. The framework of the additive watermarking is shown in Figure 2.3. In the additive watermarking framework, the defender adds some private zero-mean random value to the controller output and tries to detect the residual change under replay attack. The controller output is redesigned as

$$\tilde{u}[k] = u[k] + \epsilon[k] \tag{2.8}$$

where $u[k]$ is the original control output and $\epsilon[k]$ is the private additive input sequence (also called *authentication signal*). $\epsilon[k]$ are drawn from an i.i.d. zero-mean Gaussian distribution.

The authors prove that replay attacks will bring extra residual under additive watermarking mechanism [40]. The authors further prove the detection capability of the additive watermarking under sensor-data data deception attack [44].

There are three drawbacks of the original additive watermarking:

D 6.1 The noise on the control signal will cause a control performance degradation [40]. Due to this drawback, the covariance of the authentication signal should be carefully designed to balance detection performance and performance degradation. The design can be formulated as an optimization problem to maximize the trace of the extra residual change while keeping the performance degradation under a certain level [22, 44].

D 6.2 Additive watermarking can improve the detection performance of replay and data integrity attacks. However, the improvement lacks a divergence guarantee. Besides, because of the performance degradation,

D 6.3 Additive watermarking can improve the detection performance of replay and data integrity attacks. However, a covert attacker can easily inject a stealthy attack without knowing anything about the additive watermarking mechanism. The attacker can add a bias $b[k]$ to the controller output and then remove the effect of the bias from the sensor output as in the following equations:

$$\begin{aligned}
\tilde{u}'[k]: &\quad \tilde{u}'[k] = u[k] + \epsilon[k] + b[k] \\
y'_p[k]: &\quad y'_p[k] = C_p(A_p x_p[k] + B_p u^{'}[k]) + D_p \tilde{u}'[k] \\
\Delta y^a[k]: &\quad \Delta y^a[k] = -C_p(B_p b[k]) + D_p b[k] \\
\tilde{y}'[k]: &\quad \tilde{y}'[k] = y'_p[k] + \Delta y^a[k] = C_p(A_p x_p[k] + B_p(u[k] + \epsilon[k])) + D_p(u[k] + \epsilon[k])
\end{aligned} \tag{2.9}$$

Many extensions of the additive watermarking method have been proposed to improve the detectability or reduce performance degradation. We will go through five main extensions among them: *dynamical watermarking, switching additive watermarking, reduced additive watermarking, measurement additive watermarking* and *unstable additive watermarking*:

Dynamical Watermarking

The authors in [45] generalize the utilization of additive watermarking on more systems: ARMAX model, systems with partial observations, etc. The new utilization is called *dynamic watermarking*. The authors discuss the security guarantee of dynamic watermarking by introducing the definition of additive distortion power of malicious sensors and provide an extra test to detect malicious attacks. Furthermore, in [46], the authors consider using dynamical watermarking for linear systems affected by arbitrarily distributed noise. This article provides an example in which the system is affected by noise from Bernoulli random process, and the traditional Gaussian white noise dynamical watermarking becomes invalid. Then the authors derive sufficient and necessary conditions to design dynamical watermarking to guarantee security for LTI systems with general noise distribution.

Switching Additive Watermarking

Research in [47] shows that an attacker with a perfect system model and full access to both controller-to-plant and plant-to-controller channels can estimate the covariance property even the actual authentication signal sequence through an adaptive least mean square filter.

The *switching additive watermarking* is proposed to overcome this problem and increase the unpredictability of additive watermarking [29]. In the switching additive watermarking method, the defender periodically switches the covariance of the authentication signal among different modes based on a designed probability complex to introduce time-varying properties. Different modes have different detection performances, so switching among different modes will cause detectability degradation.

The design of the switching additive watermarking consists of two steps: The defender first generates a group of covariance modes based on different control performance loss thresholds. Then the defender will generate a probability complex by solving an optimization problem that optimizes the information entropy of the selected complex.

Reduced Additive Watermarking and Measurement Additive Watermarking

The reduced watermarking method [48] and measurement additive watermarking [49] have been proposed to improve or overcome undesired performance loss caused by additive watermarking.

The reduced watermarking method [48] reduces the performance loss by injecting the authentication signal only when needed. The authors assume the attack starting time is a random variable following a geometric distribution. The defender then determines the timestamp to insert the additive watermarking and trigger an attack alarm by solving a stochastic optimal control problem with a dynamic programming method. The stochastic optimal control problem considers the average detection delay, the false alarm rate, and the average number of injections of the authentication signal.

The measurement additive watermarking adds the authentication signal $\epsilon[k]$ to the sensor output instead of adding it to the controller output. Besides, the controller side has a watermarking remover with the same random number seed as the watermarking generator and generates a synchronized random signal. The data is encrypted and decrypted as follows:

$$\begin{aligned} \tilde{y}[k] &= y_p[k] + \epsilon[k] \\ \tilde{y}'_p[k] &= \tilde{y}'[k] - \epsilon[k] \end{aligned} \tag{2.10}$$

Because the pair of generators and removers are synchronized, the watermarking will not cause performance loss. The attack detection is based on encrypted signal $\tilde{y}[k]$ with similar detectors as the original additive watermarking.

Unstable Additive Watermarking

The authors in [50] propose a new additive watermarking such that the anomaly detector is unstable and the residual value $r[k]$ diverges to infinite under replay attacks, which makes the probability of detecting replay attacks to one. The article only considers the system in a continuous-time fashion. The watermarking signal $\epsilon[k]$ is generated as following:

$$\begin{aligned} \dot{\epsilon}(t) &= \tilde{\mathrm{A}}\epsilon(t) + \tilde{\mathrm{M}}(y(t) - C\hat{x}(t)) \\ \xi(t) &= \tilde{\mathrm{K}}\epsilon(t) \end{aligned} \tag{2.11}$$

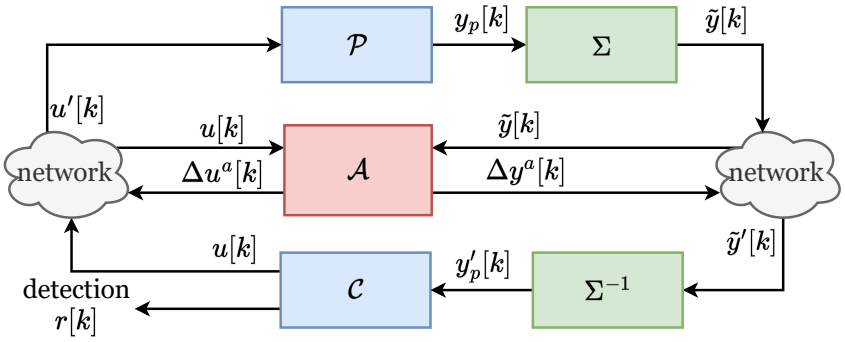


Figure 2.4: Measurement Coded Method

Where the matrices $\tilde{A}, \tilde{M}$ and $\tilde{K}$ are parameters that need to be determined by the defender. The authors also provide conditions to guarantee the stability of the original system without attack and the divergence of the residual under replay attack.

Conclusion of Additive Watermarking Method

The additive watermarking improves the detection performance with the cost of performance loss. The larger the performance loss, the higher the detection probability. The additive watermarking method can improve the detection probability of replay attacks and some sensor data injection attacks. But, the additive watermarking method cannot detect covert attacks. Different extensions of the additive watermarking method have been proposed to improve detection performance and reduce performance loss.

As one of the earliest active attack detection methods, *the additive watermarking method points out a clear direction for the general framework of active methods: adding a private structure and verifying the property of the private structure.*

2.4.2. Measurement-Coding Methods

The authors in In [23] propose a *measurement-coding method* to detect sensor output data deception attacks. Figure 2.4 shows the framework of the measurement-coding method. The sensor output data is linearly transformed by an invertible square matrix Σ and then recovered by the inverse matrix Σ^{-1} at the controller side, as shown in the following equations.

$$\begin{aligned} \tilde{y}[k] &= \Sigma y_p[k] \\ y_p^{'}[k] &= \Sigma^{-1} \tilde{y}^{'}[k] \end{aligned} \tag{2.12}$$

Because $\Sigma^{-1}\Sigma = I$, the measurement-coding method will not cause performance loss when no attacks happen (also called *normal conditions*). The authors discuss the detection performance under data deception attacks that can induce infinite estimation error ($\hat{x}[k] - x[k]$) while keeping a bounded residual. The authors prove that an invertible square matrix Σ that changes the direction of the attack vector will be sufficient to detect such attacks. In [23], the authors design Σ to make the transformed attack vector and the original attack vector orthogonal. In [28], another design method using rotation matrices is provided.

There are several flaws in the measurement-coding method:

- **D 7.1** The measurement-coding method cannot work on a single output system because there is no such feasible Σ that changes the direction of the attack vector.
- **D 7.2** The detectability of this method is only tested on data deception attacks that will cause infinite trajectory deviation. However, the detection performance for other malicious attacks is unknown.
- **D 7.3** The measurement-coding method is vulnerable to malicious parameter identification. The attacker can estimate Σ based on the input-output data and system model. With an estimated Σ, the attacker designs a stealthy harmful attack for the Σ.

The authors discuss the vulnerabilities of malicious parameter identification in [23, 28]. In [28], the authors specify the least number of input-output pairs an attacker needs to calculate Σ. The authors also claim

that if the system has internal noise, the attacker can only identify an inaccurate Σ, which is insufficient to design a new stealthy attack. Besides, the defender can frequently change the Σ on the plant and controller sides to avoid being identified. The plant and controller sides can use a synchronized pseudo-random number generator to guarantee they obtain the same value. However, generating synchronized random numbers can itself be a complex problem. Pseudo-random number generator can be unsafe and learned by the attacker, while a true random number generator cannot guarantee to generate the same value at each timestamp.

2.4.3. Moving Target Defense (MTD) Methods

Inspired by the moving target methods in computer science, researchers have proposed several moving target defence (MTD) methods to detect malicious attacks in control systems [25, 38, 41, 51–54]. These MTD methods can be divided into three categories:

1. *System Modification*: Methods in this category try to improve the detection performance by changing A_p, B_p, or C_p. For example, the methods in [41, 51].
2. *MTD by Output Switching*: Methods in this category improve the detection performance by changing the visibility of the system output. For example, the method in [38].
3. *MTD by Auxiliary system*: Methods here add an auxiliary system parallel to the plant side. Methods in [25, 52–54] belong to this category.

MTD by System Modification

In [51], the researchers proposed an MTD method to detect the sensor data injection attack by changing the system's parameters. The original plant is modified to a switching hybrid system. The dynamics of the plant under this MTD mechanism are shown as follows:

$$\begin{aligned} x[k+1] &= A_p[k]x[k] + B_p[k]u[k] + w[k] \\ y[k] &= C_p[k]x[k] + v[k] \end{aligned} \tag{2.13}$$

At each timestamp, the plant will randomly choose a sub-system. The controller and the plant sides use a pair of cryptographically secure pseudo-random number generators with the same seed. So that at each timestamp, the controller side will use the correct observer to estimate the system state. An attacker cannot generate a stealthy attack if they cannot access the random number generator. The authors provide an estimation method to isolate malicious sensors.

This method is a starting point for the MTD method. However, this method has several drawbacks:

D 8.1 For many industrial control systems, the model of the controlled objective is fixed and cannot be modified, especially for matrix A. A possible way is using different approximations of the original plant. However, using approximated models will result in performance degradation.

D 8.2 The secure pseudo-random number generator may introduce new risks.

The method in [41] overcomes drawback **D 8.1** by fixing A_p and making B_p, C_p matrices time-varying. The authors provide conditions for different groups of B and C to detect the measurement-dependent covert attack and measurement-independent covert attacks (i.e., the scaling attack). Compared to the MTD method with time-varying A, this method is more realistic because we can change B and C by changing the gain of the corresponding amplifier of sensors or actuators. However, this framework still needs synchronized pseudo-random generators to keep the plant and controller sides using the same parameters.

MTD by Output Switching

The authors in [38] propose a method to detect attacks by randomly changing the availability of the sensor data. The framework of this method is shown in Figure 2.5.

The plant side's dynamics and the observer's dynamics are as follows:

$$\begin{aligned} \tilde{y}(t) &= \Theta(t)y_p(t) \\ \dot{\hat{x}}_p(t) &= A\hat{x}_p(t) + Bu_p(t) + L\Theta(t)(\tilde{y}_p(t) - C\hat{x}_p(t)) \end{aligned} \tag{2.14}$$

where $\Theta(t)$ is a time-varying MTD parameter and can be expressed by $\Theta(t) = \text{diag}(\theta_1(t), \cdots, \theta_{n_y}(t))$ with $\theta_i(t) \in \{0, 1\}$. The authors assume the observer knows the real-time output visibility $\Theta(t)$.

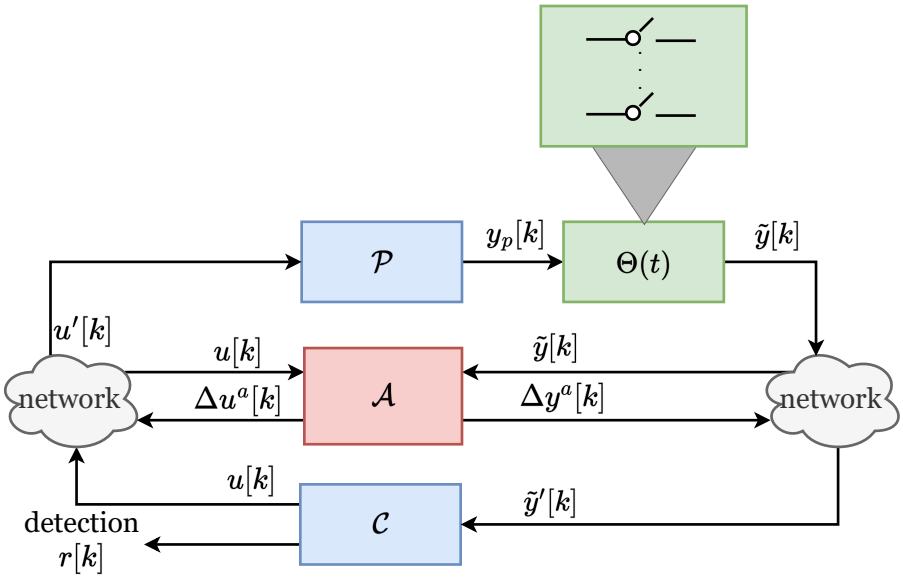


Figure 2.5: MTD by Measurement Availability Switching

For system output with n_y elements, there will be 2^{n_y} possible Θ. In [38], the authors first provide the condition that the switching probability guarantees the almost asymptotically stable almost surely (GAS a.s.). Then, the authors provide the condition for detecting the bounded residual attack. Finally, the authors formulate an optimization problem. The optimization problem considers stability and detectability and maximizes the system's resilience. The deviation of the convergence of the trajectory models resilience. With the output switching method, an attacker unaware of the MTD mechanism will easily introduce extra residual, especially when its attack strategy is based on its estimation of the system's state.

Compared to other methods, the authors have clear guidelines for designing parameters to meet stability and detectability requirements, which is a good advantage. However, the author's assumption that the observer can easily obtain the real-time $\Theta(t)$ used by the plant side (possibly by sending $\theta_i(t)$ along with $y(t)$) is not realistic.

MTD by Auxiliary systems

Modifying the system dynamics is not always feasible, and output switching may degrade system performance. Auxiliary system-based MTD methods can address these limitations. The first such method was proposed in [55], and its basic framework is illustrated in Figure 2.6. This approach introduces an auxiliary system that runs parallel to the original plant. The system's dynamics can then be described as follows:

$$\begin{aligned}
\underbrace{\begin{bmatrix} \tilde{x}_p[k+1] \\ x_p[k+1] \end{bmatrix}}_{\bar{x}[k+1]} &= \underbrace{\begin{bmatrix} \tilde{A} & \bar{A}[k] \\ 0 & A_p \end{bmatrix}}_{\mathscr{A}[k]} \underbrace{\begin{bmatrix} \tilde{x}_p[k] \\ x_p[k] \end{bmatrix}}_{\bar{x}[k]} + \underbrace{\begin{bmatrix} \tilde{B}[k] \\ B_p \end{bmatrix}}_{\mathscr{B}[k]} u[k] + \underbrace{\begin{bmatrix} \tilde{w}[k] \\ w[k] \end{bmatrix}}_{\bar{w}[k]}, \\
\underbrace{\begin{bmatrix} \tilde{y}_p[k] \\ y_p[k] \end{bmatrix}}_{\bar{y}[k]} &= \underbrace{\begin{bmatrix} \tilde{C} & \bar{C}[k] \\ 0 & C_p \end{bmatrix}}_{\mathscr{C}[k]} \underbrace{\begin{bmatrix} \tilde{x}_p[k] \\ x_p[k] \end{bmatrix}}_{\bar{x}[k]} + \underbrace{\begin{bmatrix} \tilde{v}[k] \\ v[k] \end{bmatrix}}_{\bar{v}[k]}
\end{aligned} \tag{2.15}$$

At each time step, the plant and controller will generate the same set of parameters for the auxiliary system using the same pseudo-random number generator. The system's output, denoted as $y[k]$, is not affected by the auxiliary state $\tilde{x}_p[k]$. Hence, without an attack, there will be no impact on the system's performance. However, if an attacker attempts to disrupt the system state, the dynamics of the auxiliary state $\tilde{x}_p[k]$ will also be affected due to their coupling. This increases the likelihood of detecting a malicious attack. Moreover, the parameters of the auxiliary system change frequently and randomly, making it difficult for an attacker to use the replay strategy and to identify the system parameters accurately.

In their work [55], the authors propose an optimal parameter design method to improve the detection performance. This method involves two steps:

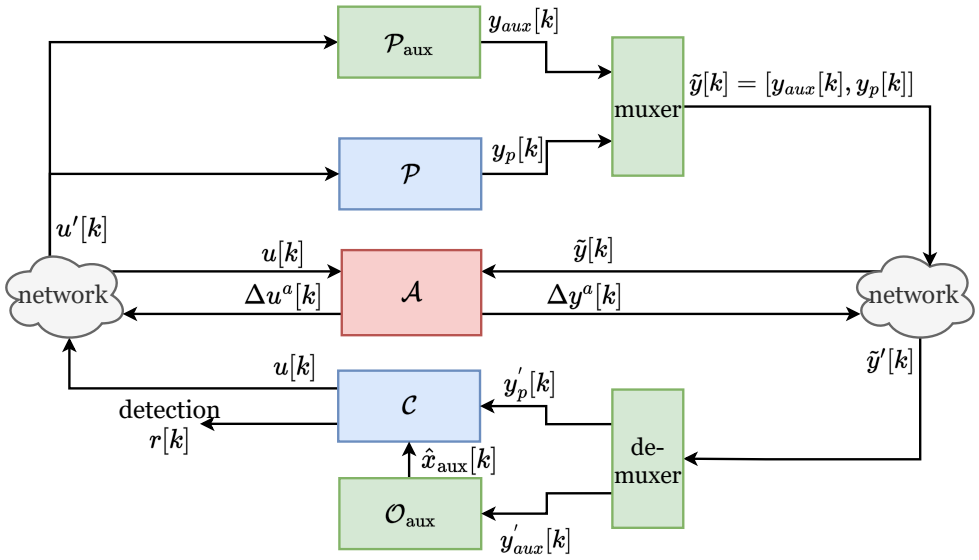


Figure 2.6: MTD Method by Auxiliary Systems

1. Design the covariance $\Sigma_{\bar{B}}$ of the random $\bar{B}[k]$ to maximize the detection performance. The detection performance is modelled by the KL divergence between the residual with and without attack.
2. Design the covariances $\Sigma_{\bar{A}}, \Sigma_{\bar{C}}$ of the random $\bar{A}[k]$, $\bar{C}[k]$ to maximize the amount of information about the state of the attacked original plant in $\tilde{y}[k]$. The Fisher information matrix models the amount of information.

Then in [25], this method is extended by introducing nonlinearities to limit the information revealed to an attacker. The output dynamic of the nonlinear moving target method is shown as follows, where $G[k]h(x[k])$ is the nonlinear part.

$$\underbrace{\begin{bmatrix} \tilde{y}_p[k] \\ y_p[k] \end{bmatrix}}_{\tilde{y}[k]} = \underbrace{\begin{bmatrix} \tilde{C} & \bar{C}[k] \\ 0 & C_p \end{bmatrix}}_{\mathscr{C}[k]} \underbrace{\begin{bmatrix} \tilde{x}_p[k] \\ x_p[k] \end{bmatrix}}_{\tilde{x}[k]} + \begin{bmatrix} G[k]h(x[k]) \\ 0 \end{bmatrix} + \underbrace{\begin{bmatrix} \tilde{v}[k] \\ v[k] \end{bmatrix}}_{\tilde{v}[k]} \tag{2.16}$$

The nonlinear part is designed to be very small when the $x[k]$ is in the normal operation region, while it is designed to be very large when the $x[k]$ is not in the normal operating region.

The moving target methods mentioned above propose a good way to detect malicious attacks while avoiding being identified. However, they still have some drawbacks:

D 9.1 These methods do not provide guidance on designing a stable auxiliary system. As the auxiliary system operates in a linear-time-varying fashion, ensuring uniform stability is crucial.

D 9.2 Synchronization of the parameter between the plant and controller sides is required for all these methods. This can be achieved by a pair of synchronized pseudo-random number generators but may introduce extra risks.

D 9.3 The auxiliary system requires knowledge of the state information $x[k]$ of the original plant, which is challenging to obtain directly but can be inferred from the sensor output.

D 9.4 The output of the auxiliary system needs to be sent to the controller side, which brings extra communication burden.

To address the issues of stability and state information mentioned in drawbacks **D 9.1** and **D 9.3**, the authors of [54] propose using a discrete-time switched system as an auxiliary system and coupling its state with the measurement of the original plant. Replacing the randomly generated system with a switched system makes it easier to create an auxiliary system that meets certain desired properties, such as stability. The authors also claim that the auxiliary system can be designed to have similar dynamics to the original plant. Such an auxiliary system can confuse the attacker, making it difficult to differentiate between the auxiliary system output and the original plant output.

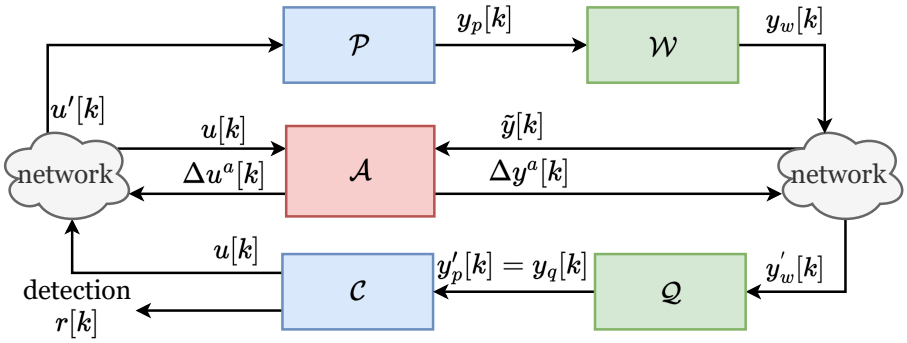


Figure 2.7: Multiplicative Watermarking

Conclusion of Moving Target Defence Method

Inspired by computer science society, various moving target defence methods are designed to detect attacks actively. However, most of these methods have the following limitations:

D 10.1 They require *explicit* synchronized parameters at the plant and controller sides. Two common methods are the synchronized pseudo-random generator and securing side-channel communication. But these methods may bring extra risks or burdens.

2.4.4. Multiplicative Watermarking (MW)

The authors in [24] proposed a multiplicative watermarking method. The framework of the multiplicative watermarking is shown in Figure 2.7. In multiplicative watermarking, the sensor output undergoes processing by a stable dynamical system (referred to as the watermarking generator $\mathcal{W}$) and is then equalized by the corresponding stable inverse system (referred to as the watermarking remover $\mathcal{W}$).

The dynamics of the $\mathcal{W}$ and $\mathcal{Q}$ are as follows. The parameters of $\mathcal{W}$ and $\mathcal{Q}$ switch periodically and follow a prior-defined sequence to guarantee synchronization. Compared to the measurement-coding method, the multiplicative watermarking method can be used for single-output systems because its detection performance does not rely on the difference in direction. Because of the inverse relation between the $\mathcal{W}$ and $\mathcal{Q}$, signal $y_p[k]$ will be accurately reversed when no attack happens so that the multiplicative watermarking will not cause performance loss.

$$\begin{aligned} \text{generator} \quad (\mathcal{W}(\theta)) &: \begin{cases} x_w[k+1] = A_w x_w[k] + B_w y_p[k] \\ y_w[k] = C_w x_w[k] + D_w y_p[k] \end{cases} \\ \text{remover} \quad (\mathcal{Q}(\theta)) &: \begin{cases} x_q[k+1] = A_q x_q[k] + B_q y_w[k] \\ y_q[k] = C_q x_q[k] + D_q y_w[k] \end{cases} \end{aligned} \tag{2.17}$$

The effectiveness of the multiplicative watermarking technique has been studied under replay and sensor data injection attacks [24, 37]. The following findings were made:

- For replay attack, if the replayed data and current watermarking remover come from different parameter groups ($\mathcal{W}(\theta_1)$ and $\mathcal{Q}(\theta_2)$), equalizing the replayed data will not be trivial and will result in additional residual. The detection performance mainly depends on the difference between two different sets of parameters.
- In sensor data deception attacks, such as zero-dynamics sensor data injection attacks, if the attacker is unaware of the watermarking generator and directly injects data intended for the original plant, the injected data will not be stealthy for the watermarking remover and will cause additional residual.

There are several limitations to the multiplicative watermarking method, including:

D 11.1 If an attacker successfully executes a malicious parameter identification to estimate the parameter before switching or estimate all sets of parameters, the prior-defined sequence and the switching period, they can design an attack based on the identified model. If new sets of parameters are generated periodically by synchronized pseudo-random generators, which will bring extra risks.

D 11.2 The detection performance for replay attacks heavily depends on the parameter mismatch between the watermarking generator and remover. This may cause the following problems:

1. The detectability depends on the difference between two sets of parameters, and if the difference is too small, the attack will not trigger an alarm. Additionally, no explicit guideline exists to design appropriate parameters to guarantee detectability.
2. The detection delay depends on the switching frequency, which in some extreme cases, can cause a significant delay in detection.

D 11.3 The multiplicative watermarking method cannot detect control-signal-injection zero-dynamics attacks.

Two extensions [30, 31]have been proposed to solve the drawback **D 11.1**. But, works on designing appropriate parameters to improve detectability are still lacking.

Switching Multiplicative Watermarking

In [30], the authors propose an event-triggered *implicit synchronization* switching protocol to achieve synchronized switching aperiodically and formalize the successful implicit synchronization: trigger-synchronized, switch-synchronized, jump synchronized and output synchronized.

To achieve this, the watermarking generator predicts the switching time of the watermarking remover and changes its parameter and state when switching happens at the remover. The sequence of parameter changes is predefined, and the new state is designed to ensure the switching occurs at the remover and to reduce the switching visibility.

This switching protocol improves the detection performance because the injected measurement signal may cause a switch at the watermarking remover and trigger an alarm.

In [31], the authors propose an improved switching mechanism to enhance the security of multiplicative watermarking. Instead of switching among predefined parameters in a prior-defined sequence, they propose a framework to generate parameters for a stable finite impulse response (FIR) filter based on the real-time sensor output $y_p[k-1]$ using cryptographic elliptic curves.

The security of this method is based on two aspects:

1. The sensor output $y_p[k-1]$ has some true random part (caused by physical noise) which includes a true random component caused by physical noise, which is not defined prior and is unknown and unpredictable to the attacker. This means that the parameter generated by the elliptic curve function is also unpredictable, making it difficult for the attacker to mount a successful attack.
2. The parameter generated by the elliptic curve function is based on cryptographic elliptic curves, which are very hard to break when using appropriate parameters.

2.4.5. Other Methods

In addition to the methods discussed above, several other active detection methods exist. Two are the blended method combining additive watermarking and the auxiliary function [27], and the output-coding method [26]. We will analyze these methods below.

A Blended Active Detection Method

A blended active detection method [27] combines synchronized additive watermarking generators/removers and a predefined auxiliary bijective nonzero nonlinear scalar function whose value is related to the system input $u[k]$. The framework of this blended method is illustrated in Figure 2.8. The encrypted controller output is decrypted by the controller using an additive watermarking remover, while the sensor output is first multiplied by the nonlinear scalar function and then encrypted using another additive watermarking generator.

The authors have demonstrated the efficacy of this method against replay attacks, covert attacks, and control-signal-injection zero-dynamics attacks. It is proven that in the presence of an attacker who does not know the additive watermarking sequence, these attack models will always result in an additional residual. Additionally, as the pairs of additive watermarking generators/removers are synchronized, and the nonlinear scalar function is bijective, this blended method does not result in any performance loss without an attack.

This method has two possible drawbacks:

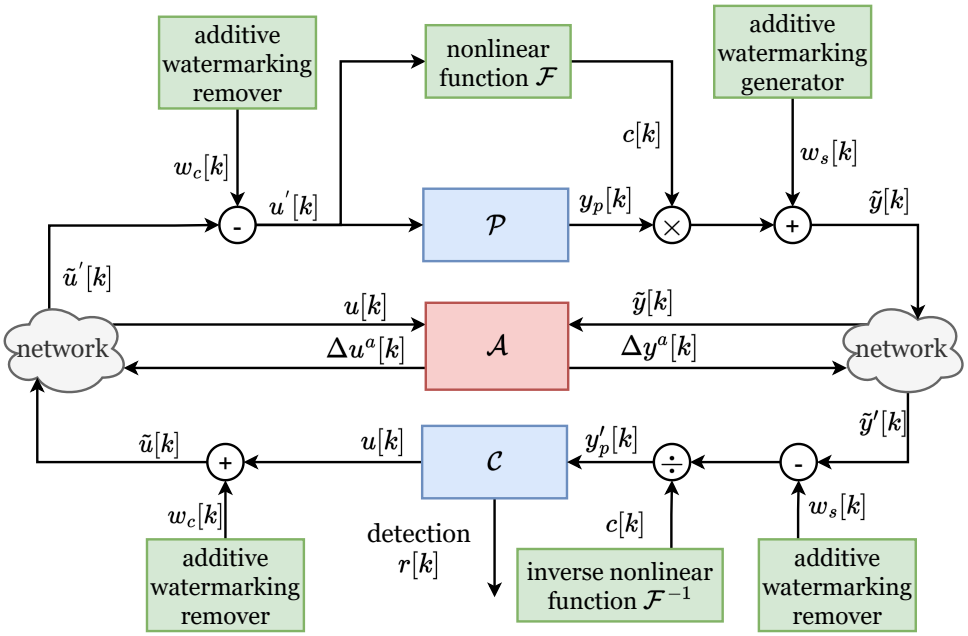


Figure 2.8: Blended Method

D 12.1 It requires two pairs of synchronized additive watermarking generators and removers. However, as discussed before, synchronized pseudo-random number generators and removers may bring additional risks.

D 12.2 Although this method guarantees that different attacks will cause extra residuals, whether the extra residual is large enough to trigger an alarm is not discussed.

Despite these drawbacks, this method provides an approach to address controller-data deception attacks by *incorporating the control signal information into the system output.* Because the controller always has the information of the true $u[k]$, any modification to the controller data will cause a mismatch in the information, potentially triggering an alarm.

Output-Coding Method

An output-coding method is proposed for detecting replay attacks in [26]. Figure 2.9 shows the framework of the output-coding method. Like the blended method, the output-coding mechanism also merges input information into the system output but in an additive manner.

The modified system output dynamics are as follows: the Γ is private and unknown to the attacker.

$$\tilde{y}[k] = y_p[k] + C\Gamma u[k-1] = Cx_p[k] + C\Gamma u[k-1] \tag{2.18}$$

The authors prove that by finding a Γ value that causes $\triangleq (I-KC)(A+BL) - KC\Gamma L$ to have an unstable eigenvalue, replay attacks can be detected with probability 1. However, the method is vulnerable to malicious parameter identification. The method may be compromised if an attacker estimates Γ and injects a covert attack based on the estimated value.

2.5. Privacy Problem in Networked Control Systems and Other Fields

Section 1.3 highlights the challenge of maintaining the secrecy of system structures and parameters from potential attackers. This challenge is similar to the issue of privacy in networked control systems, which is a hot topic in this area. This section will discuss the fundamental requirements and traditional privacy solutions for networked control systems.

Privacy is also prevalent in other scenarios, such as supply chain and federated learning. These scenarios involve distributed components exchanging data over networks, and keeping their actual information confidential from other components and outsiders is crucial. The privacy solutions developed for these scenarios

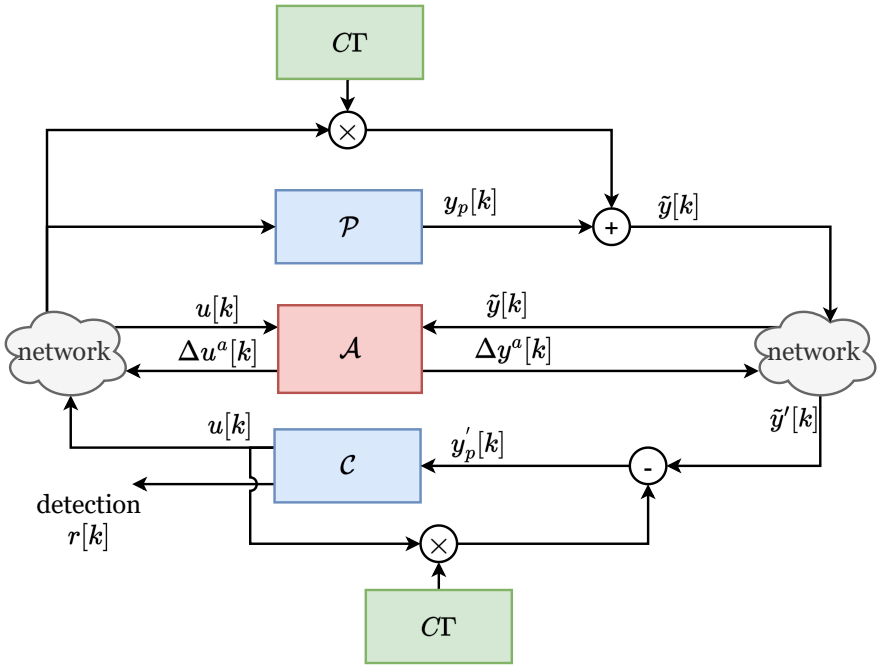


Figure 2.9: output-coding Method

can provide valuable insights. Therefore, this section will also present some privacy solutions developed for supply chain and federated learning.

2.5.1. Privacy Solutions in Networked Control Systems

The operation of large-scale systems, such as the smart power grid, depends on information from various nodes. However, the sensitive nature of the information obtained from these nodes requires that it be kept confidential [56]. Furthermore, some commercial black-box systems may have trade secrets in their structures or parameters [32]. Consequently, addressing the privacy concerns of control systems has become an essential area of research in the past two decades.

The privacy problem in a control system involves ensuring the confidentiality of sensitive information and preventing outsiders' unauthorized access to system knowledge. Various research studies have addressed privacy concerns related to different system components. In this section, we mainly discuss two kinds of privacy problems:

- *State Privacy*: This pertains to keeping the system's state undisclosed or ambiguous to outsiders. Researchers have proposed several methods to address the state privacy problem [57–61].
- *Parameter or Structure Privacy*: This involves keeping the model or parameters undisclosed or ambiguous to outsiders. Researchers have proposed various methods to address the parameter or structure privacy problem, including those in [32, 33, 62, 63].

In recent years, researchers have proposed various methods to enhance control or cyber-physical systems' privacy. In this review, we will introduce several methods and divide them into four categories:

- *Differential Privacy (DP) Based Methods*: The differential privacy (DP) method is an important method in the computer science field that aims to solve the secrecy problem in multi-party computation scenarios. DP methods add artificial noise to the original data to achieve relative privacy. Researchers in control societies have tried to transplant the DP method to improve the privacy of control systems. Methods in [32, 33] belong to this category.
- *Noisy Encoding Methods*: The noisy encoding method adds artificial noise to the original data and sends the real-time noise value to the observer side. Methods in [61] belong to this category.
- *State-Secrecy Coding Methods*: The state-secrecy coding methods leverage the acknowledgement mechanism of networked control systems to confuse eavesdroppers about the system state while keeping the

confidential state information available for the insider. Methods in [57–60] belong to this category.

- *Auxiliary System Methods*: The auxiliary system methods add additional systems parallel to the original plant to enhance the privacy of the control system. Methods like decoy-based moving target [63] and dynamic masking [62] belong to this category.

Differential Privacy (DP) Methods

Differential privacy (DP) is a computer science method to share dataset patterns while preserving individual information privacy [64]. It achieves this goal by adding artificial noise to the query result, making inferring individual information difficult. DP methods have been extensively studied in the Multi-Party Computation field.

Appropriate adjacency relations need to be defined to use DP methods, and the type and variance of artificial noise must be carefully chosen to meet the desired privacy standard. Researchers have attempted to use DP methods to achieve state or local data privacy for distributed filters, and recent works have focused on improving parameter and structure privacy by adding noise [32, 33]. In this review, we will briefly discuss these works.

In [32], the authors propose a method to preserve the privacy of identifying parameters of finite impulse response (FIR) dynamical systems by adding additive input or output noise. The optimal noise design is achieved by maximizing the estimation error of the FIR parameters through least square estimators while ensuring that the system's output variance is bounded. This method can be seen as a differential privacy approach.

Similarly, in [33], the authors aim to protect sensitive parameters in multi-agent linear time-invariant systems using differential privacy methods. They propose adding noise to agents' output to protect the sensitive parameters while constraining the controller's estimation error of the eigenvalues of the matrices related to these parameters.

While differential privacy methods have demonstrated their effectiveness in secure multi-party computation, they have drawbacks.

D 13.1 The additive noise used to ensure privacy may cause a performance loss, and designers must balance privacy levels with performance loss.

D 13.2 As the number of input-output samples obtained by an attacker increases, the amount of noise needed to ensure privacy also increases, leading to larger performance loss. This means this method may not be suitable for long-term operating control systems.

Noisy Encoding

In [61], the authors propose a novel encoding mechanism for noisy autonomous systems which can mitigate the performance loss caused by DP methods. The method involves generating an additive noise, integrating the original measurement, concatenating the real-time noise sequence, encoding the resulting data, and sending it to the controller. The control centre can then decode the data, recover the noise sequence, and subtract the noise from the noisy measurement to obtain an accurate measurement.

To encode the data, the authors introduce a matrix M_i to map the original measurement $y_i[k]$ and the generated noise $\xi_i[k]$ to a coded measurement $\mathbf{y}_i^*[k]$. The encoding procedure and the decoding procedure are then as follows:

$$\begin{aligned} &\text{Encoding procedure: } \mathbf{y}_i^*[k] = M_i \cdot \mathbf{y}_i[k] = M_i \cdot \left[y_i[k], \xi_i[k]\right]^T \\ &\text{Decoding procedure: } y_i[k] = \mathbf{e}_1 \cdot M_i^{-1} \cdot \mathbf{y}_i^*[k] = [1,0] \cdot \mathbf{y}_i^*[k] \end{aligned} \tag{2.19}$$

This method has two limitations:

D 14.1 The authors have only verified the method's effectiveness on autonomous LTI systems, and its performance on general dynamical systems is still unknown.

D 14.2 This method needs extra communication burden.

State-Secrecy Code

In [57], the state-secrecy code method is proposed to protect the system state from passive eavesdroppers in a noisy autonomous LTI system under the assumption that the acknowledgement is reliable and the passive eavesdroppers that cannot modify or block the acknowledgement. The method involves subtracting the most

recently successfully received state at t_k from the current state as follows. This can impair the eavesdropper's estimation while keeping the user's estimation optimal.

$$z[k] = x[k] - A^{k-t_k} x[t_k] \tag{2.20}$$

For unstable LTI systems, a critical event can be triggered to cause an unbounded estimation error for the eavesdropper. In practice, the system can actively trigger a critical event using more complex mechanisms such as cryptographical methods and then use the state-secrecy coding method to take over.

The state-secrecy code method is extended to stable systems in [58]. In [60], the method is adapted for unreliable acknowledgements and quantization errors. The state-secrecy method has been successfully applied in trajectory-tracking motion planning of robots [59].

While the method is a low-cost privacy solution, it has one significant drawback:

D 15.1 This method is only discussed for autonomous systems, but most CPSs are not autonomous.

Auxiliary System Methods

To improve the privacy of a system's structure or parameters, researchers in [62, 63] propose adding auxiliary structures.

In [63], the researchers suggest a decoy-based moving target method, which involves using S_i decoy-based auxiliary systems with similar dynamics to the original plant but with different artificial noise. Parallel measurements from all systems are randomly permuted and transmitted to the controller side, which then recovers the original measurements and generates a control signal for each system.

Since the controller side recovers the permuted signal, this mechanism does not cause performance loss. Additionally, the attacker will have difficulty choosing the correct input-output pair from the permuted signals due to the similar dynamics of all decoy-based auxiliary systems and the randomly generated permutation matrix. This method reduces the attacker's probability of successfully identifying the system.

The decoy-based moving target method has two limitations:

D 16.1 The decoy-based moving target method requires a synchronized pseudo-random number generator to ensure the same permutation matrix, which may introduce additional risk.

D 16.2 This method increases the communication burden, and the probability reduction is related to the number of decoy-based auxiliary systems, creating a trade-off between extra communication and probability reduction

In [62], the authors propose a dynamic masking method to protect the structure of the plant. As shown in Figure 2.10, the method involves adding a simulated plant $-\hat{P}$ and a cipher plant S at the plant side. The $\hat{P}$ has the same transfer functions between $y[k]$ and $u[k]$ as P but ignores the effect of noise on the $y[k]$. Similarly, another pairs of simulated and cipher plants are added at the controller side with opposite transfer functions.

Each plant updates its state and output based on $u[k]$ at each timestamp. The accumulated signal from all the plants $-\hat{P}$, P and S is sent to the controller side, and the output of plants on the controller side is used to recover the output of the original plant. Because the auxiliary systems at the controller side and the plant side have matching dynamics, the output of the original plant will be accurately recovered at the controller side if the initial state of the auxiliary system is the same.

The authors propose a definition of *Privacy of a property of G with respect to an adversary* to evaluate the privacy performance of the dynamic masking method. This method ensures that the attacker can only obtain the model of the cipher plant S when trying to estimate the plant model. Moreover, the dynamic masking method can enhance the system's security by utilizing a S with a distinct malicious attack vector space compared to G. This makes detecting attacks targeting the original plant $G(s)$ possible.

However, the dynamic masking method can only guarantee performance and security if the system plant is initially unknown to the attacker. In scenarios where the dynamic of the plant is public, the attacker can estimate the model of the $S(s)$ and perform a covert attack.

2.5.2. Privacy Solution in Supply Chain and Federated Learning

As mentioned, the privacy solutions developed for supply chain and federated learning scenarios can provide valuable insights. Therefore, this section will also present some privacy solutions developed for supply chain and federated learning.

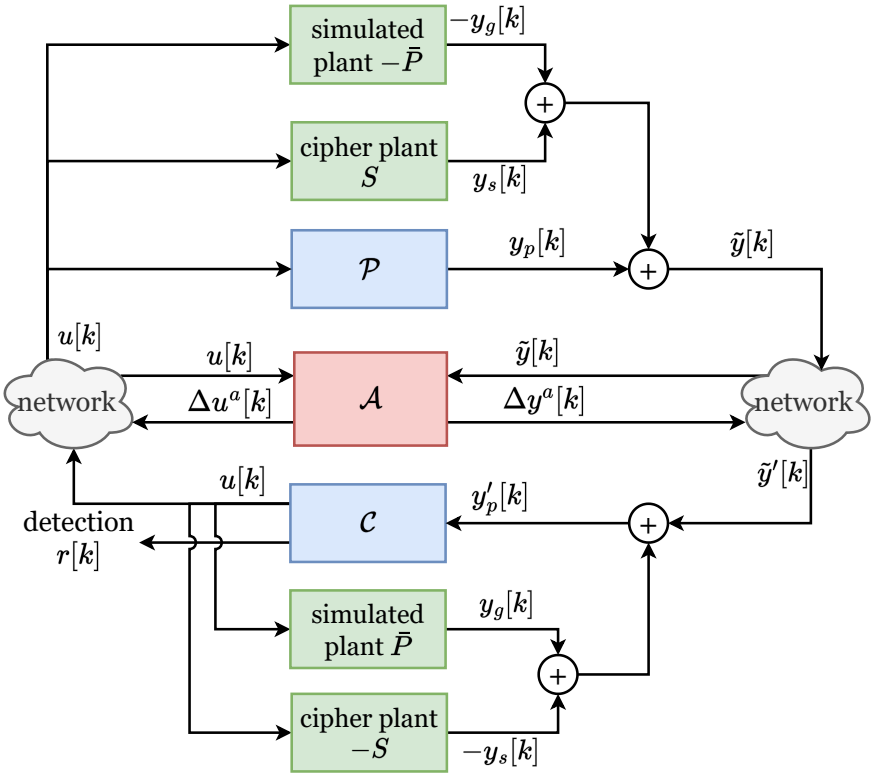


Figure 2.10: Dynamic Masking Method

In [65], the authors consider solving the state trajectory privacy problem in the supply chain networks. The supply chain system in this article can be modelled by an autonomous LTI system as follows:

$$\begin{aligned} x[k+1] &= Ax[k] \\ y[k] &= x[k] \end{aligned} \tag{2.21}$$

This article tries to keep the privacy of A by adding noise to the output $y[k]$. The sensitivity of the parameter A is defined as the ones-step trajectory difference from the same initial condition. The sensitivity proposed in this article provides a first-step solution of using the DP method to guarantee the parameter privacy of the control systems.

In [36], researchers propose a privacy-preserving federated learning system using the system immersion method from nonlinear control theory. Each decentralized centre maps its model to a higher-dimensional space (the so-called *target system*) using the immersion map, trains its model using Stochastic Gradient Decent (SGD) in the higher-dimensional space, and sends the model to the aggregator. The aggregator aggregates the local models and sends the aggregated model to the central server. The server maps the aggregated model back to the original space and distributes it to the decentralized centres.

The authors provide a linear example of the system immersion coding method, which uses random matrix encryption and includes a b^t component that acts as one-time pad encryption, making the system-immersion method unconditionally secure without bringing any performance loss. Moreover, matrices operations have much lower computational overhead than cryptographic encryption methods.

2.5.3. Conclusion for Privacy Problems

Maintaining the secrecy of system structures and parameters from potential attackers is akin to the structure or parameter privacy problem in networked control systems. In this section, we reviewed the privacy problem in the control systems. A direct consideration is combining privacy solutions with active detection methods to protect the secrecy of parameters. However, existing privacy solutions for control systems have limitations when doing so.

One approach is to use DP methods to guarantee the privacy of the private component. However, control system attack and fault detection tasks are noise-sensitive, and adding artificial noise can increase the false-

alarm rate and harm regular operation. Researchers in [34] study the relationship between false-alarm rate and artificial noise properties in the anomaly detection scenario. Also, defining proper adjacency relations for dynamical systems is challenging, which limits research on parameter/structure privacy for control systems.

Other methods, such as the state-secrecy method, have only been verified in autonomous system scenarios. Under a feedback loop, attackers can postulate more information from the control signal $u[k]$, and the performance of these methods in general system cases is unknown.

Moreover, privacy and security requirements differ. While privacy solutions can prevent passive attackers from inferring information about sensitive parameters or data, attackers without such information can still execute stealthy attacks by replaying attacks or other ill-designed methods. Privacy problems focus on preventing attackers from inferring information, while security problems require preventing attackers from imitating the structure's behaviour with sensitive parameters.

Although not discussed in this section, encrypted control [35] is another approach to maintaining privacy. However, the overheads it introduces may be unacceptable and could impact stability margins.

2.6. Conclusion

In this chapter, we first propose the framework of Cyber-Physical Systems (CPSs) from a control perspective. Then we introduce a modelling framework to model attacks on CPSs from seven perspectives (*model knowledge, disclosure resources, disruption resources,* emphattack stealthiness, *attack effect, attack objectives,* and *attack policy*) and we introduce several classical attack models. A summary of the *model knowledge, disclosure resources, disruption resources* and *attack stealthiness* properties of each attack model in Table 2.1.

This chapter also introduces attack detection methods. We divide attack detection methods into two classes. The first class is the passive detection method, which does not change the structure of the plant side. One of the main drawbacks of the passive detection method cannot defend against attackers with perfect plant knowledge. The second class is the active detection methods, which detect malicious attacks by adding some private structures to the system. A summary of these active detection methods can be seen in Table 2.2. The table tries to compare these methods from the following aspects:

1. *detectability*: This aspect is about the method's detection performance for different attack models.
2. *vulnerabilities of parameter identification (for original methods)*: This aspect considers whether the active detection method can defend malicious parameter identification. Here we only discuss the original version of the method.
3. *performance loss*: This aspect relates to whether the method will cause performance loss when there is no attacker.
4. *explicit synchronization*: This aspect considers two parts: whether the method needs explicit synchronization for parameters between plant and controller; if the method has an extended version to defend against malicious parameter identification attacks; whether the extended version need.

The table shows that there is no perfect active detection method. Some of them can only guarantee a high detection probability of a limited amount of attack, while most of them are vulnerable to malicious parameter identification attacks. Indeed, if the additional security measures put in place for defence are successfully identified by the attacker, the injected data can be suitably adapted to evade detection.

Different methods have been proposed as countermeasures to malicious parameter identification. Most of them need explicit synchronization [25, 28], which always requires pairs of synchronized pseudo-random generators and may introduce extra risk. The switching mechanism in [30] does not require explicit synchronization and relies on an event-triggered strategy to define when to update the parameters of the multiplicative watermarking systems.

The problem of maintaining the structures and parameters secret to attackers is similar to the privacy problem in CPSs. This chapter introduces privacy solutions in control systems. Their limitations make them unsuitable for directly utilizing to protect parameter secrecy in active detection methods. This section also introduces privacy solutions in fields like supply chain and federated learning, these methods can be enlighting.

Table 2.1: Summary of Attacker Models

models	attack objectives	model knowledge	disclosure resources	disruption resouces	attack stealthiness
eavesdropping system identification attack	1. collect input-output data, 2. identify plant models or parameters	no requirement	controller-to-plant and plant-to-controller	no requirement	0-stealthy
DoS attack	disturb normal operation	no requirement	no requirement	controller-to-plant or plant-to-controller	1. poor network: stealthy 2. good network: detectable
replay attack (simple)	disturb normal operation	no requirement	plant-to-controller	plant-to-controller	depends on system properties
replay attack (covert)	disturb normal operation	may need knowledge	plant-to-controller	controller-to-plant and plant-to-controller	depends on system properties
control-signal-injection zero-dynamics attack	disturb normal operation	perfect knowledge	no requirement	controller-to-plant	stealthy
sensor-signal-inection zero-dyanmics attack	disturb normal operation	perfect knowledge	no requirement	plant-to-controller	stealthy
covert attack	disturb normal operation	perfect knowledge	no requirement	controller-to-plant and plant-to-controller	stealthy

Table 2.2: Summary of Active Detection Methods

method	detectability	vulnerabilities of parameter identification (for original methods)	performance loss	explicit synchronization
additive watermarking	1. increase detectability for replay attacks, 2. no divergence is guaranteed 3. ineffective for covert attack	yes	1. measurement additive watermarking: no 2. others: yes	1. measurement additive watermarking: yes 2. others: no
measurement coded	1. infinite estimation error lead to divergence	yes	no	1. original: no 2. extension: yes
moving target	1. increase detectability 2. no divergence guaranteed	no	1. by system modification: yes 2. by output switching: yes 3. by auxiliary system: no	yes
multiplicative watermarking	1. increase detectability for most attacks 2. no divergence guaranteed 3. ineffective for control-signal zero-dynamics attack	yes	no	1. original : no 2. switching : yes
blended method	1. crease detectability for most attacks 2. no divergence guaranteed	no	no	yes
output-coding method	1. increase detectability for replay attacks, 2. no divergence guaranteed 3. performance unknown for other attacks	yes	no	not discussed

3

Problem Formulation

We consider a cyber-physical system composed of a physical plant $\mathscr{P}$ and a controller $\mathscr{C}$, containing a steady-state Kalman filter, a static state feedback controller and an anomaly detector. The information between the controller and plant is exchanged over a communication network, thus exposing the CPS to attacks. To counteract this, we suppose the CPS is equipped with an active detection pair $(\mathscr{C}_e, \mathscr{C}_d)$. Figure 3.1 shows the considered CPS structure.

3.1. System Model

The plant is modelled as an LTI system with dynamics

$$\begin{aligned} x_p[k+1] &= A_p x_p[k] + B_p u[k] + w_p[k]; \\ y_p[k] &= C_p x_p[k] + v_p[k] \end{aligned} \tag{3.1}$$

where $x_p \in \mathbb{R}^{n_x}, y_p \in \mathbb{R}^{n_y}$ are the plant's state and measurement output, and $u[k] \in \mathbb{R}^{n_u}$ is the control input. The signals $w_p \in \mathbb{R}^{n_x}$ and $v_p \in \mathbb{R}^{n_y}$ represent process and measurement noise, assumed to be realizations of identically and independently distributed zero-mean Gaussian processes with covariances $\Sigma_w \succ 0, \Sigma_v \succ 0$. We assume that:

A 1.1 All matrices are of appropriate dimensions;

A 1.2 The matrices A_p, B_p, C_p, D_p are the minimal realization of the dynamical systems.

A 1.3 (A_p, B_p) and (C_p, A_p) are respectively controllable and observable pairs.

The controller $(\mathscr{C})$ is constituted of three components: a steady-state Kalman filter with observer gain L, a static state feedback controller with controller gain K and a χ^2 detector as an anomaly detector[1]. These three components can be represented as the following dynamical system:

$$\mathscr{C}: \begin{cases} \hat{x}_p[k+1] = A_p \hat{x}_p[k] + B_p u[k] + L(y_p[k] - C_p \hat{x}_p[k]) \\ u[k] = -K(\hat{x}_p[k] - x_{p,ref}) + u_{ref} \\ r[k] = y_p[k] - C_p \hat{x}_p[k] \end{cases} \tag{3.2}$$

where $\hat{x}_p \in \mathbb{R}^n$ is the estimated state, and $x_{p,ref} \in \mathbb{R}^n$, $u_{ref} \in \mathbb{R}^m$ are the reference state and control input, which are assumed to be piecewise constant.

The steady-state Kalman filter has a converged estimation error covariance matrix and a static gain which meets the following equations:

$$\begin{aligned} A_p P A_p^\top - P + \Sigma_w &= A_p P C_p^\top \left(\Sigma_v + C_p P C_p^\top\right)^{-1} C_p P A_p^\top \\ L &= \left(A_p P C_p^\top\right)\left(\Sigma_v + C_p P C_p^\top\right)^{-1} \end{aligned} \tag{3.3}$$

[1] Note that, although not the focus of this paper, we have included an anomaly detector, as our proposed methods are predominantly methods for active attack diagnosis.

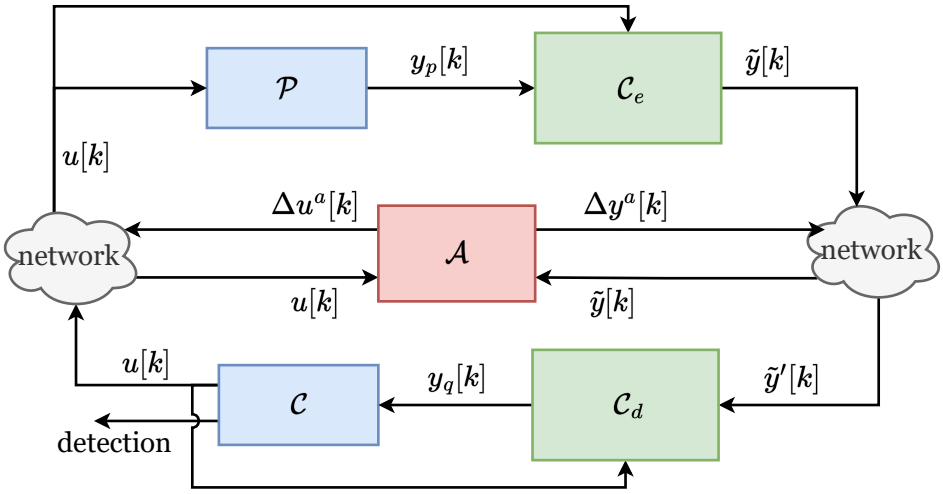


Figure 3.1: Overal System Description

To detect these attacks, the following condition based on the residual $r \in \mathbb{R}^{n_x}$ is evaluated:

$$r[k]^T \Sigma_{ad}^{-1} r[k] > \bar{r}[k] \tag{3.4}$$

where $\Sigma_{ad} = C_p P C_p^T + \Sigma_v$, and $r \in \mathbb{R}^{n_x}$ is the Kalman filter innovation. $\bar{r}[k]$ is a suitably defined threshold. If (3.4) holds, an alarm is triggered.

3.2. Attacker Capabilities

In the following chapters, we consider four kinds of attackers: replay attacker $\mathscr{A}_r$, control-signal-injection zero-dynamics attacker $\mathscr{A}_c$, sensor-signal-injection zero-dynamics attacker $\mathscr{A}_s$ and eavesdropping learning attacker $\mathscr{A}_e$. The replay attacker, the control-signal-injection zero-dynamics attacker, and the sensor-signal-injection zero-dynamics attacker have already been introduced in Section 2.2.

The eavesdropping attacker $\mathscr{A}_e$ monitors the information transmitted over the communication network between the plant and the controller, as depicted in Figure 3.1. Specifically, the eavesdropping attacker can be formalized as the following threat model:

System knowledge: The attacker knows the parameters of the plant and controller models $\{A_p, B_p, C_p, L, K\}$.

Disclousure resources: The attacker has direct access to signals y_w and u transmitted over the communication network. The set of information available to the attacker at time k can be therefore defined as:

$$\mathscr{I}_a[k] \triangleq \{A_p, B_p, C_p, L, K, u[0:k], \tilde{y}[0:k]\}. \tag{3.5}$$

Attack objective: The malicious agent attempts to reconstruct the parameters of $\mathscr{C}_e$ and $\mathscr{C}_d$. The agent aims to use the estimated parameter to design attack sequences tuned for the active detection methods to disturb the system's normal operation without being detected.

3.3. Active Detection Methods and Privacy Solutions Considered

This section will briefly recap the active detection methods and privacy solutions we will consider in the following chapters. Some of their necessary details will be covered. The methods and solutions we will recap include the system immersion method for federated learning [34], the measurement-coding method [23], the output-coding method [26] and the multiplicative watermarking method [24, 30].

3.3.1. System Immersion Method in Federated Learning

Researchers in [36] propose a privacy-preserving federated learning system using the system immersion method from nonlinear control theory. The method keeps data secrecy by mapping original data to a higher-dimensional space and deploying data operations in the mapped space.

The authors provide a linear example of immersion and inverse maps using random matrix encryption. The example linear mapping supports the SGD algorithm in the higher dimensional space and has no performance loss.

$$\begin{aligned} \text{immersion map:} \quad & \pi(s) := Gs + b^\top \\ \text{inverse map:} \quad & \pi^L(s) := Ms \end{aligned} \tag{3.6}$$

Where $G \in \mathbb{R}^{m\times n_s}, M \in \mathbb{R}^{n_s\times m}, b^\top \in \mathbb{R}^m, m > n_s$ with $MG = I$ and $b^\top$ is a time-varying random matrix which lies in the kernel space of M. The $b^\top$ component acts as one-time pad encryption, making the system-immersion method unconditionally secure while causing no performance loss compared to differential-privacy-based methods.

3.3.2. Measurement-Coding Method and Output-Coding Method

The authors in [23] propose a *measurement-coding method* to detect sensor output data deception attacks with an invertible square matrix as follows:

$$\begin{aligned} \tilde{y}[k] &= \Sigma y_p[k] \\ y'_p[k] &= \Sigma^{-1}\tilde{y}'[k] \end{aligned} \tag{3.7}$$

The sensor output data is linearly transformed by an invertible square matrix Σ and then recovered by the inverse matrix Σ^{-1} at the controller side. Because $\Sigma^{-1}\Sigma = I$, the measurement-coded method will not cause performance loss when no attacks happen and can detect data deception attacks that can induce infinite estimation error with suitable-defined Σ. However, this method cannot detect replay attacks and is vulnerable to malicious parameter identification.

An output-coding method is proposed for detecting replay attacks in [26]. The output-coding mechanism additively merges input information into the system output as follows:

$$\tilde{y}[k] = y_p[k] + C\Gamma u[k-1] = Cx[k] + C\Gamma u[k-1] \tag{3.8}$$

The authors prove this method can detect the replay attack with probability 1 with a suitable-defined Γ. However, the method is vulnerable to malicious parameter identification.

3.3.3. Multiplicative Watermarking

Proposed in [24, 30], switching multiplicative watermarking is an active technique for attack detection, whereby a watermarking generator ($\mathscr{W}$) filters y_p before its transmission over the communication network to the controller. Once received, a suitably defined watermarking remover ($\mathscr{Q}$) then processes the information, returning a signal used by the controller.

Let us start by defining the information available at the MWM generator and remover at time k as follows:

$$\begin{aligned} \mathscr{I}_w[k] &\triangleq \{y_w[0:k], y_p[0:k], x_w[0:k], \theta_w[0:k]\}, \\ \mathscr{I}_q[k] &\triangleq \{y_w[0:k], y_q[0:k], x_q[0:k], \theta_q[0:k], u[0:k]\}. \end{aligned} \tag{3.9}$$

Both the watermarking generator and remover are time-varying systems, with dynamics described as follows:

$$\begin{aligned} \mathscr{W}: &\begin{cases} x_w[k+1] = A_w(\theta_w[k])x_w[k] + B_w(\theta_w[k])y_p[k] \\ y_w[k] = C_w(\theta_w[k])x_w[k] + D_w(\theta_w[k])y_p[k] \end{cases} \\ \mathscr{Q}: &\begin{cases} x_q[k+1] = A_q\left(\theta_q[k]\right)x_q[k] + B_q\left(\theta_q[k]\right)y_w[k] \\ y_q[k] = C_q\left(\theta_q[k]\right)x_q[k] + D_q\left(\theta_q[k]\right)y_w[k] \end{cases} \\ \mathscr{F}: &\theta_w[k] = f_w(\mathscr{I}_w[k]); \quad \theta_q[k] = f_q(\mathscr{I}_q[k]) \end{aligned} \tag{3.10}$$

where $x_w, x_q \in \mathbb{R}^{n_w}$ are the watermark generator and remover states, $y_w, y_q \in \mathbb{R}^{n_y}$ their outputs, and $\theta_w[k], \theta_q[k] \in \mathbb{R}^{n_\theta}$ are their parameters at time k, with $n_\theta = (n_w + p)^2$; $f_w : \mathscr{I}_w \to \mathbb{R}^{n_\theta}$ and $f_q : \mathscr{I}_q \to \mathbb{R}^{n_\theta}$ are switching functions. We give the following definition of multiplicative watermarking pairs, following [30].

Definition 3.3.1 (Watermarking pair)**.** Two systems $(\mathscr{W}, \mathscr{Q})$, with dynamics (3.10), are said to be a watermarking pair if the following hold:

a. $\mathscr{W}$ and $\mathscr{Q}$ are stable and invertible;

b. if $\theta_w[k] = \theta_q[k]$, $y_q[k] = y_p[k]$, i.e., $\mathscr{Q} = \mathscr{W}^{-1}$. ◁

To meet Definition 3.3.1.a., the system matrices for $\mathscr{Q}$ are defined as:

$$\begin{aligned} &D_q(\theta) = D_w(\theta)^{-1}; A_q(\theta) = A_w(\theta) - B_w(\theta)D_w(\theta)^{-1}C_w(\theta); \\ &B_q(\theta) = B_w(\theta)D_q(\theta); \quad C_q(\theta) = -D_q(\theta)C_w(\theta). \end{aligned} \tag{3.11}$$

where $\theta = \theta_w[k] = \theta_q[k]$.

3.4. Problem Formulation

In Section 1.4 Problem 1, we introduced our general research question, i.e. upgrading existing or proposing new active detection methods to address the malicious parameter identification. We then introduced two research problems that we will focus on in this thesis, Problem 2 and Problem 3. They focus on a combination of the system immersion method [36], the output-coding method [26] and the multiplicative watermarking method [30] to address Problem 1.

This section will formulate our research question more specifically based on Problem 2 and Problem 3.

3.4.1. Problem Formulation: System Immersion Coding Method

Given the scenario presented in the previous subsections and Problem 2, the problems we address in chapter 4 can be formalized as the following three sub-questions.

Problem 2.1. Try to combine the system immersion method [36] and the output-coding method [26] to propose a new active detection method. ◁

Remark 1. We call the method proposed for Problem 2.1 *system immersion coding method.* ◁

The following question is related to the main question of this thesis: defending against malicious parameter identification.

Problem 2.2. Given a cyber-physical system(3.1)-(3.2), equipped with a system immersion coding method, analyze the performance of the system immersion coding method on defending against malicious parameter identification. ◁

Moreover, as an active detection method, the proposed approach should also answer the following research question:

Problem 2.3. Given a cyber-physical system(3.1)-(3.2), equipped with system immersion coding method components, design the parameters such that the defender can use the system immersion coding method to detect reply attack, control-signal-injection zero-dynamics attack and sensor-signal-injection zero-dynamics attack ◁

3.4.2. Problem Formulation: Hybrid Multiplicative Watermarking Method

Given the scenario presented in the previous subsections and Problem 1, the problems we address in Chapter 5 and Chapter 6 can be formalized as the following three sub-questions.

Problem 3.1. Try to combine the output-coding method [26] and the multiplicative watermarking method [30] to propose a new multiplicative watermarking structure. ◁

Remark 2. We call the method proposed for Problem 3.1 as *hybrid multiplicative watermarking method.* ◁

The switching rules represented by f_w and f_q affect the difficulty for a malicious agent to identify the hybrid multiplicative watermarking parameters. The switching rules we are to design should meet several requirements:

- **R1** **Fast Switching**: The mode should switch rapidly;
- **R2** **Randomness**: The switching sequence should not be known in advance;
- **R3** **Synchronization**: $\mathscr{W}$ and $\mathscr{Q}$ should have synchronized modes, i.e., the mode should be chosen based on common information of $\mathscr{I}_w[k]$ and $\mathscr{I}_q[k]$.

Remark 3. Requirement **R1** states that an objective of our solution is that it be fast switching. We set this requirement to avoid any design strategy that includes a minimum dwell time, as it has been shown to be beneficial for parameter identification, as is pointed out in Section 5.3. ◁

The following question is related to the main question of this thesis: defending against malicious parameter identification.

Problem 3.2. Given a cyber-physical system (3.1)-(3.2), equipped with a hybrid multiplicative watermarking scheme (3.10), design the time-varying parameters θ_w, θ_q such that:

a. $(\mathscr{W}, \mathscr{Q})$ is a watermarking pair, as per Definition 3.3.1;

b. the CPS maintains closed-loop stability under switching;

c. an attacker with the information set $\mathscr{I}_a$ and capabilities defined in Section 3.2 cannot exactly reconstruct $\theta_w[k], \theta_q[k]$, for all $k \geq K_{id}^a$, i.e. the time and data complexity to exactly identify the parameters can be arbitrarily large. In this thesis, it relates to meeting the requirements **R1**-**R3**. ◁

Moreover, as an active detection method, the proposed approach should also answer the following research question:

Problem 3.3. Given a cyber-physical system (3.1)-(3.2), equipped with a hybrid multiplicative watermarking scheme (3.10), design the parameters such that the defender can use the hybrid multiplicative watermarking method to detect different attacks. ◁

3.5. Testbench Overview

In this section, we will introduce two testbenches we will use in the later chapters. The model of these testbenches will be given. We will show the system's normal operation when the attacks are absent and the attacker performance of different attackers on these test benches.

3.5.1. Testbench 1

The first testbench is the linearized quadruple-tank water system used in [27]. This testbench is vulnerable to the replay attack and the control-signal-injection zero-dynamics attack. The linearized model is as follows:

$$A_p = \begin{bmatrix} 0.975 & 0 & 0.042 & 0 \\ 0 & 0.977 & 0 & 0.044 \\ 0 & 0 & 0.958 & 0 \\ 0 & 0 & 0 & 0.956 \end{bmatrix}, \quad B_p = \begin{bmatrix} 0.0515 & 0.0016 \\ 0.0019 & 0.0447 \\ 0 & 0.0737 \\ 0.0850 & 0 \end{bmatrix}, \quad C_p = \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \end{bmatrix}, \quad D_p = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{3.12}$$

The noise parameter, the linearized operating points and the controller parameters are as follows:

$$\begin{aligned} &\mu_w = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top, \quad \mu_v = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top, \\ &\Sigma_w = 10^{-3} I_4, \quad \Sigma_v = 10^{-1} I_2 \\ &x_{\text{ref}} = [5, 5, 2.044, 1.399]^\top, \quad u_{\text{ref}} = [0.724, 1.165]^\top, \\ &K = \begin{bmatrix} -3.0993 & -4.0721 & 2.0528 & -2.8417 \\ -3.9353 & -3.3330 & -2.8461 & 1.9997 \end{bmatrix} \end{aligned} \tag{3.13}$$

When the attack is absent, the system's performance is shown in Figure 3.2. The simulation length is 1000 steps. We can see from Figure 3.2 (d) that under normal operation, no alarming is triggered after timestamp 1. An alarming is triggered at timestamp 0 because in the simulation, we set the initial state of the observer different from the true system state [2].

The operation result of the system under the replay attack is shown in Figure 3.3. In the replay attack scenario, the simulation length is 1000 steps. The replay attacker starts to replay the sensor signal at $k_a = 800$, and the replayed signal is recorded from $k_1 = 600$ to $k_2 = 800$. Figure 3.3(a) and Figure 3.3(b) show that the system state and system output is not affected heavily because of the replay attack. We can see from Figure 3.3 (d) that no alarm is triggered during the attack period.

[2] The same phenomenon can be seen in all later simulations; we will not recap it later

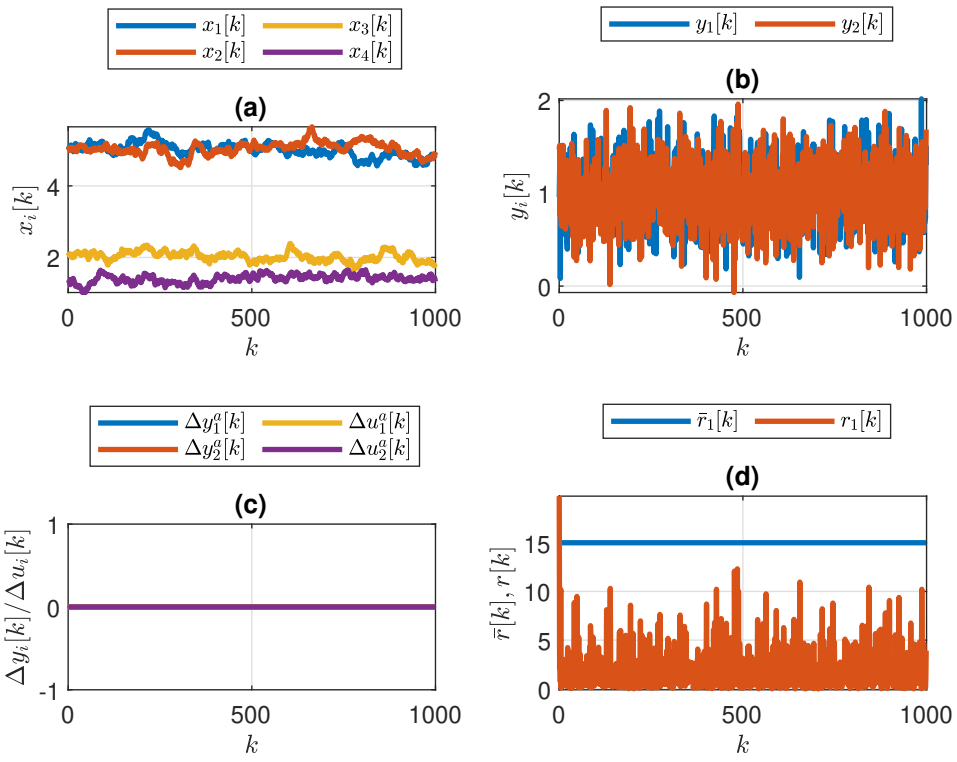


Figure 3.2: Testbench 1: without attack. (a) the plant state. (b) the plant output. (c) the attackers' injected sequence. (d) the residual value.

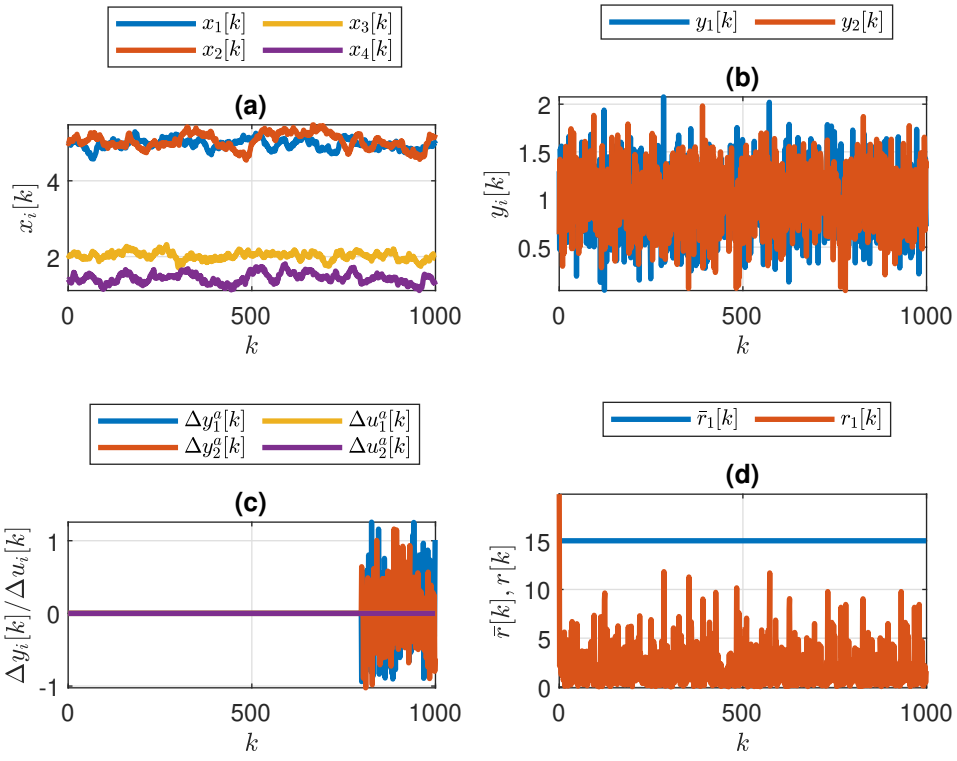


Figure 3.3: Testbench 1 under replay attack. (a) the plant state. (b) the plant output. (c) the attackers' injected sequence. (d) the residual value.

The operation result of the system under the control-signal-injection zero-dynamics attack is shown in Figure 3.4. In the replay attack scenario, the simulation length is 1000 steps, and the parameter of the attack sequence is $v = 1.03, \quad g = 10^{-2} \times [-0.26, 0.3]^\top$. The attacker starts to inject a malicious signal from timestamp $k_a = 800$, and that attack last until the end of the simulation. Figure 3.4(a) and Figure 3.4(b) show that the system state is affected heavily because of the attack. However, we can see from Figure 3.4(d) that no alarm is triggered during the attack period. It is because the control-signal-injection zero-dynamics only affect the system state but not the system output.

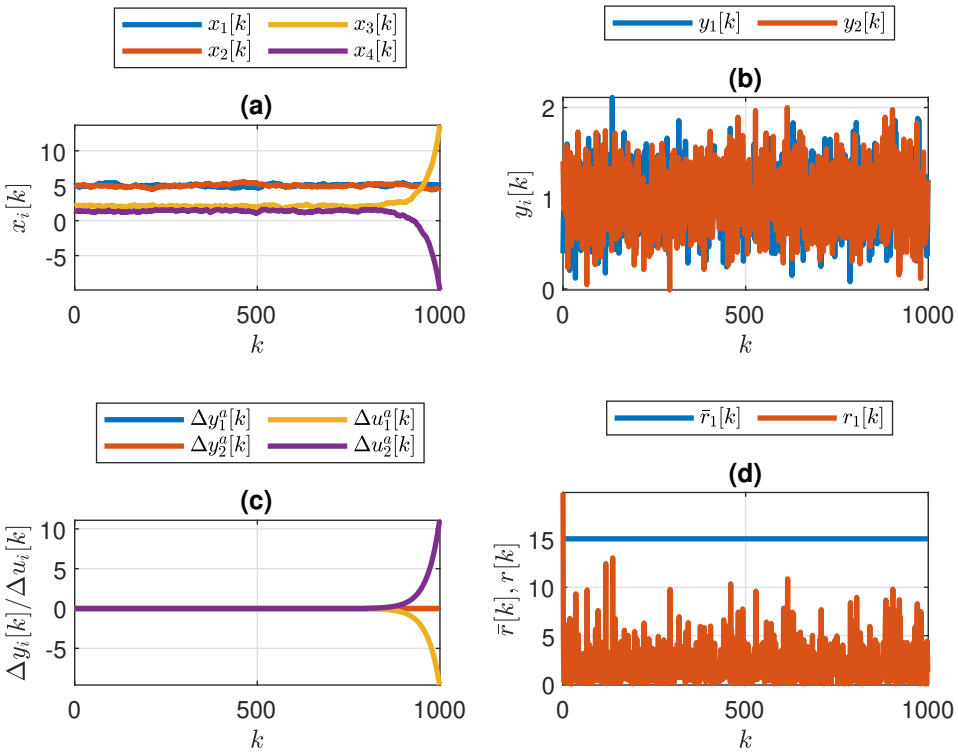


Figure 3.4: Testbench 1: under control-signal-injection zero-dynamics attack. (a) the plant state. (b) the plant output. (c) the attackers' injected sequence. (d) the residual value.

3.5.2. Testbench 2

The second testbench is the testbench model used in [30]. This testbench is vulnerable to the replay attack and the sensor-signal-injection zero-dynamics attack. The model is as follows:

$$A_p = \begin{bmatrix} 1 & 0.1 \\ 0.035 & 0.99 \end{bmatrix}, \quad B_p = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C_p = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad D_p = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{3.14}$$

The noise parameter, the reference points and the controller parameters are as follows:

$$\begin{aligned} \mu_w &= [0 \quad 0 \quad 0 \quad 0]^\top, \quad \mu_v = [0 \quad 0]^\top, \\ \Sigma_w &= 10^{-3} I_4, \quad \Sigma_v = 10^{-1} I_2 \\ x_{\text{ref}} &= [1.5, 0]^\top, \quad u_{\text{ref}} = -0.0525, \\ K &= [-0.5350 \quad -0.5900] \end{aligned} \tag{3.15}$$

When the attack is absent, the system's performance is shown in Figure 3.5. The simulation length is 1000 steps. We can see from Figure 3.5(d) that under normal operation, no alarming is triggered after timestamp 1.

The operation result of the system under the replay attack is shown in Figure 3.6. In the replay attack scenario, the simulation length is 1000 steps. The replay attacker starts to replay the sensor signal at $k_a = 800$, and the replayed signal is recorded from $k_1 = 600$ to $k_2 = 800$. Figure 3.6(a) and Figure 3.6(b) show that the

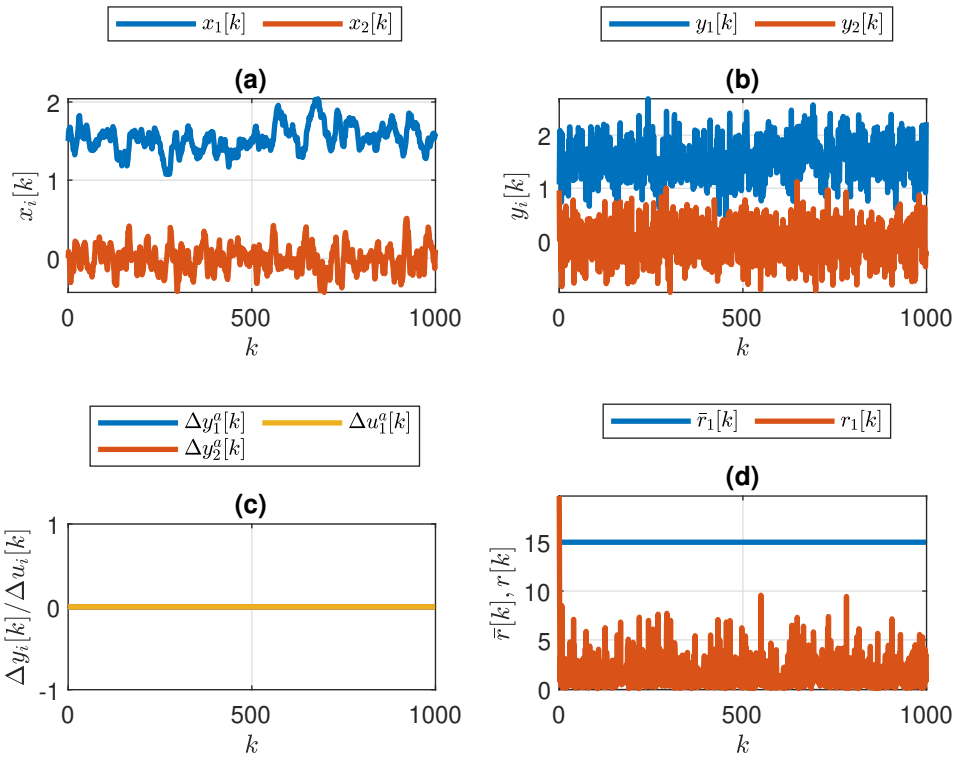


Figure 3.5: Testbench 2: without attack. (a) the plant state. (b) the plant output. (c) the attackers' injected sequence. (d) the residual value.

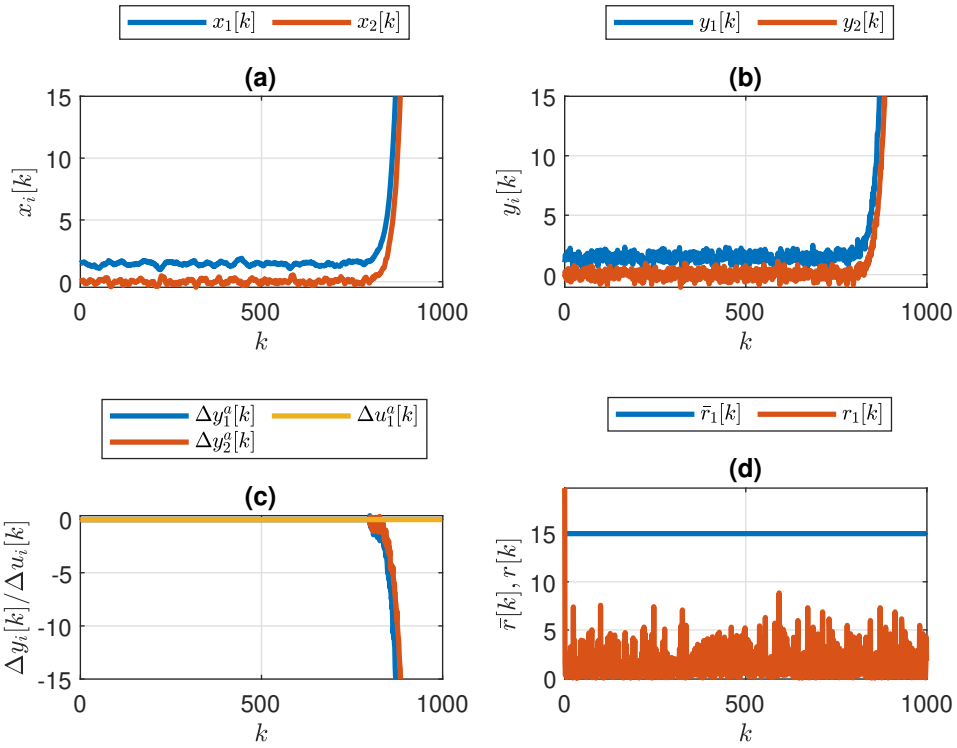


Figure 3.6: Testbench 2: replay attack. (a) the plant state. (b) the plant output. (c) the attackeWhendual value.

system state and system output is affected heavily because of the replay attack. However, we can see from Figure 3.6 (d) that no alarm is triggered during the attack period.

The operation result of the system under the sensor-signal-injection zero-dynamics attack is shown in Figure 3.7. In the replay attack scenario, the simulation length is 1000 steps, and the parameter of the attack sequence is $v = 1.0544$, $g = 10^{-2} \times [0.6212, 0.3378]^\top$. The attacker starts to inject a malicious signal from timestamp $k_a = 800$, and that attack last until the end of the simulation. Figure 3.7(a) and Figure 3.7(b) show that the system state and system output is affected heavily because of the sensor-signal-injection zero-dynamics attack. However, we can see from Figure 3.7 (d) that no alarm is triggered during the attack period.

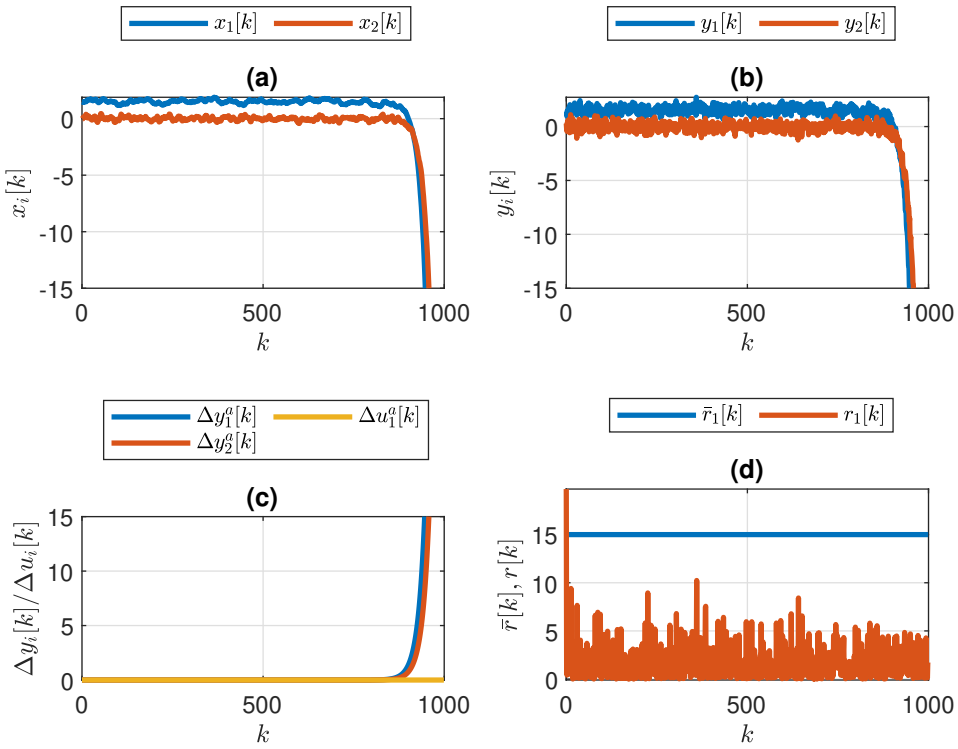


Figure 3.7: Testbench 2: sensor-signal-injection zero-dynamics attack. (a) the plant state. (b) the plant output. (c) the attackers' injected sequence. (d) the residual value.

4

System Immersion Coding Method

In Section 3.4, we specified our research problems of this chapter, i.e. Problem 2.1, Problem 2.2, and Problem 2.3. The specified problems involve proposing a new active detection method based on the system immersion method, and the output-coding method and studying its performance in defending against malicious parameter identification and detecting malicious attacks.

In this chapter, we propose a new active detection method named *system immersion coding method*. We will provide guidelines for design parameters of the system immersion coding method to help detect malicious attacks and study its performance in defending against malicious parameter identification.

Figure 4.1 shows the considered CPS structure of this chapter. Compared to the structure in Chapter 3, the active detection part is specified to be a system immersion coding pair $(\mathscr{E},\mathscr{D})$.

In Section 4.1, we will see that the system immersion coding method maps the original plant-to-controller signal $y_p[k] \in \mathbb{R}^{n_y}$ to a signal $s[k] \in \mathbb{R}^{n_s}$ with a higher dimension, i.e. $n_s > n_y$. So, for the sensor-signal-injection zero-dynamics attack, we assume the attacker firstly guesses a matrix $G' \in \mathbb{R}^{n_s \times n_y}$ and then injects malicious signal $\Delta y^a[k] = G' v^k g$, where v and g are calculated by equation (2.5).

We will also see in Section 4.1 that the system immersion coding method has two important parameters $G \in \mathbb{R}^{n_s \times n_y}$ and $H \in \mathbb{R}^{n_y \times n_u}$. The eavesdropping attacker $\mathscr{A}_e$ will then be defined as the following threat model:

System knowledge: The attacker knows the parameters of the plant and controller models $\{A_p, B_p, C_p, L, K\}$.

Disclousure resources: The attacker has direct access to signals s and u transmitted over the communication network. The set of information available to the attacker at time k can be therefore defined as:

$$\mathscr{I}_a[k] \triangleq \{A_p, B_p, C_p, L, K, u[0:k], s[0:k]\}. \tag{4.1}$$

Attack objective: The malicious agent attempts to reconstruct the multiplicative watermarking parameters $G \in \mathbb{R}^{n_s \times n_y}$ and $H \in \mathbb{R}^{n_y \times n_u}$.

4.1. System Immersion Coding for Active Detection Method

From Section 3.3.1 and Section 3.3.2, we know that:

- The system immersion method can protect data privacy and introduce an arbitrary magnitude of noise without causing performance loss.
- The output coding method additively merges the $u[k]$ information to detect replay attacks, but it suffers from malicious parameter identification. From [27], A possible way to detect the control-signal-injection zero-dynamics attack is the merging of $u[k]$.
- The mathematical operations of the system immersion method and the output coding method involve matrix multiplication and addition, which makes them easy to be combined.

Therefore, a possible way to solve malicious parameter identification is to combine the output coding and system immersion methods. Intuitively, a large signal-noise ratio(SNR) can disturb the performance of system identification methods. We hope that the method of protecting data privacy can also work when used to protect parameter privacy.

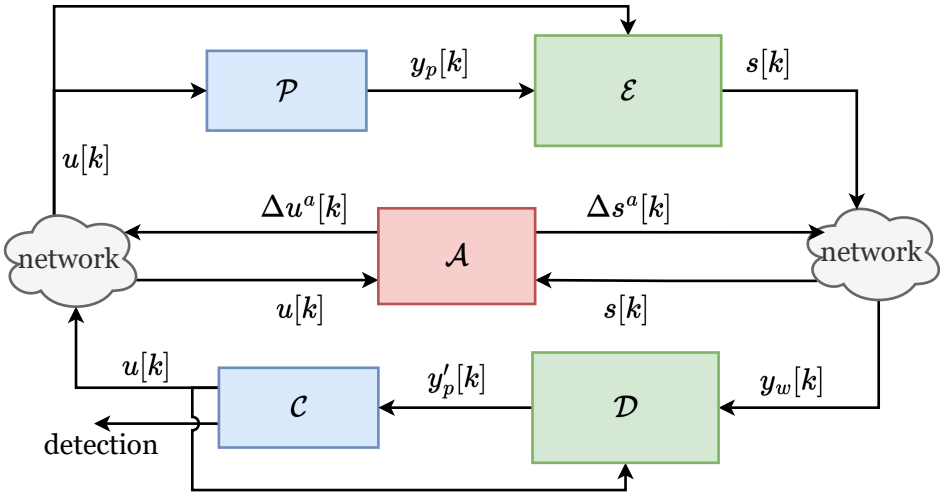


Figure 4.1: Overal System Description

Our proposed strategy consists of an encoding module ($\mathscr{E}$) and a decoding module ($\mathscr{D}$). The $\mathscr{E}$ encodes the unaltered measurement signal y_p into s and the $\mathscr{D}$ then decodes $s'[k]$ to define $y'_p[k]$

$$\begin{aligned} \mathscr{E} &: s[k] = G\big(y_p[k] + Hu'[k]\big) + NR[k]. \\ \mathscr{D} &: y'_p[k] = Ms'[k] - Hu[k]. \end{aligned} \tag{4.2}$$

In (4.2) $G \in \mathbb{R}^{n_s \times n_y}$ with $n_s > n_y$, $M \in \mathbb{R}^{n_y \times n_s}$ are designed such that $MG = I$, $H \in \mathbb{R}^{n_y \times n_u}$, each column of N, N_i satisfy $N_i \in \ker(M)$, and R is a random number generated by a random number generator. We expect that G, M, H, N are unknown to the attacker, and the random number $R[k]$ contains some information unknown to the attacker. We call the $NR[k]$ part "artificial noise".

Proposition 4.1.1. *Given the matrix definition, if there is no attack in the system, then $y'_p[k] = y_p[k]$ and the control performance is unaffected by the system immersion coding method.* □

Proof. Assuming no attack sequence is injected into the system, we have the following:

$$s'[k] = s[k] \quad u'[k] = u[k] \tag{4.3}$$

From the definition of M, G, N, we have $MG = I$ and $MN = 0$.Then we can compute $y_p[k]$:

$$\begin{aligned} y'_p[k] &= Ms'[k] - Hu[k] \\ &= MG(y_p[k] + Hu'[k]) + MNR[k] - Hu[k] \\ &= y_p[k] + Hu'[k] - Hu[k] \\ &= y_p[k] \end{aligned} \tag{4.4}$$

Because of $y'_p[k] = y_p[[k]$, the controller can obtain the actual system output, which means the system immersion coding does not affect the control performance when no attack happens. ■

4.1.1. Detectability Design of System Immersion Coding

In this section, we propose and prove several theorems to provide guidelines for designing the system immersion coding method's parameters. These theorems focus on helping the system immersion coding method detect different attacks, including the replay attack, the control-signal-injection zero-dynamics attack and the sensor-signal-injection zero-dynamics attack. Which mainly addresses Problem 2.3.

Replay Attack

Theorem 1. *If the matrix H is designed to make matrix $\Phi_H = A_p - B_pK - LC_p - LHK$ has at least one unstable eigenvalue and the pair $(C_p + HK, \Phi_H)$ is made observable, then the replay attack will be detected.* □

Proof. When the system is under replay attack, the control signals the plant received $u'[k]$, the measurement signals the controller received $s'[k]$ and the controller's decoded value $y'_p[k]$ will be:

$$\begin{aligned} s'[k] &= s'[k-\Delta k] \\ &= G\left(y_p[k-\Delta k] + Hu[k-\Delta k]\right) + NR[k-\Delta k] \\ u'[k] &= u[k] \\ y'_p[k] &= y_p[k-\Delta k] + H\left(u[k-\Delta k] - u[k-1]\right) \end{aligned} \tag{4.5}$$

Define $\Delta\hat{x}_p[k] = \hat{x}_p[k-\Delta k] - \hat{x}_p[k]$, $\Delta u[k] = u[k-\Delta k] - u[k]$, then the residual value $r[k]$ and the observer's estimation of the system state $\hat{x}[k+1]$ under the replay attack can be expressed as:

$$\begin{aligned} r[k] &= y'_p[k] - \hat{y}_p[k] \\ &= \left(y_p[k-\Delta k] - \hat{y}_p[k-\Delta k]\right) + \left(\hat{y}_p[k-\Delta k] - \hat{y}_p[k]\right) + H\left(u[k-\Delta k] - u[k]\right) \\ &= r[k-\Delta k] + \left(C_p\hat{x}_p[k-\Delta k] - C_p\hat{x}[k]\right) + H\left(u[k-\Delta k] - u[k]\right) \\ &= r[k-\Delta k] + (C_p + HK)\Delta\hat{x}_p[k] \\ \hat{x}_p[k+1] &= A_p\hat{x}_p[k] + B_pu'[k] + L\left(y'_p[k] - \hat{y}_p[k]\right) \\ &= A_p\hat{x}_p[k] + B_pK\hat{x}_p[k] + L\left(y_p[k-\Delta k] + H\left(u[k-\Delta k] - u[k]\right) - \hat{y}_p[k]\right) \\ &= (A_p - B_pK - LC_p)\hat{x}_p[k] + Ly_p[k-\Delta k] + LH\Delta u[k] \\ &= (A_p - B_pK - LC_p + LHK)\hat{x}_p[k] + Ly[k-\Delta k] + LHK\Delta\hat{x}_p[k] \end{aligned} \tag{4.6}$$

The $\hat{x}_p[k-\Delta k+1]$ can be expressed as:

$$\begin{aligned} \hat{x}_p[k-\Delta k+1] &= A_p\hat{x}_p[k-\Delta k] + B_pu'[k-\Delta k] + L\left(y'_p[k-\Delta k] - \hat{y}_p[k-\Delta k]\right) \\ &= (A_p - B_pK - LC_p)\hat{x}_p[k-\Delta k] + Ly_p[k-\Delta k] \end{aligned} \tag{4.7}$$

Then the $\Delta\hat{x}_p[k+1]$ can be represented as:

$$\begin{aligned} \Delta\hat{x}_p[k+1] &= (A_p - B_pK - LC_p)\hat{x}_p[k-\Delta k] + Ly_p[k-\Delta k] \\ &\quad - (A_p - B_pK - LC_p)\hat{x}_p[k] - Ly_p[k-\Delta k] - LHK\Delta\hat{x}_p[k] \\ &= \underbrace{(A_p - B_pK - LC_p - LHK)}_{\Phi_H}\Delta\hat{x}_p[k] \\ &= \Phi_H^{k+1}\Delta\hat{x}_p[0] \end{aligned} \tag{4.8}$$

Then the residual can be rewritten as:

$$\begin{aligned} r[k] &= r[k-\Delta k] + (C_p + HK)\Delta\hat{x}_p[k] \\ &= r[k-\Delta k] + (C_p + HK)\Phi_H^{k+1}\Delta\hat{x}_p[0] \end{aligned} \tag{4.9}$$

Define $C_H = (C_p + HK)$ If the matrix $C_H \neq 0$ and Φ_H has at least one unstable eigenvalue, then the part of $C_H\Phi_H^{k+1}\Delta\hat{x}[0]$ will diverge to infinite. This will lead to an infinite residual value and trigger the alarm. ■

Control-Signal-Injection Zero-Dynamics Attack

Theorem 2. *The original control-signal-injection zero-dynamics attack, i.e. with the g value meeting equation (2.3), will keep being a zero-dynamics attack if and only if there exists $x_1 \in \mathbb{R}^{n_x}$, such that:*

$$\begin{bmatrix} \nu I - A_p & -B_p \\ C_p & H \end{bmatrix} \begin{bmatrix} x_1 \\ g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.10}$$

□

Proof. When a control-signal-injection zero-dynamics attack happens, the attacker tries to inject signal $\Delta u^a[k]$ into the system. Under the control-signal-injection zero-dynamics attack, the encoded signal $s[k]$ and the

decoded signal $y'_p[k]$ will become:

$$\begin{aligned} s[k] &= G\left(y_p[k] + Hu'[k]\right) + NR[k] \\ &= G\left(y_p[k] + Hu[k] + H\Delta u^a[k]\right) + NR[k] \\ y'_p[k] &= Ms'[k] - Hu[k] \\ &= y_p[k] + H\Delta u^a[k] \end{aligned} \tag{4.11}$$

The dynamics of the plant state $x_p[k+1]$ then become:

$$\begin{aligned} x_p[k+1] &= A_p x_p[k] + B_p u[k] + w[k] \\ &= A_p x_p[k] + B_p u[k] + B_p \Delta u^a[k] + w[k] \end{aligned} \tag{4.12}$$

Define estimation error $e[k] = x_p[k] - \hat{x}_p[k]$, then the dynamics of observer's estimation of the plant state $\hat{x}_p[k+1]$ will become:

$$\begin{aligned} \hat{x}_p[k+1] &= A_p \hat{x}_p[k] + B_p u[k] + L\left(y'_p[k] - \hat{y}_p[k]\right) \\ &= A_p \hat{x}_p[k] + B_p u[k] + L\left(y_p[k] + H\Delta u^a[k] - \hat{y}_p[k]\right) \\ &= (A_p - B_p K)\hat{x}_p[k] + L\left(C_p x_p[k] + v[k] - C_p \hat{x}_p[k] + H\Delta u^a[k]\right) \\ &= (A_p - B_p K)\hat{x}_p[k] + LC_p e[k] + LH\Delta u^a[k] + Lv[k] \end{aligned} \tag{4.13}$$

The estimation error $e[k+1]$ can be expanded as follows:

$$\begin{aligned} e[k+1] &= A_p x_p[k] + B_p u[k] + B_p \Delta u^a[k] + w[k] \\ &\quad - (A_p - B_p K)\hat{x}_p[k] - LC_p e[k] - LH\Delta u^a[k] - Lv[k] \\ &= A_p(x_p[k] - \hat{x}_p[k]) - LC_p e[k] + (B_p - LH)\Delta u^a[k] - Lv[k] + w[k] \\ &= (A_p - LC_p)e[k] + (B_p - LH)\Delta u^a[k] + w[k] - Lv[k] \end{aligned} \tag{4.14}$$

The residual value $r[k]$ can be expanded as follows:

$$\begin{aligned} r[k] &= y'_p[k] - \hat{y}_p[k] \\ &= y_p[k] + H\Delta u^a[k] - C_p \hat{x}_p[k] \\ &= C_p x_p[k] - C_p \hat{x}_p[k] + H\Delta u^a[k] \\ &= C_p e[k] + H\Delta u^a[k] \end{aligned} \tag{4.15}$$

Then the joint dynamics of $e[k]$ and $r[k]$ will become:

$$\begin{aligned} e[k+1] &= (A_p - LC_p)e[k] + (B_p - LH)\Delta u^a[k] + w[k] - Lv[k] \\ r[k] &= C_p e[k] + H\Delta u^a[k] \end{aligned} \tag{4.16}$$

If the attack sequence $\Delta u^a[k] = v^k g$ still is a malicious zero-dynamics attack sequence, it should meet the following condition:

$$\begin{bmatrix} vI - (A_p - LC_p) & LH - B_p \\ C_p & H \end{bmatrix} \begin{bmatrix} x_1 \\ g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \exists x_1 \in \mathbb{R}^{n_x} \tag{4.17}$$

Equation (4.17) equals to:

$$\begin{bmatrix} vI - A_p & -B_p \\ C_p & H \end{bmatrix} \begin{bmatrix} x_1 \\ g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \exists x_1 \in \mathbb{R}^{n_x} \tag{4.18}$$

This is the same as equation (4.10) and finish the proof.

■

Proposition 4.1.2. *If $g \notin \ker(H)$, equation 4.10 can only be met if:*

1. *v is both a transmission zero of the system and an eigenvalue of A_p.*
2. *Assume $\mathbb{E}_v$ is the eigenspace of A_p that corresponding the eigenvalue of v. Then there exists $x \in \mathbb{E}_v$, such that $Hg = C_p x$.* □

Proof. We will discuss the equation (4.10) in two cases: (1) $x_1 - x_0 = 0$; (2) $x_1 - x_0 \neq 0$.

1. If $x_1 - x_0 = 0$, then from equations (2.3) and (4.10), we have:

$$C_p x_1 + Hg = C_p x_0 + Hg = Hg \tag{4.19}$$

   which means if the designed H make $g \notin \ker(H)$, then the attack will not be stealthy;

2. If $x_1 - x_0 \neq 0$, then Equation (2.3) and equation (4.10) require:

$$(vI - A_p)x_1 + B_p g = (vI - A_p)x_0 + B_p g \Leftrightarrow (vI - A_p)(x_1 - x_0) = 0 \tag{4.20}$$

   Define $\Delta x = x_1 - x_0$, equation (4.20) means $\Delta x \in \ker(vI - A_p)$, which is possible only when the v is an eigenvalue of A_p. The equation (4.10) also requires $C_p \Delta x + Hg = 0$, which is possible only when there exists $\Delta x \in \mathbb{E}_v$, such that $Hg = C_p \Delta x$.

■

Sensor-Signal-Injection Zero-Dynamics Attack

Theorem 3. *The original sensor-signal-injection zero-dynamics attack, i.e. with the g value meeting equation (2.5), will keep being a zero-dynamics attack if and only if there exists $x_1 \in \mathbb{R}^{n_x}$, such that:*

$$\begin{bmatrix} vI - A_p & 0 \\ C_p & MG' \end{bmatrix} \begin{bmatrix} x_1 \\ g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.21}$$

□

Proof. When a sensor-signal-injection zero-dynamics attack happens, the attacker tries to inject signal $\Delta y^a[k]$ into the system. Under the sensor-signal-injection zero-dynamics attack, the encoded signal $s[k]$ and the decoded signal $y'_p[k]$ follow:

$$\begin{aligned} s[k] &= G\big(y_p[k] + Hu[k]\big) + NR[k] \\ s'[k] &= s[k] + \Delta y^a[k] \\ y'_p[k] &= Ms'[k] - Hu[k] \\ &= y_p[k] + M\Delta y^a[k] \end{aligned} \tag{4.22}$$

Define estimation error $e[k] = x_p[k] - \hat{x}_p[k]$, then the dynamics of observer's estimation of the plant state $\hat{x}_p[k+1]$ will become:

$$\begin{aligned} \hat{x}_p[k+1] &= A_p \hat{x}_p[k] + B_p u[k] + L\Big(y'_p[k] - \hat{y}_p[k]\Big) \\ &= A_p \hat{x}_p[k] + B_p u[k] + L\big(y_p[k] + M\Delta y^a[k] - \hat{y}_p[k]\big) \\ &= (A_p - B_p K)\hat{x}_p[k] + L\big(C_p x_p[k] + v[k] - C_p \hat{x}_p[k] + M\Delta y^a[k]\big) \\ &= (A_p - B_p K)\hat{x}_p[k] + LC_p e[k] + LM\Delta y^a[k] + Lv[k] \end{aligned} \tag{4.23}$$

The estimation error $e[k]$ can be expanded as follows:

$$\begin{aligned} e[k+1] &= A_p x_p[k] + B_p u[k] + w[k] - (A_p - B_p K)\hat{x}_p[k] - LC_p e[k] - LM\Delta y^a[k] - Lv[k] \\ &= A_p(x_p[k] - \hat{x}_p[k]) - LC_p e[k] - LM\Delta y^a[k] - Lv[k] + w[k] \\ &= (A_p - LC_p)e[k] - LMG' v^k g + w[k] - Lv[k] \end{aligned} \tag{4.24}$$

The residual value $r[k]$ can be expanded as follows:

$$\begin{aligned} r[k] &= y'_p[k] - \hat{y}_p[k] \\ &= y_p[k] + M\Delta y^a[k] - C_p \hat{x}_p[k] \\ &= C_p x_p[k] - C_p \hat{x}_p[k] + M\Delta y^a[k] \\ &= C_p e[k] + MG' v^k g \end{aligned} \tag{4.25}$$

Then the joint dynamics of $e[k]$ and $r[k]$ will become:

$$\begin{aligned} e[k+1] &= (A_p - LC_p)e[k] - LMG'v^k g + w[k] - Lv[k] \\ r[k] &= C_p e[k] + MG'v^k g \end{aligned} \tag{4.26}$$

If the attack sequence $\Delta y^a[k] = G'v^k g$ still wants to be a malicious zero-dynamics attack sequence, it should meet the following condition:

$$\begin{bmatrix} vI - (A_p - LC_p) & LMG' \\ C_p & MG' \end{bmatrix} \begin{bmatrix} x_1 \\ g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \exists x_1 \in \mathbb{R}^{n_x} \tag{4.27}$$

Equation (4.27) equals to:

$$\begin{bmatrix} vI - A_p & 0 \\ C_p & MG' \end{bmatrix} \begin{bmatrix} x_1 \\ g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.28}$$

This is the same as equation (4.21) and finished the proof. ■

4.2. Identification Resistance of System Immersion Coding Method

In [34], the author analyzes the privacy performance utilizing the conception of *unconditional security*. In the system immersion method, because the $b^\top$ component plays a role as one-time pad encryption, it can theoretically achieve perfect secrecy.

In this section, we will show that with the system immersion coding method proposed in section 4.1, the defender can disturb the attacker's estimation of the matrices G, H. However, this section will also show that the biased estimation is not enough for the security scenario because an attacker with a biased estimation of parameters can still inject a stealthy malicious attack.

4.2.1. Identification of the System Immersion Parameter

In the system immersion coding method, because of $N \in \ker(M)$, the magnitude of $NR[k]$ can be arbitrarily large and will not cause any performance loss. Intuitively speaking, an arbitrarily large $NR[k]$ can cause an arbitrarily large signal-noise ratio (SNR), which can disturb the estimation of the system parameter G, H.

Assume the attacker starts to collect data at timestamp 0. Consider that at timestamp k the attacker tries to use two ways of estimating the parameters G, H based on the data $u[0:k], s[0:k]$ the attacker collected.

1. *Least Square Method:* At each timestamp $k_a \leq k$, the attacker can firstly approximately estimate a system output $\hat{y}_p^a[k_a]$ based on $\mathscr{I}_a[0:k_a]$. Then at timestamp k, it can estimate the parameters G, H based on collected $u[0:k], s[0:k]$ and estimated $\hat{y}_p^a[0:k]$ using a least square method.
2. *System Identification Method*: The attacker can regard the plant and the system immersion coding components as a single dynamic system. It can then try to find the parameter of the entire dynamic system.

Remark 4. One of the possible way to obtain $\hat{y}_p^a[k_a]$ is execute an open-loop estimation as follows:

$$\begin{aligned} \hat{x}_p^a[k+1] &= A_p \hat{x}_p^a[k] + B_p u[k] \\ \hat{y}_p^a[k] &= C_p \hat{x}_p^a[k] \end{aligned} \tag{4.29}$$

There may be other ways to obtain $\hat{y}_p^a[k_a]$. In [66], the researchers study how to use plant output $y_p[k]$ to estimate the controller's and the anomaly detector's states. Probably a similar way can be used to estimate plant state and output based on $u[k]$. However, finding the optimal approach to obtain $\hat{y}_p^a[k_a]$ is not the central research question of this thesis work. ◁

Remark 5. Except for the two approaches mentioned above, there may be various other methods. However, finding the optimal choice is not the central research question of this thesis work. ◁

Standard System Identification Method

The attacker can regard the plant and the system immersion coding components as an augmented dynamical system. The joint dynamics will become:

$$\begin{aligned} x_p[k+1] &= A_p x_p[k] + B_p u[k] + w[k] \\ s[k] &= \underbrace{GC_p}_{C_{\text{aug}}} x_p[k] + \underbrace{GH}_{D_{\text{aug}}} u[k] + Gv[k] + NR[k] \end{aligned} \tag{4.30}$$

Then the attacker can deploy standard system identification methods like a prediction-error method or subspace identification method to identify the corresponding matrices $C_{\text{aug}}, D_{\text{aug}}$ and then find the matrices of G and M.

If we assume that the noise $w[k], v[k], NR[k]$ follow different zero-mean Gaussian distributions and assume the input signal $u[k]$ meets the persistence excitation condition, from system identification perspective, the attacker's estimation will be a consistent estimation with methods like prediction error method when the samples size is large enough.

Remark 6. It is realistic to assume that the persistence excitation condition is met because of the existence of Gaussian noise $w[k]$ and $v[k]$. Under the existence of $w[k]$ and $v[k]$, the $u[k]$ will also be a random process, which meets the persistence excitation condition. ◁

Least Square Method with Output Estimation

Define the $\hat{y}_p^a[k]$ as the attacker's estimation of the plant output at timestamp k and define $\Delta\hat{y}_p^a[k] = y_p[k] - \hat{y}_p^a[k]$, the relation between $u[0:k]$, $\hat{y}_p^a[0:k]$, $y_p[0:k]$, and $s[0:k]$ is as follows:

$$\begin{aligned} &\underbrace{\begin{bmatrix} G & GH \end{bmatrix}}_{G_{\text{aug}}} \underbrace{\begin{bmatrix} y_p[0] & \cdots & y_p[k] \\ u[0] & \cdots & u[k] \end{bmatrix}}_{\bar{D}} + \underbrace{\begin{bmatrix} NR[0] & \cdots & NR[k] \end{bmatrix}}_{\bar{N}} = \underbrace{\begin{bmatrix} s[0] & \cdots & s[k] \end{bmatrix}}_{\bar{S}} \\ \Rightarrow \quad & G_{\text{aug}} \underbrace{\begin{bmatrix} \hat{y}_p^a[1] & \cdots & \hat{y}_p^a[k] \\ u[1] & \cdots & u[k] \end{bmatrix}}_{\hat{D}} + G_{\text{aug}} \underbrace{\begin{bmatrix} \Delta\hat{y}_p^a[1] & \cdots & \Delta\hat{y}_p^a[k] \\ 0 & \cdots & 0] \end{bmatrix}}_{D_d} + \bar{N} = \bar{S} \end{aligned} \tag{4.31}$$

Now we have:

$$G_{\text{aug}}\bar{D} + \bar{N} = G_{\text{aug}}\hat{D} + (G_{\text{aug}}D_d + \bar{N}) = \bar{S} \tag{4.32}$$

Define the attacker's estimation of G as $\hat{G}_{\text{aug}}$. Assume the mean of $R[k]$ is zero, the attacker can estimate G by solving a least square problem as follows:

$$\hat{G}_{\text{aug}}^\top = (\hat{D}\hat{D}^\top)^{-1}\hat{D}\bar{S}^\top \tag{4.33}$$

Equation (4.33) can be expanded as follows:

$$\begin{aligned} \hat{G}_{\text{aug}}^\top &= (\hat{D}\hat{D}^\top)^{-1}\hat{D}\left(G_{\text{aug}}\hat{D} + (G_{\text{aug}}D_d + \bar{N})\right)^\top \\ &= G_{\text{aug}}^\top + (\hat{D}\hat{D}^\top)^{-1}\left(D_d^\top G_{\text{aug}}^\top + \bar{N}^\top\right) \end{aligned} \tag{4.34}$$

The variance of the estimation will be:

$$\begin{aligned} \text{Var}\left\{\hat{G}_{\text{aug}}^\top\right\} &= \mathbb{E}\left\{(\hat{G}_{\text{aug}}^\top - G^\top)(\hat{G}_{\text{aug}}^\top - G^\top)^\top\right\} \\ &= \mathbb{E}\left\{(\hat{D}\hat{D}^\top)^{-1}\left(D_d^\top G_{\text{aug}}^\top + \bar{N}^\top\right)\left((\hat{D}\hat{D}^\top)^{-1}\left(D_d^\top G_{\text{aug}}^\top + \bar{N}^\top\right)\right)^\top\right\} \end{aligned} \tag{4.35}$$

Equation (4.35) shows that the variance of the estimation is affected by the variance of $\bar{N}$ and D_d, i.e. $NR[k]$ and the precision of the estimation of the system output.

Remark 7. The precision of the estimation of the system output depends on different factors, including but not limited to plant property, system noise, system initial state, estimation of the initial state, estimation method, etc. ◁

4.2.2. Problem of System Immersion Coding Method: Under the Known-plaintext Attack

Equation (4.35) shows that the variance of the estimation is related to $NR[k]$ and the goodness of the estimation of the system output. No matter how small the error of the system output estimation is, we can always disturb the attacker's estimation by injecting a large enough $NR[k]$ part and maintaining the detection performance under the malicious parameter identification attack. However, if the estimation error of the system output is very small, the attacker can degrade the detection performance under the malicious parameter identification.

In the following section, we want to consider a particular case: the attacker can execute a Known-Plaintext attack and the least square method to estimate G, H. The case is an ideal case for the attacker. We will show that the system immersion coding method can easily be broken by the least-square estimation under the so-called *known-plaintext attack*.

The known-plaintext attack is one of the common attack methods in cryptography analysis. A classical definition of the known-plaintext attack in cryptography analysis is as follows:

Definition 4.2.1 (known-plaintext attack in cryptography)**.** [67, Page 20] In the known-plaintext attack, the adversary can learn one or more plaintext/ciphertext pairs generated using the same key in the known-plaintext attack. The adversary can then use the information to reveal the key or the plaintext of other ciphertext encrypted with the same key.

Similar to the definition 4.2.1, we formalize the known-plaintext attack for the system immersion coding method as follows:

Definition 4.2.2 (known-plaintext attack for system immersion coding method)**.** Under the known-plaintext attack, at timestamp k the attacker knows accurate $u[0:k]$, $y_p[0:k]$ and $s[0:k]$.

With the known-plaintext attack in the definition 4.2.2, the least-square estimation in (4.34) become:

$$\begin{aligned} \hat{G}_{\text{aug}}^{\top} &= (\hat{D}\hat{D}^{\top})^{-1}\hat{D}\left(G_{\text{aug}}\hat{D}+\bar{N}\right)^{\top} \\ &= G_{\text{aug}}^{\top} + (\hat{D}\hat{D}^{\top})^{-1}\bar{N}^{\top} \end{aligned} \tag{4.36}$$

The variance of the estimation will then be:

$$\begin{aligned} \text{Var}\left\{\hat{G}_{\text{aug}}^{\top}\right\} &= \mathbb{E}\left\{(\hat{G}_{\text{aug}}^{\top}-G^{\top})(\hat{G}_{\text{aug}}^{\top}-G^{\top})^{\top}\right\} \\ &= \mathbb{E}\left\{(\hat{D}\hat{D}^{\top})^{-1}\bar{N}^{\top}\left((\hat{D}\hat{D}^{\top})^{-1}\bar{N}^{\top}\right)^{\top}\right\} \end{aligned} \tag{4.37}$$

Equation 4.37 shows that the covariance of the estimation is still affected by $NR[k]$. It seems that the defender still can affect the accuracy of the attacker's estimation of G_{aug} by setting suitable $R[k]$. However, the following equation shows that even if the $\hat{G}_{\text{aug}}$ is different from that of G_{aug}, the attacker is still able to design a malicious attack sequence based on $\hat{G}_{\text{aug}}$.

$$\begin{aligned} M\hat{G}_{\text{aug}} &= M\left(G_{\text{aug}} + \left((\bar{D}\bar{D}^{\top})^{-1}\bar{N}^{\top}\right)^{\top}\right) \\ &= MG_{\text{aug}} + M\bar{N}(\bar{D}^{\top}\bar{D})^{-1} \\ &= \begin{bmatrix} I & H \end{bmatrix} \end{aligned} \tag{4.38}$$

Equation 4.38 shows that the attacker can easily simulate the system's behaviour equipped with a system immersion coding encoder with the estimated parameter. The attacker can inject an arbitrary signal into the $u[k]$ while using the simulated signal to cheat the anomaly detector.

Remark 8. Figure 4.2 shows an example of using the estimated parameter to inject a malicious attack. The example attack is similar to a covert attack. The attacker can first run a simulated system model and update the model based on real-time $u[k]$. Then the attacker can use estimated system immersion encoder $\hat{\mathscr{E}}$ to generate an $s^a[k]$ based on the simulated plant output $y^a[k]$ and the system input $u[k]$ as follows:

$$s^a[k] = \hat{G}_{\text{aug}}\begin{bmatrix} y^a[k] \\ u[k] \end{bmatrix} \tag{4.39}$$

The simulated $s^a[k]$ will be decoded as follows:

$$\begin{aligned} Ms^a[k] &= M\hat{G}_{\text{aug}}\begin{bmatrix} y^a[k] \\ u[k] \end{bmatrix} \\ &= \begin{bmatrix} I & H \end{bmatrix}\begin{bmatrix} y^a[k] \\ u[k] \end{bmatrix} \end{aligned} \tag{4.40}$$

. This means the controller will get the $y^a[k]$ and regard it as $y_p[k]$. Because the $y^a[k]$ is from a simulated model, it will behave like a normal system, and no alarm will trigger. Simultaneously, the attacker can inject malicious sequence $\Delta u^a[k]$ into the system to disturb the normal operation of the system. ◁

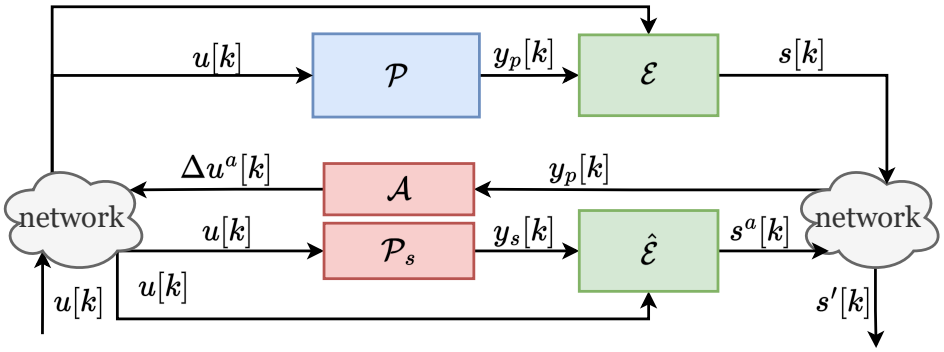


Figure 4.2: Example attack with estimated parameters

Remark 9. The known-plaintext attack is an ideal case for the attacker, which means when practically using the system immersion coding method, the attacker's estimation of parameters may not be good enough to execute the attack. However, we need to stress that the scenario close to the known-plaintext attack may happen, especially when the following conditions are met:

1. The plant is stable, i.e. the absolute value of all eigenvalues of A_p is smaller than 1.
2. The plant noise $w[k]$ and $v[k]$ are very small.

Assume the attacker tries to estimate the plant's output based on equation (4.30). The error between the true and estimated plant's output is as follows:

$$\begin{aligned}\Delta\hat{y}_p^a[k] &= y_p[k] - \hat{y}_p^a[k] \\ &= C_p A_p(x_p[k] - \hat{x}_p^a[k]) + C_p w[k-1] + v[k] \\ &= C_p A_p^k(x_p[0] - \hat{x}_p^a[0]) + \sum_{i=0}^{k-1} C_p A_p^i w[k-1-i] + v[k]\end{aligned} \tag{4.41}$$

If the plant is stable, the first item in equation (4.41) will decrease to 0, and the second item will not diverge. Suppose the noise $w[k]$ and $v[k]$ are very small. In that case, the second and third items in equation (4.41) will be very small. The known-plaintext attack may also be approximated when the attacker can physically deploy some sensor or eavesdrop on the sensor-encoder channel. ◁

4.3. Simulation Study

In this section, we will use the numerical example of testbench 1 and testbench 2 to study the property of the system immersion coding method introduced above.

4.3.1. Detection Performance

This section will use the simulation example on testbench 1 and testbench 2 to study the detection performance of the system immersion coding method. The numerical examples will also be used to verify Theorem 1, Theorem 2 and Theorem 3.

Testbench 1

For testbench 1, we use the following parameter to detect the replay attack and control-signal-injection zero-dynamics attack. The parameter is generated randomly and meets the detectability condition in Theorem 1 (maximum absolute value of eigenvalue is 1.01) and Theorem 3.

$$G = \begin{bmatrix} -1.1075 & 2.3244 \\ 0.2344 & -1.7119 \\ 2.2875 & 2.3530 \end{bmatrix}, \quad H = \begin{bmatrix} 4.5717 & 3.0028 \\ -0.1462 & -3.5811 \end{bmatrix}, \quad M = \begin{bmatrix} -0.2478 & 0.0872 & 0.3082 \\ 0.2106 & -0.1386 & 0.1162 \end{bmatrix} \tag{4.42}$$

Figures 4.3a and 4.3b shows the detection performance under the replay attack and the control-signal-injection zero-dynamics attack on testbench 1. Figure 4.3c shows the detection performance of the replay

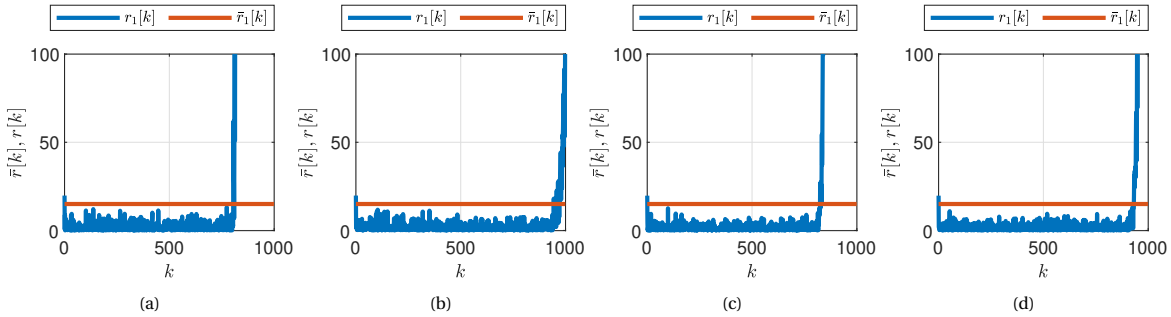


Figure 4.3: Detection performance of the system immersion coding method, , attacks start at $k_a = 800$. (a) residual value on testbench 1 under replay attack. (b) residual value on testbench 1 under the control-signal-injection zero-dynamics attack. (c) residual value on testbench 2 under replay attack. (d) residual value on testbench 2 under sensor-signal-injection zero-dynamics attack

attack. Different from Figure 3.3, the residual value exceeds $\bar{r}$ soon after the attack starts and successfully triggers an alarm. Figure 4.3b shows the detection performance of the control-signal-injection zero-dynamics attack. Different from Figure 3.4, the residual value exceeds $\bar{r}$ soon after the attack starts and successfully triggers an alarm.

Testbench 2

For testbench 2, we use the following parameter to detect the replay attack and sensor-signal-injection zero-dynamics attack. The parameter is generated randomly and meets the detectability condition in Theorem 1 (maximum absolute value of eigenvalue is 1.12) and Theorem 2.

$$G = \begin{bmatrix} 0.6618 & 0.2344 \\ -2.0123 & 2.2875 \\ -1.1075 & 2.3244 \end{bmatrix}, \quad H = \begin{bmatrix} 4.4205 \\ 4.5613 \end{bmatrix}, \quad M = \begin{bmatrix} 0.7409 & -0.4629 & 0.3808 \\ 0.5086 & -0.0901 & 0.4676 \end{bmatrix} \tag{4.43}$$

Figures 4.3c and 4.3d show the detection performance of the system immersion coding method under the replay attack and the sensor-signal-injection zero-dynamics attack on testbench 2. Figure 4.3c shows the detection performance of the replay attack. Different from Figure 3.6, the residual value exceeds $\bar{r}$ soon after the attack starts and successfully triggers an alarm. Figure 4.3d shows the detection performance of the control-signal-injection zero-dynamics attack. Different from Figure 3.7, the residual value exceeds $\bar{r}$ soon after the attack starts and successfully triggers an alarm.

4.3.2. Theorem Verification

We use testbench 2 to verify Theorem 1. We randomly generate different values of matrix H and make sure that the maximum absolute value of eigenvalue $\max|v|$ of $\Phi_H = (A_p - B_pK - LC_p + LHK)$ located in different regions: $[0,1), [1,1.2), [1.2,1.4)$. Figure 4.4 shows the verification result. From the figure, we can see that when $H = 0$ or $\max|v| \in [0,1)$, the system cannot detect replay attacks. When $\max|v| \geq 1$, the larger the $\max|v|$, the faster the residual exceeds the $\bar{r}$ and the attack is detected.

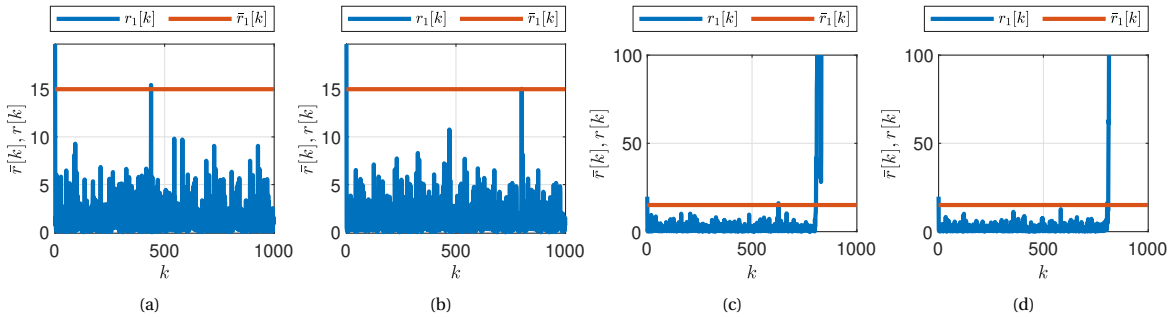


Figure 4.4: Test the theorem of replay attack with different values of H, attacks start at $k_a = 800$. (a) residual value when $H = 0$. (b) residual value when max eigenvalue $\max|v| \in [0,1)$. (c) residual value when max eigenvalue $\max|v| \in [1,1.2)$. (d) residual value when max eigenvalue $\max|v| \in [1.2,1.4)$.

We use testbench 1 to verify Theorem 2. The ν value of the control-signal-injection zero-dynamics attack of testbench 1 is not the eigenvalue of A_p, which does not meet the condition in Proposition 4.1.2. We prepare two values of H as follows. H_1 is very large, with $H_1 g = 0$. H_2 is small, but $H_2 g \neq 0$. The verification result is in Figure 4.5. Figure 4.5a shows that if $Hg = 0$, no alarm is triggered even with a very large H. Figure 4.5b shows that if $Hg \neq 0$, the alarm will be successfully triggered even if H is small. The simulation result matches with Theorem 3.

$$H_1 = \begin{bmatrix} 300 & 260 \\ 300 & 260 \end{bmatrix}, \quad H_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.44}$$

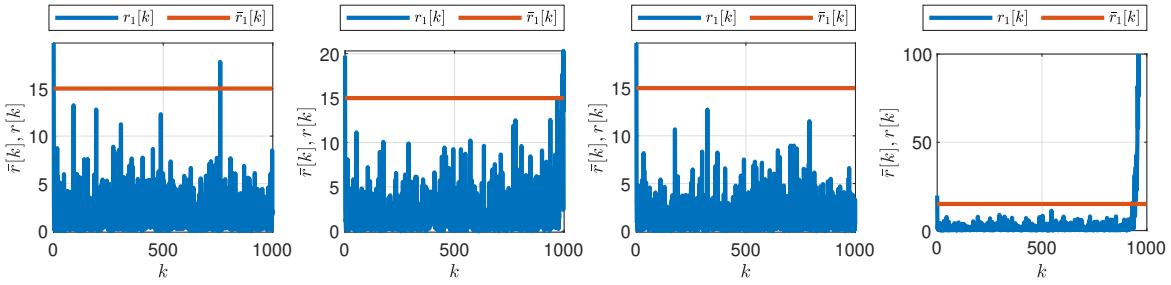


Figure 4.5: Test theorem of control-signal-injection zero-dynamics attack with different values of H and theorem of sensor-signal-injection zero-dynamics attack with different values of G', attacks start at $k_a = 800$. (a) residual value under control-signal-injection zero-dynamics attack when $H_1 g = 0$. (b) residual value under control-signal-injection zero-dynamics attack when $H_2 g \neq 0$. (c) residual value under sensor-signal-injection zero-dynamics attack when when $MG' = I$. (d) residual value under sensor-signal-injection zero-dynamics attack when when $MG' \neq I$.

We use testbench 2 to verify Theorem 3. We first randomly generate an M and then prepare two different values of G': the first G' meet $MG' = I$, and the second meets $MG' \neq I$. The verification result is in Figure 4.5. Figure 4.5c shows that if $MG' = I$, no alarm is triggered and the attacker's attack will keep stealthy. Figure 4.5d shows that if $MG' \neq I$, the alarm will be successfully triggered and the attack will be detected. The simulation result matches with Theorem 2.

4.3.3. Identification Resistance: Testbench 1 & Testbench 2

This section will show the system immersion method's defence ability against malicious parameter identification on testbench 1 and testbench 2. For each testbench, two scenarios are considered: $R[k] = 0 \forall k$ and $R[k]$ is from a uniform distribution in $[-10, 10]$. For each scenario, three identification situations are considered:

1. Assume the attacker cannot execute the known-plaintext attack, i.e. the attacker knows $u[0:k]$ and $s[0:k]$ but not $y_p[0:k]$. The attacker tries to use the system identification method to estimate the parameters. The system identification method is provided by MATLAB function `ssest`
2. Assume the attacker cannot execute the known-plaintext attack, i.e. the attacker knows $u[0:k]$ and $s[0:k]$ but not $y_p[0:k]$. The attacker tries first to estimate $\hat{y}^p[0:k]$ and uses the least square method to estimate the parameters.
3. Assume the attacker executes the known-plaintext attack, i.e. the attacker knows $u[0:k]$, $y_p[0:k]$, and $s[0:k]$. The attacker tries to use the least square method to estimate the parameters.

The identification result is shown in 4 matrices:

1. $|\hat{G} - G|_F$: The Frobenius norm of the estimation error of G. The closer to 0, the better. The sub-figures (a) of Figures 4.6 - Figures 4.17 show this value.
2. $|\hat{H} - H|_F$: The Frobenius norm of the estimation error of H. The closer to 0, the better. The sub-figures (b) of Figures 4.6 - Figures 4.17 show this value.
3. $|(M\hat{G})_i|$: The absolute value of each element in the matrix $(M\hat{G})$. The closer to the identity matrix, the better. The sub-figures (c) of Figures 4.6 - Figures 4.17 show this value. The 1 - 4 position in the graph is related to the element at (1, 1), (1, 2), (2, 1) and (2, 2) separately.
4. $|M\hat{G}\hat{H} - H|_F$: The Frobenius norm of the estimation error between $M\hat{G}\hat{H}$ and G. The closer to 0, the better. The sub-figures (d) of Figures 4.6 - Figures 4.17 show this value.

Remark 10. The matrices $|(M\hat{G})_i|$ and $|M\hat{G}\hat{H} - H|_F$ is more related to whether an attacker can inject a malicious attack based on the estimated parameter. ◁

Each testbench under each scenario with each identification method has been simulated 10 times. And the result of these 4 matrices is shown in the box plot.

Figures 4.6, 4.8 and 4.10 show the identification result of testbench 1 when $R[k] = 0$. Figures 4.7, 4.9 and 4.11 show the identification result of testbench 1 when $R[k]$ is from a uniform distribution in $[-10, 10]$.

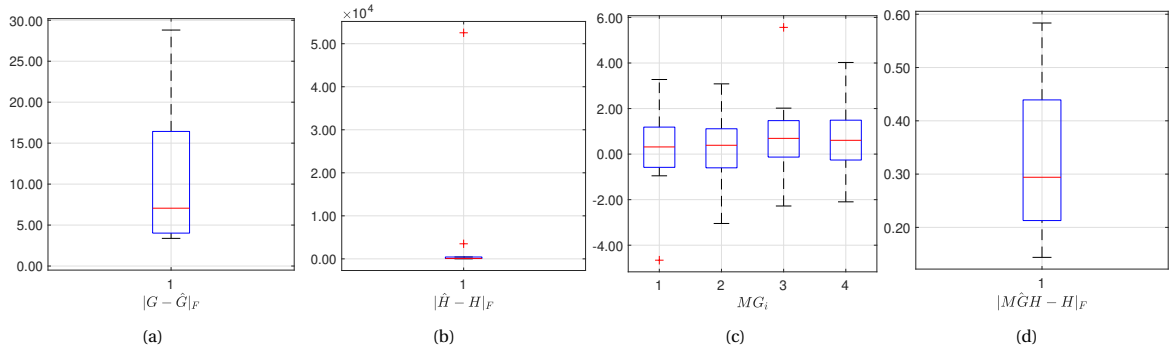


Figure 4.6: Identification of system immersion coding parameter on Testbench 1 when $R[k] = 0$ and with system identification method: (a) The Frobenius norm of the estimation error of G. (b) The Frobenius norm of the estimation error of H. (c) The absolute value of each element in the matrix $(M\hat{G})$; the 1 - 4 position in the graph is related to the element at (1, 1), (1, 2), (2, 1) and (2, 2) separately. (d) The Frobenius norm of the estimation error between $M\hat{G}\hat{H}$ and G.

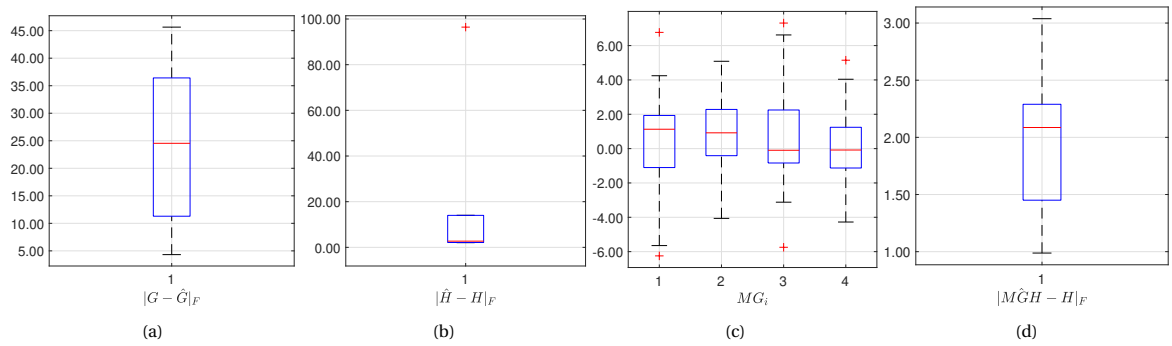


Figure 4.7: Identification of system immersion coding parameter on Testbench 1 when $R[k]$ are from uniform distribution and use system identification method: (a) The Frobenius norm of the estimation error of G. (b) The Frobenius norm of the estimation error of H. (c) The absolute value of each element in the matrix $(M\hat{G})$; the 1 - 4 position in the graph is related to the element at (1, 1), (1, 2), (2, 1) and (2, 2) separately. (d) The Frobenius norm of the estimation error between $M\hat{G}\hat{H}$ and G.

Figures 4.12, 4.14 and 4.16 show the identification result of testbench 2 when $R[k] = 0$. Figures 4.13, 4.15 and 4.17 show the identification result of testbench 2 when $R[k]$ is from a uniform distribution in $[-10, 10]$.

From the figures, we can conclude the following results:

1. The $NR[k]$ part can disturb the attacker estimation of G and H.
   (a) Comparing figure 4.6 and 4.7, 4.8 and 4.9, 4.12 and 4.13, 4.14 and 4.15, when $NR[k] \neq 0$, the estimation is less accuracy on matrices $|\hat{G} - G|_F$, $|M\hat{G}|_F$, $|M\hat{G}\hat{H} - H|_F$then the case $NR[k] = 0$. The accuracy of $|\hat{H} - H|_F$ varies, which may be because $\hat{H}$'s estimation is related to $\hat{G}$.
   (b) Compared subfigures (a) and (b) of figures 4.10 and 4.11, 4.16 and 4.17, when $NR[k] \neq 0$, the estimation is less accuracy on first two matrices.
2. The system immersion coding method is vulnerable to the known-plaintext attack.
   (a) The figures 4.10 and 4.16 show that when $NR[k] = 0$, the attacker will achieve accurate estimation of all 4 matrices.
   (b) In figures 4.11, and 4.17, the subfigures (c) and (d) show that $M\hat{G}$ is close to $[I \quad H]$, even their subfigures (a) and (b) show that the attackers' estimate of G and H is not accurate. This verifies the equation (4.38)

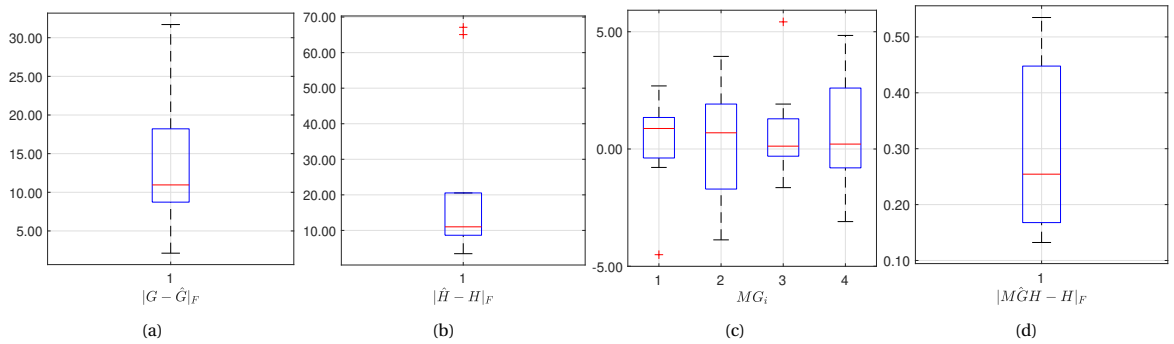


Figure 4.8: Identification of system immersion coding parameter on Testbench 1 when $R[k] = 0$ and with least square method: (a) The Frobenius norm of the estimation error of G. (b) The Frobenius norm of the estimation error of H. (c) The absolute value of each element in the matrix $(M\hat{G})$; the 1 - 4 position in the graph is related to the element at (1, 1), (1, 2), (2, 1) and (2, 2) separately. (d) The Frobenius norm of the estimation error between $M\hat{G}\hat{H}$ and G.

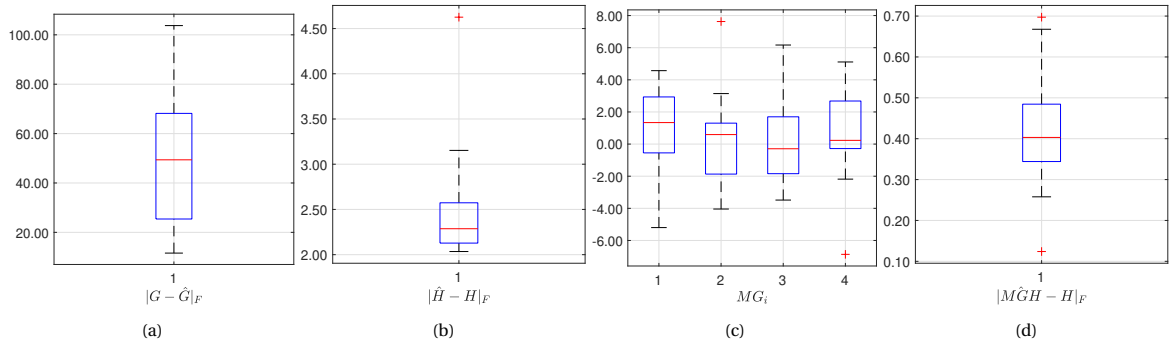


Figure 4.9: Identification of system immersion coding parameter on Testbench 1 when $R[k]$ are from uniform distribution and with least square method: (a) The Frobenius norm of the estimation error of G. (b) The Frobenius norm of the estimation error of H. (c) The absolute value of each element in the matrix $(M\hat{G})$; the 1 - 4 position in the graph is related to the element at (1, 1), (1, 2), (2, 1) and (2, 2) separately. (d) The Frobenius norm of the estimation error between $M\hat{G}\hat{H}$ and G.

3. When the known-plaintext attack cannot be executed, the identification performance of the attacker is not good. Figures 4.6, 4.8, 4.10, 4.12, 4.14 and 4.16 show that the estimation are not good on 4 matrices even when $NR[k] = 0$. For the figure when $NR[K] \neq 0$, the result is worse.
4. When the known-plaintext attack cannot be executed, first estimating $\hat{y}_p[0:k]$ then using the least-square method always has higher precision than the system identification method.

The results show that the system immersion coding method can disturb the attacker's parameter identification of the accurate G and H. However, under a known-plaintext attack, even if the estimation is inaccurate, it is still sufficient to design a malicious attack based on the estimated G and H.

4.4. Conclusion

In this chapter, we studied Problem 2.1, Problem 2.2 and Problem 2.3. First, we proposed a system immersion coding method to detect malicious attacks actively. With suitably defined parameters, the system immersion coding method can detect the replay attack, certain types of the control-signal-injection zero-dynamics attack and certain types of the sensor-signal-injection zero-dynamics attack. We provided theorems to guide the parameter design and condition of detecting malicious attacks. The system immersion coding method can disturb the attacker's parameter identification by enlarging the variance of the attacker's estimation. However, we show that, under a known-plaintext attack, the attacker can still inject a malicious attack with inaccurately estimated parameters.

Although the system immersion coding method's capability of defending against malicious parameter identification is not working, the detection performance shows that the additive integration of $u[k]$, i.e. $Hu[k]$ helps detect the replay attack and the control-signal-injection zero-dynamics attack.

In the next chapter, inspired by the cryptography analysis of the known-plaintext attack, we will propose

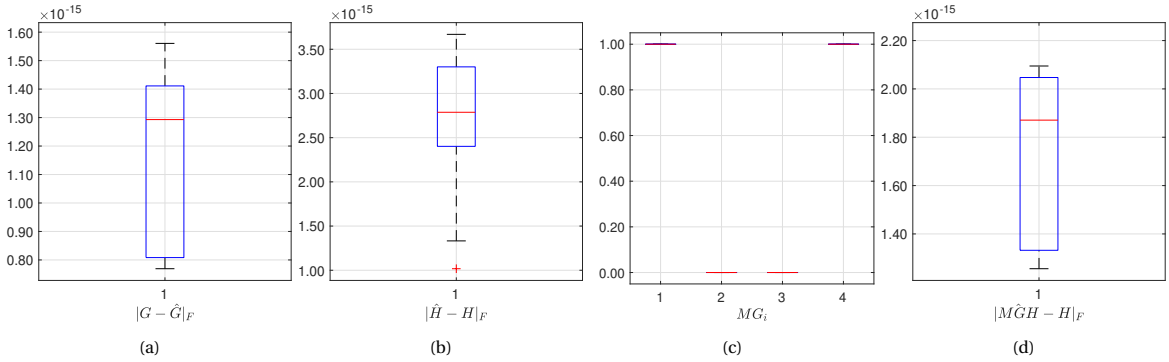


Figure 4.10: Identification of system immersion coding parameter on Testbench 1 when $R[k] = 0$ and with least square method under known-plaintext attack: (a) The Frobenius norm of the estimation error of G. (b) The Frobenius norm of the estimation error of H. (c) The absolute value of each element in the matrix $(M\hat{G})$; the 1 - 4 position in the graph is related to the element at (1, 1), (1, 2), (2, 1) and (2, 2) separately. (d) The Frobenius norm of the estimation error between $M\hat{G}\hat{H}$ and G.

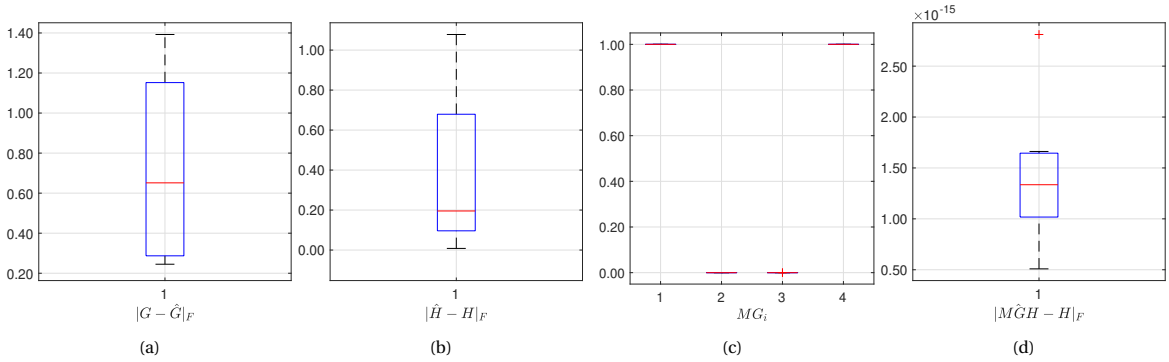
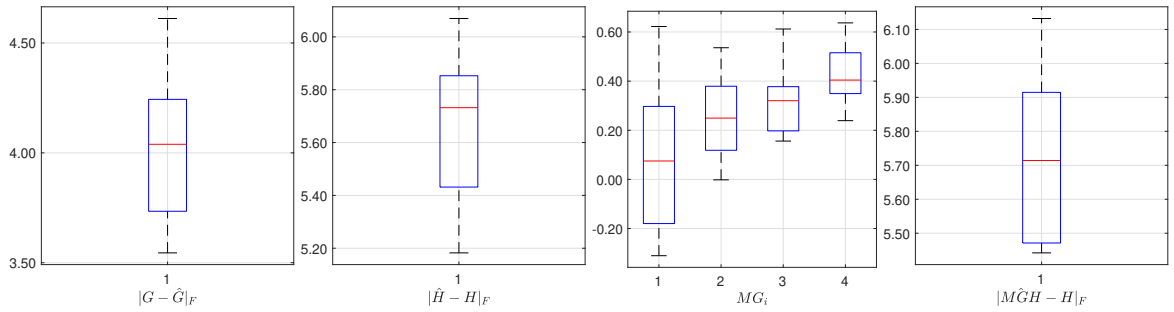


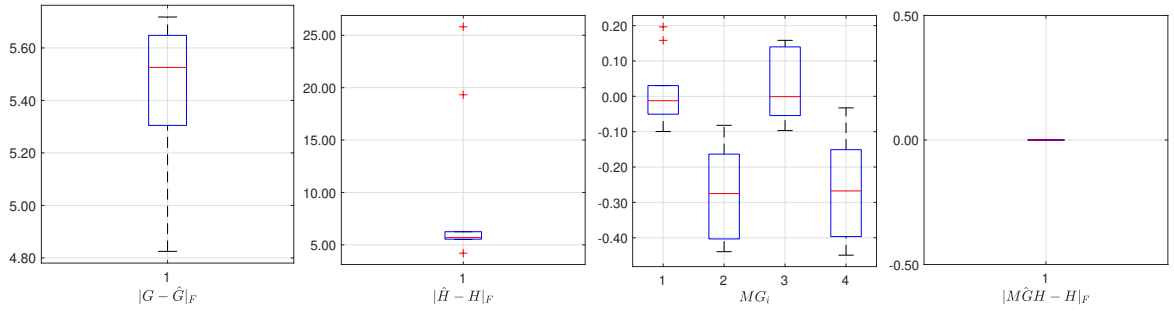
Figure 4.11: Identification of system immersion coding parameter on Testbench 1 when $R[k]$ are from uniform distribution and with least square method under known-plaintext attack: (a) The Frobenius norm of the estimation error of G. (b) The Frobenius norm of the estimation error of H. (c) The absolute value of each element in the matrix $(M\hat{G})$; the 1 - 4 position in the graph is related to the element at (1, 1), (1, 2), (2, 1) and (2, 2) separately. (d) The Frobenius norm of the estimation error between $M\hat{G}\hat{H}$ and G.

the hybrid multiplicative watermarking method, which can defend against malicious parameter identification even under the known-plaintext attack. Besides, the hybrid multiplicative watermarking method also integrates the additive integration of $u[k]$ to improve the detection performance.



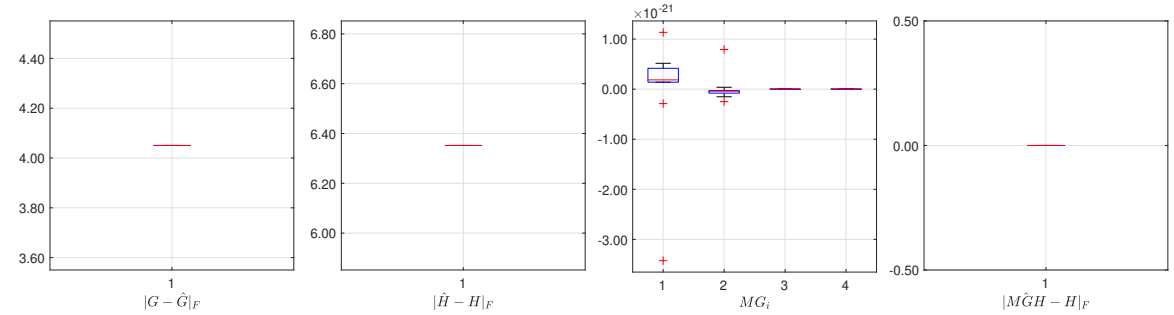
(a) (b) (c) (d)

Figure 4.12: Identification of system immersion coding parameter on Testbench 2 when $R[k] = 0$ and with system identification method: (a) The Frobenius norm of the estimation error of G. (b) The Frobenius norm of the estimation error of H. (c) The absolute value of each element in the matrix $(M\hat{G})$; the 1 - 4 position in the graph is related to the element at (1, 1), (1, 2), (2, 1) and (2, 2) separately. (d) The Frobenius norm of the estimation error between $M\hat{G}\hat{H}$ and G.



(a) (b) (c) (d)

Figure 4.13: Identification of system immersion coding parameter on Testbench 2 when $R[k]$ are from uniform distribution and with system identification method: (a) The Frobenius norm of the estimation error of G. (b) The Frobenius norm of the estimation error of H. (c) The absolute value of each element in the matrix $(M\hat{G})$; the 1 - 4 position in the graph is related to the element at (1, 1), (1, 2), (2, 1) and (2, 2) separately. (d) The Frobenius norm of the estimation error between $M\hat{G}\hat{H}$ and G.



(a) (b) (c) (d)

Figure 4.14: Identification of system immersion coding parameter on Testbench 2 when $R[k] = 0$ and with least square method: (a) The Frobenius norm of the estimation error of G. (b) The Frobenius norm of the estimation error of H. (c) The absolute value of each element in the matrix $(M\hat{G})$; the 1 - 4 position in the graph is related to the element at (1, 1), (1, 2), (2, 1) and (2, 2) separately. (d) The Frobenius norm of the estimation error between $M\hat{G}\hat{H}$ and G.

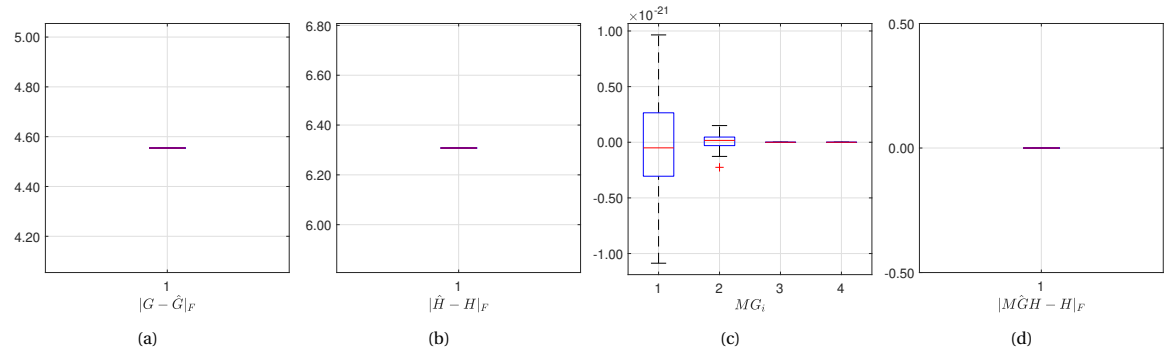


Figure 4.15: Identification of system immersion coding parameter on Testbench 2 when $R[k]$ are from uniform distribution and with least square method: (a) The Frobenius norm of the estimation error of G. (b) The Frobenius norm of the estimation error of H. (c) The absolute value of each element in the matrix ($M\hat{G}$); the 1 - 4 position in the graph is related to the element at (1, 1), (1, 2), (2, 1) and (2, 2) separately. (d) The Frobenius norm of the estimation error between $M\hat{G}\hat{H}$ and G.

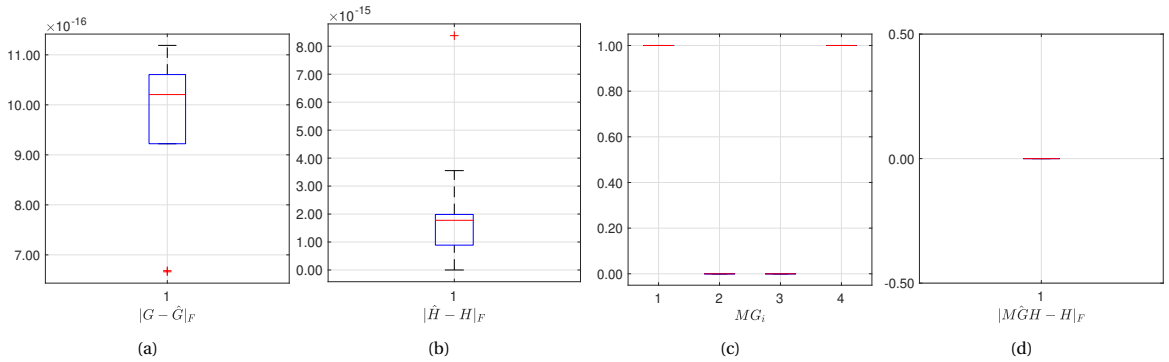


Figure 4.16: Identification of system immersion coding parameter on Testbench 2 when $R[k] = 0$ and with least square method under known-plaintext attack: (a) The Frobenius norm of the estimation error of G. (b) The Frobenius norm of the estimation error of H. (c) The absolute value of each element in the matrix ($M\hat{G}$); the 1 - 4 position in the graph is related to the element at (1, 1), (1, 2), (2, 1) and (2, 2) separately. (d) The Frobenius norm of the estimation error between $M\hat{G}\hat{H}$ and G.

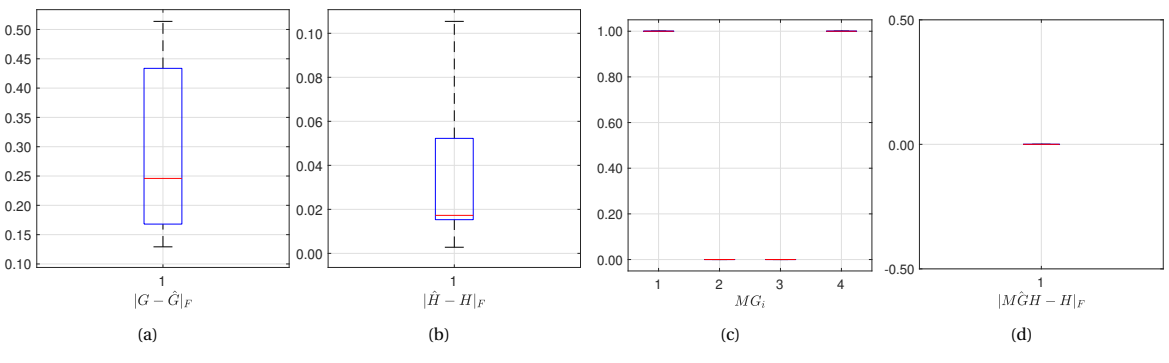


Figure 4.17: Identification of system immersion coding parameter on Testbench 2 when $R[k]$ are from uniform distribution and with least square method under known-plaintext attack: (a) The Frobenius norm of the estimation error of G. (b) The Frobenius norm of the estimation error of H. (c) The absolute value of each element in the matrix ($M\hat{G}$); the 1 - 4 position in the graph is related to the element at (1, 1), (1, 2), (2, 1) and (2, 2) separately. (d) The Frobenius norm of the estimation error between $M\hat{G}\hat{H}$ and G.

5

Hybrid Multiplicative Watermarking: Theory

Chapter 4 shows that the system immersion coding method is vulnerable to the known-plaintext attack. The known-plaintext is widely studied in cryptography analysis. Cryptographical tools defend the plaintext attack by making the problem of learning the cypher key equal to solving a so-called *computational hardness assumptions*. For example, in the Diffie Hellman key exchange protocol, the attacker must solve the discrete logarithm problem to find the cypher key directly.

Therefore, a possible way is to use a structure that is hard to identify to overcome the known-plaintext-related malicious parameter identification. The problem is whether there is any problem that is proven or assumed to be a tough problem in the system identification field. In [68], the identification of discrete-time hybrid switching affine systems has been proved to be $\mathcal{NP}$-hard problem, which inspires us to propose the hybrid-multiplicative watermarking method.

Based on Problems 3.1, 3.2, and 3.3, we will extend the multiplicative watermarking method and propose the *hybrid multiplicative watermarking method* in this chapter. We will then provide guidelines for design parameters and the switching rule of the hybrid multiplicative watermarking method to help detect malicious attacks. We will then study its performance in defending malicious parameter identification.

Figure 5.1 shows the considered CPS structure. Compared to the structure in Chapter 3, the active detection part is specified to be a switching multiplicative watermarking pair $(\mathscr{W}, \mathscr{Q})$.

The eavesdropping attacker $\mathscr{A}_e$ will then be defined as the following threat model:

System knowledge: The attacker knows the parameters of the plant and controller models $\{A_p, B_p, C_p, L, K\}$.

Disclousure resources: The attacker has direct access to signals y_w and u transmitted over the communication network. The set of information available to the attacker at time k can be therefore defined as:

$$\mathscr{I}_a[k] \triangleq \{A_p, B_p, C_p, L, K, u[0:k], y_w[0:k]\}. \tag{5.1}$$

Note that $\theta_w[0], \theta_q[0] \notin \mathscr{I}_a$.

Attack objective: The malicious agent attempts to reconstruct the multiplicative watermarking parameters $\theta_w[k]$ and $\theta_q[k]$ for all $k \geq K_{id}^a, k \in \mathbb{Z}_+$. Without loss of generality, $K_{id}^a = 0$.

5.1. Background: Hybrid System Theory

In this section, we will provide a concise introduction to the fundamental hybrid systems theory. This section starts with the definition of hybrid systems and then focuses on two specific types: piecewise affine systems and switching affine systems. The stability theorem and the identification methods of these two types of systems will be introduced. The contents and the definitions of hybrid systems, the piecewise affine systems and the switching affine systems in this thesis are mainly from [68–71].

5.1.1. Hybrid System Introduction

This thesis only considers *discrete-time hybrid systems*. A discrete-time hybrid system can be represented by two forms: *state-space (SS) form* and *input-output (IO) form*. A discrete-time hybrid system model in state-

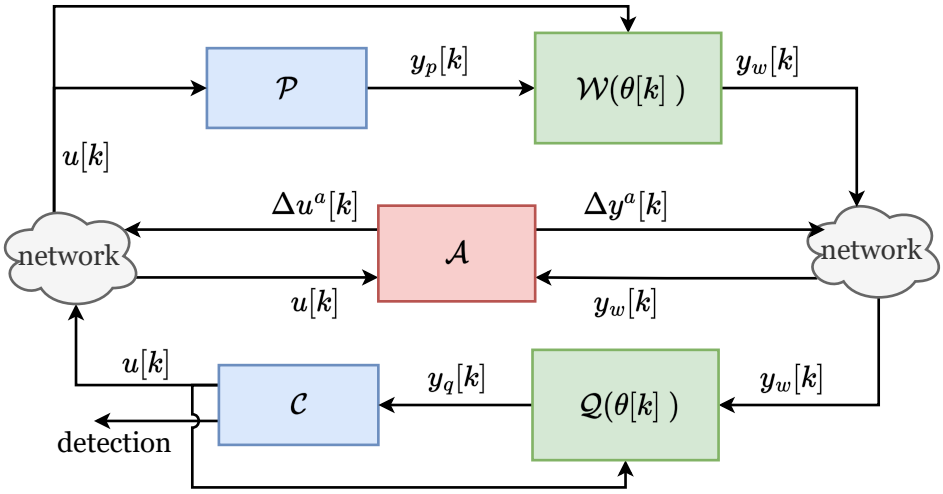


Figure 5.1: Overal System Description

space (SS) form is described by

$$\textbf{SS form} \quad \begin{cases} x[k+1] = \mathbf{f}_{q[k]}(x[k], u[k]) + w[k] \\ y[k] = \mathbf{g}_{q[k]}(x[k], u[k]) + v[k], \end{cases} \tag{5.2}$$

In the SS Form, the $x[k] \in \mathbb{R}^{n_x}$, $u[k] \in \mathbb{R}^{n_u}$, and $y[k] \in \mathbb{R}^{n_y}$ are, respectively, the continuous state, input and output vectors. The $w[k] \in \mathbb{R}^{n_x}$ and $v[k] \in \mathbb{R}^{n_y}$ are state noise vectors and output noise vectors. The hybrid system have a finite number s of possible modes: $q[k] \in \{1,\dots,s\}$ and $(\mathbf{f}_{q[k]}, \mathbf{g}_{q[k]}) \in \{(\mathbf{f}_1,\mathbf{g}_1),\dots,(\mathbf{f}_s,\mathbf{g}_s)\}$, with $\mathbf{f}_j : \mathbb{R}^{n_x+n_u} \to \mathbb{R}^{n_x}$ and $\mathbf{g}_j : \mathbb{R}^{n_x+n_u} \to \mathbb{R}^{n_y}$. At the time k, the system activates a particular mode j, i.e., $q[k] = j$, and the "submodel" (or "mode") $(\mathbf{f}_j, \mathbf{g}_j)$ is active.

A discrete-time hybrid system model in the input-output (IO) form is described by:

$$\textbf{IO form:} \quad y[k] = \boldsymbol{f}_{q[k]}(r[k]) + \varepsilon[k] \tag{5.3}$$

where $y[k] \in \mathbb{R}^{n_y}$ and $r[k] \in \mathbb{R}^d$ are respectively, system output vectors and regression vector. The regression vector always consists of current input, historical input and output in a finite horizon. The $\varepsilon[k]$ is the output noise. The IO form has a finite time number s of modes $q[k] \in \{1,\dots,s\}$ and $\boldsymbol{f}_{q[k]} \in \{\boldsymbol{f}_1,\cdots,\boldsymbol{f}_s\}$. At each timestamp k the system activates a particular mode j, i.e., $q[k] = j$ and $\boldsymbol{f}_{q[k]} = \boldsymbol{f}_j$.

There are different kinds of discrete-time hybrid systems. One specific category is the hybrid system whose dynamics of each mode is affine, i.e. $\boldsymbol{f}_i$ is affine for all $i \in \{1,\dots,s\}$ in (5.3) or $\mathbf{f}_i, \mathbf{g}_i$ are affine for all $i \in \{1,\dots,s\}$ in (5.2). In this thesis, we name this category of the hybrid system as *switching affine systems (SAS)*. More specifically, in this thesis, we mainly focus on two classes of the discrete-time switching affine systems: the discrete-time *piecewise affine systems (PWA)* and discrete-time *arbitrarily switching affine systems (aSAS)*.

Piecewise Affine Systems

In a discrete-time piecewise affine system, the state $q[k]$ depends on the continuous state $x[k]$ and input $u[k]$ [69], that is:

$$q[k] = i \quad \text{if} \begin{bmatrix} x[k] \\ u[k] \end{bmatrix} \in \Omega_i, \quad i = 1,\dots,s \tag{5.4}$$

Where Ω_j, $j \in \{1,\dots,s\}$, are regions that form a partition of the state-input plane. The partition regions can be an arbitrary shape that forms a complete partition of the state-input domain $\mathscr{R}^(n_x+n_u)$, i.e. $\cup_{i=1}^{s} \Omega_i = \Omega$, and $\Omega_i \cap \Omega_j = \emptyset$, $\forall i \neq j$. Most of the literature considers the partition are convex polyhedrons, i.e. [71]

$$\Omega_i = \left\{ \begin{bmatrix} x \\ u \end{bmatrix} \in \mathbb{R}^{n_x+n_u} : H_i \begin{bmatrix} x \\ u \\ 1 \end{bmatrix} \leq \mathbf{0} \right\} \tag{5.5}$$

with $H_i \in \mathbb{R}^{\mu_i \times (n_x+n_u)+1}, i = 1,\dots,s$, and μ_i is the number of linear inequalities defining the i th polyhedral region Ω_i.

The SS form of the discrete-time PWA systems is as follows:

$$\textbf{SS Form:} \quad \begin{bmatrix} x[k+1] \\ y[k] \end{bmatrix} = \begin{bmatrix} A_{q[k]} & B_{q[k]} \\ C_{q[k]} & D_{q[k]} \end{bmatrix} \begin{bmatrix} x[k] \\ u[k] \end{bmatrix} + \begin{bmatrix} f_{q[k]} \\ g_{q[k]} \end{bmatrix} \quad \text{if} \begin{bmatrix} x[k] \\ u[k] \end{bmatrix} \in \Omega_i, \quad i = 1,\dots,s \tag{5.6}$$

where $(A_{q[k]}, B_{q[k]}, C_{q[k]}, D_{q[k]}, f_{q[k]}, g_{q[k]}) \in \{(A_1, B_1, C_1, D_1, f_1, g_1), \cdots, (A_s, B_s, C_s, D_s, f_s, g_s)\}$

The SS form 5.6 can also be written as:

$$\textbf{SS Form:} \quad \begin{cases} x[k+1] = \sum_{i=0}^{N} \beta_{q[k]} \left(A_{q[k]} x[k] + B_{q[k]} u[k] + f_i \right) \\ y_w[k] = \sum_{i=0}^{N} \beta_{q[k]} \left(C_{q[k]} x[k] + D_{q[k]} u[k] + g_i \right) \end{cases} \quad \beta_{q[k]} = \begin{cases} 1, \text{ if} \begin{bmatrix} x[k] \\ u[k] \end{bmatrix} \in \Omega_i \\ 0, \text{ otherwise} \end{cases} \tag{5.7}$$

Remark 11. If $(f_i, g_i) = 0, \forall i \in \{1, \cdots, s\}$ in models (5.6) and (5.7), we then call the system as *piecewise linear systems (PWL).* ◁

The PWARX IO form of the discrete-time PWA systems is as follows [71]:

$$\begin{aligned} \textbf{IO Form:} \quad & y[k] = \Theta_j \begin{bmatrix} r[k] \\ 1 \end{bmatrix} \quad \text{if} \quad r[k] \in \mathscr{R}_j, \quad j = 1,\dots,s \\ & r[k] = \left[y^T[k-1], \cdots, y^T[k-n_a], u^T[k], u^T[k-1], \cdots, u^T[k-n_b] \right]^T \end{aligned} \tag{5.8}$$

where $r[k]$ is the regression vector corresponding to the PWARX model with fixed orders n_a and n_b, the regions $\{\mathscr{R}_j\}_{j=1}^{s}$ form a complete partition of the regressor domain $\mathscr{R} \subseteq \mathbb{R}^{n_a n_y + (n_b+1) n_u}$. In our thesis, We consider each region $\mathscr{R}_j$ is a convex polyhedron described by [71]

$$\mathscr{R}_j = \left\{ r \in \mathbb{R}^{n_a n_y + (n_b+1) n_u} : \bar{H}_j \begin{bmatrix} r \\ 1 \end{bmatrix} \le 0 \right\} \tag{5.9}$$

where $\bar{H}_j \in \mathbb{R}^{\mu_j \times (n_a n_y + (n_b+1) n_u)}, j = 1,\dots,s, \mu_j$ is the number of linear inequalities needed to define the j th polyhedral region $\mathscr{R}_j$.

The SS and IO form of the PWA can sometimes convert to each other. One reason for studying the conversion is that we want to use theories derived from another form to analyse a given form of a PWA system.

Converting an IO model of a PWA system to an equivalent SS form can be straightforward [71]. In [69, 71], the researchers provide solutions for transforming a SS form PWA model to an IO form IO model. Assume a PWA system in SS form has s modes, the researcher studies the following questions:

1. Whether the SS form PWA system has an equivalent IO form.
2. If the SS form PWA admit an equivalent IO form, how many modes $\bar{s}$ the equivalent IO form has?
3. If the equivalent IO form exists, how can we get that?

Some significant results are as follows, the proofs are omitted, and readers can turn to the reference literature for detailed proof.

Definition 5.1.1 (Input-output equivalence)**.** [69, Def.2.4.] Models (5.9) and (5.6) are said to be *(input-output) equivalent* if the sets of input-output trajectories of (5.9) and (5.6) coincide.

Condition 1. *[71, Cond. C1] Let $\bar{n} \in \mathbb{Z}^+$. For any two initial states x and $\boldsymbol{x}'$, and input sequence $u(\cdot)$, satisfying $y(k; \boldsymbol{x}, u(\cdot)) = y\left(k; \boldsymbol{x}', u(\cdot)\right)$ for $k = 0, 1, \dots, \bar{n}-1$, it holds that*

$$y(\bar{n}; \boldsymbol{x}, u(\cdot)) = y\left(\bar{n}; \boldsymbol{x}', u(\cdot)\right). \tag{5.10}$$

Remark 12. [71] If Condition 1 holds, any two output trajectories of a PWA system model (5.6) that are indistinguishable from time 0 to time $\bar{n}-1$, are also equal at time $\bar{n}$. ◁

Condition 2. *[71, Cond. C2] Let $\bar{n} \in \mathbb{Z}^+$. For any feasible mode sequence $(i_0, i_1, \ldots, i_{\bar{n}})$, there exists a vector $\xi \in \mathbb{R}^{2\bar{n}+1}$ such that*

$$\left[\xi^\top - 1\right] \begin{bmatrix} \Gamma_{i_1,\ldots,i_{\bar{n}}} \\ \nu_{i_0,i_1,\ldots,i_{\bar{n}}} \end{bmatrix} V_{i_0,i_1,\ldots,i_{\bar{n}}} = 0, \tag{5.11}$$

where $\Gamma_{i_1,\ldots,i_{\bar{n}}}$ and $\nu_{i_0,i_1,\ldots,i_{\bar{n}}}$ are defined in Table I of [71]. [1].

Theorem 4. *[71, Thm.1] A PWA state space model (3) with $\Omega = \mathbb{R}^{n_x+n_u}$ admits an equivalent PWARX representation (15) if and only if there exists $\bar{n} \in \mathbb{Z}^+$ such that Condition 1 and Condition 2 are satisfied.* □

Remark 13. The proof of the theorem 4 in [71] provides a constructive method to convert the PWA SS model to an equivalent PWA IO model if it exits. ◁

Remark 14. [71] There are some comments about the conversion from the PWA SS form to the IO form:

1. A given PWA state space model (5.6) may not admit any equivalent PWARX representation (5.9).
2. A given PWARX model (5.9) always admits an equivalent PWA state space representation (5.6).
3. If Conditions 1 and 2 are satisfied for $\bar{n} \in \mathbb{Z}^+$, the constructive method proposed in the proof of Theorem 4 provides an equivalent PWARX model with $\bar{s}$ modes, where $\bar{s}$, being the number of feasible mode sequences of length $\bar{n}+1$, is at most $s^{\bar{n}+1}$. Although the number of modes of the constructed equivalent PWARX model can sometimes be reduced, $\bar{s}$ is typically much higher than s. ◁

Definition 5.1.2 (observability of PWA SS form)**.** [71, Def.3.2.] Model (5.6) is called *observable* if the mapping $\begin{bmatrix} x_0 \\ u_0^\infty \end{bmatrix} \to \begin{bmatrix} y_0^\infty \\ u_0^\infty \end{bmatrix}$, with $\boldsymbol{y}(\cdot) = y(\cdot; x_0, u)$, is invertible.

Definition 5.1.3 (observable in finite time)**.** [71, Def.3.3.] If model (5.6) is observable, it is called *observable in finite time N* if there exists a nonnegative integer N such that the mapping $\begin{bmatrix} x_0 \\ u_0^* \end{bmatrix} \to \begin{bmatrix} y_0^N \\ u_0^* \end{bmatrix}$ is invertible. Otherwise, it is called observable in infinite time.

Proposition 5.1.1. *[71, Prop.3.1.] A PWA state space model (5.6) with $\Omega = \mathbb{R}^{n_x+n_u}$ that is observable in infinite time does not admit any equivalent PWARX representation (5.9).* □

Remark 15. The above definitions, theorems, remarks and propositions are for the PWA model (5.6). We can easily conclude that they are also suitable for PWL systems cases because the PWL systems belong to PWA models. ◁

Arbitrary Switching Affine Systems

In a discrete-time arbitrary switching affine system, the state $q[k]$ is assumed to depend on an arbitrary external signal but not depend on the continuous state $x[k]$ and input $u[k]$ [70]. The SS form of a discrete-time arbitrary switching affine system is as follows [70] :

$$\textbf{SS Form:} \quad \begin{bmatrix} x[k+1] \\ y[k] \end{bmatrix} = \begin{bmatrix} A_{q[k]} & B_{q[k]} \\ C_{q[k]} & D_{q[k]} \end{bmatrix} \begin{bmatrix} x[k] \\ u[k] \end{bmatrix} + \begin{bmatrix} f_{q[k]} \\ g_{q[k]} \end{bmatrix} \tag{5.12}$$

where $(A_{q[k]}, B_{q[k]}, C_{q[k]}, D_{q[k]}, f_{q[k]}, g_{q[k]}) \in \{(A_1, B_1, C_1, D_1, f_1, g_1), \cdots, (A_s, B_s, C_s, D_s, f_s, g_s)\}$. The discrete state $q[k]$ determines the currently activated mode and is determined by an external signal.

The IO form of a discrete-time arbitrary switching affine system is as follows [70]:

$$\begin{aligned} \textbf{IO Form:} \quad & y[k] = \Theta_j \begin{bmatrix} r[k] \\ 1 \end{bmatrix} \\ & r[k] = \left[y^T[k-1], \cdots, y^T[k-n_a], u^T[k], u^T[k-1], \cdots, u^T[k-n_b]\right]^T \end{aligned} \tag{5.13}$$

Where $r[k]$ is the regression vector corresponding to the PWARX model with fixed orders n_a and n_b.

In [70], the authors discuss the problem of finding an equivalent IO model of a given SS form of a discrete-time arbitrary switching affine system. The equivalence IO model problem is not important for this thesis. The reader can read [70] for details.

Remark 16. In [70], the authors use the name "switched affine systems". This thesis uses the name "arbitrary switching affine system" to stress the discrete state is irrelevant to the system state and system input. ◁

[1] Because Table I is very long, interesting readers can refer to [71] for the detail of Table I.

5.1.2. Stability of Discrete-time Switching Affine Systems

Different stability-related research questions exist in hybrid systems [72, 73]. The thesis will focus on the stability analysis of discrete-time switching affine systems. In this section, we will first consider the Lyapunov Stability of discrete-time autonomous switching affine systems, especially the globally uniform asymptotically stable (G.U.A.S.) Then we will go through some important results of the input-state-stability of discrete-time switching affine systems.

Lyapunov Stability of Autonomous Discrete-time Switching Affine Systems

Consider the following autonomous discrete-time switching affine systems:

$$x[k+1] = A_i x[k], k \in \mathbb{Z}^+ \tag{5.14}$$

where $A_i \in \{A_1, \cdots, A_s\}$

The globally uniform asymptotically stable (G.U.A.S.) is then defined as follows:

Definition 5.1.4 (Globally Uniform Asymptotically Stable (G.U.A.S.))**.** [72] If a hybrid system is globally asymptotically stable under the arbitrary switch with all initial conditions, then it is globally uniform asymptotically stable (G.U.A.S.).

Remark 17. For a mathematical description of the G.U.A.S, please see [74]. ◁

Then based on that, a sufficient Lypuanov-Function-based criterion for G.U.A.S. is as follows:

Theorem 5. *[75, Lemma 2.1] If there exists a symmetric positive-definite matrix P such that*

$$A_i^T P A_i - P < 0, \forall i \in \{1, 2, \cdots, s\} \tag{5.15}$$

then the state trajectory of the system (5.14) is G.U.A.S. □

Remark 18. A function $V(x) = x^T P x$ with a P matrix meets the requirement in 5 is a Lyapunov function for system (5.14). We called it *common quadratic Lyapunov Function* for system (5.14). The definitions of the Lyapunov function of model (5.14) are similar to that of linear-time-invariant systems, so we will not introduce them in this thesis. ◁

Theorem 5 can be relaxed as follows:

Theorem 6. *[72, Theo.5] If there exist s positive definite symmetric matrices $P_i \in \mathbb{R}^{n_x \times n_x}$ $\left(P_i = P_i^T\right)$ and matrices $F_i, G_i \in \mathbb{R}^{n_x \times n_x} (i \in \mathscr{I})$, satisfying*

$$\begin{bmatrix} A_i F_i^T + F_i A_i^T - P_i & A_i Q_i - F_i \\ G_i^T A_i^T - F_i^T & P_j - C_i - C_i^T \end{bmatrix} < 0 \quad \forall i, j \in \{1, 2, \cdots, s\} \tag{5.16}$$

then the switched linear system (5.14) is asymptotically stable. □

For piecewise affine systems, theorem 5 can be relaxed as follows:

Theorem 7. *[76, Theo.1] The PWA (5.14) is asymptotically stable on R if there exist s matrices P_i, such that the following LMIs are satisfied:*

$$\begin{aligned} P_i > 0 \quad & \forall i \in \{1, 2, \cdots, s\} \\ A_j^T P_i A_j - P_j < 0 \quad & \forall (j, i) \in \{1, 2, \cdots, s\} \end{aligned} \tag{5.17}$$

□

Remark 19. A Lyapunov function can be designed based on the s matrices P_i meet the theorem 7:

$$v(x) = x^T P_i x \quad \text{if} x \in \mathscr{X}_i \tag{5.18}$$

where $\mathscr{X}_i, i \in \{1, \ldots, s\}$, are regions that form a partition of the state plane. We call the Lyapunov function in (5.18) as *piecewise affine quadratic Lyapunov function.* ◁

Input-State-Stability of Discrete-time Switching Affine Systems

In Section 5.1.2, we review some important results of the G.U.A.S. of the autonomous SASs. In this section, we will consider the stability of the SASs with the input signal. The result in this section is mainly from [77].

Consider a discrete-time perturbed nonlinear system:

$$x[k+1] = g(x[k], v[k]), \quad k \in \mathbb{Z}_+, \tag{5.19}$$

where $x \in \mathbb{R}^{n_x}$ is the state, $v \in \mathbb{R}^{n_v}$ is an unknown disturbance input and $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_x}$ is a nonlinear, possibly discontinuous function. For simplicity of notation, we assume that the origin is an equilibrium for (5.19) and zero disturbance input, meaning that $g(0,0) = 0$. We use the notation $x[k]$ to denote the solution of (5.19) at time $k \in \mathbb{Z}_+$, obtained from initial condition x_0 at time $k = 0$.

Remark 20. We mentioned the discrete-time perturbed nonlinear system because in [77] also use the same model to illustrate the result of the robust stability of discrete-time piecewise affine systems. Readers should notice that the piecewise affine system with input is a specific type of perturbed nonlinear system. ◁

A review of the definitions of Lyapunov Functions of discrete-time perturbed nonlinear systems is as follows:

Definition 5.1.5 (Lyapunov Functions)**.** [77, Def. II.6] Let $\mathbb{X} \subseteq \mathbb{R}^{n_x}$ be a positively invariant set for $x[k+1] = g(x[k]0)$ with $0 \in \text{int}(\mathbb{X})$, let $\alpha_1, \alpha_2, \alpha_3 \in \mathscr{K}_\infty$, let $V : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$, $V(0) = 0$, and consider the inequalities:

$$_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|), \qquad \forall x \in \mathbb{X}, \tag{5.20}$$
$$V(g(x,0)) - V(x) \leq 0, \qquad \forall x \in \mathbb{X}, \tag{5.21}$$
$$V(g(x,0)) - V(x) < 0, \qquad \forall x \in \mathbb{X} \setminus \{0\}, \tag{5.22}$$
$$V(g(x,0)) - V(x) \leq -\alpha_3(\|x\|), \qquad \forall x \in \mathbb{X}. \tag{5.23}$$
$$\tag{5.24}$$

A function $V(\cdot)$ that satisfies (5.20) and is called a *Lyapunov function.* A function $V(\cdot)$ that satisfies (5.20) and is called a *strict Lyapunov (SL) function.* A function $V(\cdot)$ that satisfies (5.20) and is called a *uniformly strict Lyapunov (USL) function.*

Remark 21. The common quadratic Lyapunov Function in theorem 5 is a continuous USL function. [2]. ◁

Consider the following system:

$$x[k+1] = h(x_k), \quad k \in \mathbb{Z}_+, \tag{5.25}$$

where $h : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ with $h(0) = 0$ is an arbitrary nonlinear, possibly discontinuous function. For a continuous function $\delta : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ we define a perturbed version of (5.25) as follows:

$$x[k+1] \in h_\delta(x_k) := \left\{ h(x_k + \delta(x_k)\, v) + \delta(x_k)\, v \mid v \in \mathscr{B} \right\}, \quad k \in \mathbb{Z}_+. \tag{5.26}$$

Let $\mathscr{S}_\delta(x_0)$ denote the set of all solutions of (5.26) corresponding to initial state x_0 at time $k = 0$.

The definition of the robust globally asymptotic stability (R.G.A.S) of (5.25) is then as follows:

Definition 5.1.6. (robust globally asymptotic stability (R.G.A.S)) [77, II.7] We call system (5.25) *robust globally asymptotic stability (R.G.A.S)* if there exist a $\delta : \mathbb{R}^{n_x} \to \mathbb{R}_{\geq 0}$ continuous and positive definite function and a $\beta_\delta \in \mathscr{K}\mathscr{L}$ such that for every $x_0 \in \mathbb{R}^{n_x}$ and all solutions $x_k^\delta \in \mathscr{S}_\delta(x_0)$ it holds that $\left\|x_k^\delta\right\| \leq \beta_\delta(\|x_0\|, k)$, for all $k \in \mathbb{Z}_{\geq 0}$.

The following proposition can be summarized from [77]

Proposition 5.1.2. *[77] The following two statements are equivalent:*

1. *System* (5.25) *admits a continuous USL function*
2. *System* (5.25) *is R.G.A.S.* □

From proposition 5.1.2, we can derive the following proposition:

[2] The result is direct from another form to describe USL in [77]

Proposition 5.1.3. *The following two statements hold:*

1. *System 5.6 is R.G.A.S if it admits a continuous USL function when* $B_i = D_i = 0 \forall i \in \{1, 2, \cdots, s\}$
2. *System 5.6 is R.G.A.S if it admits a common quadratic Lyapunov function.* □

Proof. The proof of this position is quite straightforward. The first part hold because PWA systems are a specific type of nonlinear systems, so the property hold for nonlinear systems also holds for PWA systems. The second part is a direct conclusion from Remark 21 and the first part of the proposition. ■

5.1.3. Discrete-time Switching Affine Systems Identification

Different research has been proposed to identify discrete-time switching affine systems. This section will summarize the results related to the identification problem of discrete-time switching affine systems, especially for discrete-time piecewise affine systems (PWA) and discrete-time arbitrarily switching affine systems (aSAS). The summarized results are mainly from [68].

Identification Problem Formulation

The problem of identifying a general hybrid system with unknown modes can be formulated as follows:

Problem 4 (Identify a general hybrid system with unknown modes)**.** *[68, Prob.4.1.] Given a data set* $\mathscr{D} = \{(x[k], y[k])\}_{k=1}^{N}$*, generated by a switched system, estimate the number of submodels* s*, the submodels* $\{f_j \in \mathscr{F}_j\}_{j=1}^{s}$*, and the switching sequence* $\boldsymbol{q} = \{q[k]\}_{k=1}^{N} \in [s]^N$*.* ◁

Assume that the number of submodels is fixed. Then we can formalize the problem of identifying discrete-time PWA and discrete-time aSAS as follows:

Problem 5 (Identify an aSAS with a fixed number of submodels)**.** *[68, Prob.4.2] Given a data set* $\mathscr{D} = \{(x[k], y[k])\}_{k=1}^{N}$ *and a bound* ϵ *on the mean loss, estimate the minimal number of submodels* $\{f_j\}_{j=1}^{s}$ *and the switching sequence needed to achieve that bound:*

$$\begin{aligned} &\min_{s\in\mathbb{N},\{f_j\in\mathscr{F}\},\boldsymbol{q}\in\mathbb{N}^N} s \\ &s.t.\ \boldsymbol{q} \in [s]^N \\ &\frac{1}{N}\sum_{k=1}^{N} \ell\left(y[k] - f_{q[k]}(x[k])\right) \le \epsilon. \end{aligned} \tag{5.27}$$

◁

Problem 6 (Identify a PWA with a fixed number of submodels)**.** *[68, Prob.4.3] Given a data set* $\mathscr{D} = \{(x[k], y[k])\}_{k=1}^{N}$ *and a number of submodels* s*, estimate the submodels* $\{f_j\}_{j=1}^{s}$ *and the partitioning function* g *by minimizing the error:*

$$\min_{\{f_j\in\mathscr{F}_j\}_{j=1}^{s}, g\in\mathscr{G}} \frac{1}{N}\sum_{k=1}^{N} \ell\left(y[k] - f_{g(x_k)}(x[k])\right) \tag{5.28}$$

◁

Exact Methods for Hybrid System Identification and Hardness of Identification

Assume that the number of submodels is fixed. There are three methods to solving the problems 5 and 6.

Problem 7 (Switching linear regression with fixed submodels)**.** *[68, Prob.5.1] Given a data set* $\{(x[k], y[k])\}_{k=1}^{N} \subset \mathbb{R}^{n_x} \times \mathbb{R}^{n_y}$ *and a positive integer* s*, find a globally solution to*

$$\min_{\{\boldsymbol{\theta}_j\in\mathbb{R}^{n_x}\}_{j=1}^{1}, q\in[s]^N} \frac{1}{N}\sum_{k=1}^{N} \ell\left(y[k] - x[k]^\top \boldsymbol{\theta}_{q[k]}\right). \tag{5.29}$$

◁

Problem 8 (Bounded-error estimation with fixed submodels)**.** *[68, Prob.5.2] Given a data set* $\{(x[k], y[k])\}_{k=1}^{N} \subset \mathbb{R}^{n_x} \times \mathbb{R}^{n_y}$ *and* $\epsilon \geq 0$*, find a globally solution to*

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{n_x}} \sum_{k=1}^{N} \ell_{p,\epsilon}\left(y[k] - x[k]^\top \boldsymbol{\theta}\right). \tag{5.30}$$

◁

Problem 9 (PWA Regression with fixed submodels)**.** *[68, Prob.5.3] Given a data set* $\{(x[k], y[k])\}_{k=1}^{N} \subset \mathbb{R}^{n_x} \times \mathbb{R}^{n_y}$ *and a number of modes,* s*, find a globally solution to*

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{xd}, g \in \mathcal{G}} \frac{1}{N} \sum_{k=1}^{N} \sum_{j \in [s]} \mathbb{1}_{g(x[k])=j} \ell\left(y[k] - x[k]^\top \boldsymbol{\theta}_j\right) \tag{5.31}$$

◁

The following theorems show the hardness of problems 7, 8 and 9.

Theorem 8. *[68, Theo.5.1] With a loss function* ℓ *such that*

$$\begin{cases} \ell(0) = 0, \\ \forall e \in \mathbb{Q}, \ell(-e) = \ell(e), \\ \forall (e, e') \in \mathbb{Q}^2, \ell(e) < \ell(e') \Leftrightarrow |e| < |e'|, \end{cases} \tag{5.32}$$

Problem 7 is $\mathcal{NP}$*-hard.* □

Theorem 9. *[78, Theo.1] With a loss function* ℓ *such that* $\ell(e) = 0 \Leftrightarrow e = 0$*, Problem 9 is* $\mathcal{NP}$*-hard.* □

Theorem 10. *[68] With a loss function* ℓ *such that* $\ell(e) = 0 \Leftrightarrow e = 0$*, Problem 8 is* $\mathcal{NP}$*-hard.* □

Remark 22. Several important remarks can be concluded from [68]:

1. The modes s play a particular role in the complexity of 7, 8 and 9. The proofs of the $\mathcal{NP}$-hardness are using the smallest value $s = 2$, meaning a larger s will typically incur a larger complexity.
2. With a fixed dimension of models, the problems can be solved in polynomial time. But it only happens when the sample data meet some really easy distribution. ◁

Remark 23. In [68, Ch.5], analysis of the complexity of different bounded-error identification strategies for switched systems is conducted by restricting solutions to the set of rational numbers rather than the reals. ◁

Remark 24. The difference between the problem 6 and the problem 5 is that the problem 6 need to find out the partition of each region. So the method for solving problem 5 can be used to solve 6 by first determining the belongings of each data point and then computing the partition based on the belongings. ◁

To exactly identify a hybrid system, a certain number and types of data are needed to meet the *persistence of exciation* condition [79]. The latest result of the persistence of excitation for identifying switched linear systems is as follows:

Theorem 11. *[79, Thm.2] To ensure the regressors and corresponding membership indices are PE for the [switched linear] system, the minimum number of required samples is*

$$\frac{(n_\theta - 1)s^2 + (n_\theta + 1)s}{2}. \tag{5.33}$$

where n_θ *is the parameter dimension and* s *is the number of modes.* □

The hardness of the exact solution of switching affine system identification indicates the sample complexity of the identification means that we need some heuristic and approximate methods to identify them. In the following parts, we will introduce some of these solutions. We will divide them into two classes: the first is methods based on the IO form, and the second is methods based on the SS form.

Algorithm 1 Two-stage approach to PWA system identification

1: Estimate the modes $\{\hat{q}[k]\}_{k=1}^{N}$ (and $\hat{s} = \max_{k\in[N]} \hat{q}[k]$)
2: Estimate the classifier g from $\{(x[k], \hat{q}[k])\}_{k=1}^{N}$.
3: Estimate the submodels f_j from the local data sets $\{(x[k], y[k]) : \hat{q}[k] = j\}, j = 1, \ldots, \hat{s}$.

Identification based on the IO Form

[68] summarizes the research progress of identifying switched affine and PWA models. Readers can read it for more details; we will not introduce most of them in this thesis. The important thing for this paper is that the piecewise property can be heuristically utilized when identifying PWA models. Based on that, a two-stage approach is proposed for PWA system identification [68, Alg.8]:

Algorithm 1 shows that for the PWA model identification, a data clustering method can be used at the first stage to allocate data into different submodels, such as the statistical clustering method [80].

Identification based on the SS Form

The problem of identifying linear discrete-time state-space models can be formalized as follows:

Problem 10. *[78, Prob.8.1.] Given only an input-output data set $\{(u[k], y[k])\}_{k=1}^{N}$ generated by a system in the form (8.10), estimate the switching sequence $\{q[k]\}_{k=1}^{N}$, the number s of discrete states, i.e., the number of submodels, the model order $n_{\mathbf{x}}$, all of the system parameter matrices A_j, B_j, C_j, D_j, $j = 1, \ldots, s$, and states $\{\mathbf{x}_k\}$, and, if the model is piecewise linear, the regions Ω_j, $j = 1, \ldots, s$.*

Problem 10 can be transformed into Mixed-Integer Nonlinear Program as in Problem 5, so the hardness of exactly solve Problem 10 is also $\mathcal{NP}$-Hard. Different methods have been proposed to heuristically and approximately identify a SS form model. These methods can be classified into four categories:

1. A direct way is first to identify an IO form model and then transform it into an SS model.
2. Some methods assume there is an upper bound of the switching times in the collected data. Based on this assumption, the authors in [81, 82] propose transforming the identification problem to combining the least square problem and the binary integer problem.
3. Some methods try first to divide collected data into small sets and identify local models on these sets [83, 84]. In [83], the authors use a sliding window to divide small sequential sets, whose size should be smaller than the dwell time and large enough to support the subspace identification method. The authors then identify local models on each small set using the subspace identification method, cluster the local models, classify the original data again, and repeat the identification process. The authors in [84] propose a change detection method to detect the submodel change in the collected data to generate the small data sets. Both [83] and [84] assume a minimal dwell time τ between two consecutive switches.
4. In [85], the author proposes a method that doesn't require the minimum dwell time. The method identifies SS models by constructing structure intermediary matrices, iterating between data classification and parameter updating.

These methods have two main drawbacks:

D 16.1 Most of these methods require a minimum dwell time [83, 84] or upper bound of the switching times [81, 82]. The requirement makes them not applicable to fast switching systems [68].

D 16.2 Although the method in [85] does not require minimum dwell time, it needs the system to be pathwise observable.

5.2. Design of HMWM

In this section, we give an overview of the design of the multiplicative watermarking scheme in (3.10) as composed of hybrid systems with piecewise affine (PWA) dynamics. This proves to be beneficial when analyzing the method's resilience to attacks defined in Section 3.2, as shown in Section 5.3

5.2.1. HMWM Structure

We propose a design strategy that defines the dynamics of $\mathscr{W}$ and $\mathscr{Q}$ as piecewise affine (PWA) linear switched systems. More precisely, the dynamics of $\mathscr{W}$ are[3]:

$$\mathscr{W}: \begin{cases} x_w[k+1] = \sum_{i=0}^{N} \beta_{w,i} \left(A_{w,i} x_w[k] + B_{w,i} \left(y_p[k] + Hu'[k] \right) \right) \\ y_w[k] = \sum_{i=0}^{N} \beta_{w,i} \left(C_{w,i} x_w[k] + D_{w,i} \left(y_p[k] + Hu'[k] \right) \right) \end{cases} \tag{5.34}$$
$$\mathscr{F}: \theta_w[k] = \sum_{i=0}^{N} \beta_{w,i} \theta_i \,, \; \beta_{w,i} = \begin{cases} 1, \text{ if } x_{w,u}[k] \in \mathscr{P}_i \\ 0, \text{ otherwise} \end{cases}$$

where subscript $i \in \mathscr{N} = \{1, \dots, N\}$ indicates one of N *modes* of operation, and the boolean variables $\beta_{w,i}[k] \in \{0,1\}, \forall i \in \mathscr{N}$ are used to determining which mode is active at any given time. In the definition of the switching function, we rely on the evaluation of a logical rule, namely the evaluation of $x_{w,u} \in \mathbb{P}_i$. Here, $x_{w,u} \in \mathbb{R}^{n_u}$, defined in the following, is a portion of the state x_w which is unobservable from y_w, and $\mathscr{P}_i \subset \mathbb{R}^{n_u}, i \in \mathscr{N}$ are polyhedrons, which are defined such that $\bigcup_{i \in \mathscr{N}} \mathscr{P}_i = \mathbb{R}^{n_u}$. An example of such a partition can be seen in Figure 6.1a. In order to guarantee that $\sum_{i \in \mathscr{N}} \beta_{w,i} = 1$, a necessary condition for (5.34) to be a PWA linear switching system, $\mathscr{P}_i \cap \mathscr{P}_j = \emptyset$ must hold for all $i, j \in \mathscr{N}, i \neq j$. Given that $\mathscr{P}_i$ are closed sets, this does not hold for adjacent partitions; therefore, an additional logical law must be applied. Specifically, we impose that, if $x_{w,u}[k] \in \mathscr{P}_i \cap \mathscr{P}_j, i, j \in \mathscr{N}, i \neq j$, $\beta_{w,i} = 1$ if and only if $i < j$. This guarantees that *in practice*, the partitioning of $\mathbb{R}^{n_u}$ is non-overlapping.

Matrices $A_{w,i}, B_{w,i}, C_{w,i}, D_{w,i}$ are defined as follows:

$$A_{w,i} = \begin{bmatrix} A^-_{w,i} & 0 \\ 0 & A_{w,u} \end{bmatrix}, \quad B_{w,i} = \begin{bmatrix} B^-_{w,i} \\ B_{w,u} \end{bmatrix}, \\ C_{w,i} = \begin{bmatrix} C^-_{w,i} & 0 \end{bmatrix}, \quad D_{w,i} = D^-_{w,i}. \tag{5.35}$$

Note that, given (5.35), the resulting systems are unobservable by definition, with $x_{w,u} \in \mathbb{R}^{n_u}$ the unobservable portion of the state $x_w = \left[x_{w,o}^\top, x_{w,u}^\top \right]^\top$. In (5.35), $A_w = \text{diag}(a_{w,1}, \dots, a_{w,n_u})$ and $B_{w,u} \in \mathbb{R}^{n_u \times p}$ are common to all modes, with $|a_{w,j}| \leq \sqrt{0.5}, \forall j \in \{1, \dots, n_u\}$; the matrices $A^-_{w,i}$ are defined as $A^-_{w,i} \triangleq \bar{T}^\top \bar{A}^-_{w,i} \bar{T}$, where $\bar{T}$ is an orthogonal matrix common to all modes, while $\bar{A}^-_{w,i}$ are randomly defined, stable diagonal matrix. The matrices $B^-_{w,i}$ are defined randomly, such that $(A^-_{w,i}, B^-_{w,i})$ is stable. Finally, $C^-_{w,i}$ and $D^-_{w,i}$ are defined as follows: firstly, a matrix K_i stabilizing $A^-_{w,i} - B^-_{w,i} K_i$ is found satisfying

$$\begin{bmatrix} X & A^-_{w,i} X + B^-_{w,i} Z_i \\ (A^-_{w,i} X + B^-_{w,i} Z_i)^\top & X \end{bmatrix} \succ 0 \\ X \succ 0; \quad K_i = -Z_i X^{-1} \quad \forall i \in \mathscr{N}, \tag{5.36}$$

then $D^-_{w,i}$ is defined randomly to be a square and invertible matrix, and $C^-_{w,i} = D_{w,i} K_i$. The procedure for designing the watermarking matrices is summarized in Algorithm 2. In the following, we prove that this design procedure satisfies Problem 3.2.a. and Problem 3.2. b..

5.2.2. Generator and Remover Stability

To prove closed-loop stability and ISS stability, we exploit the definition of globally uniformly asymptotic stability (GUAS), using a common quadratic Lyapunov function.

Theorem 12. *Given a watermark generator and remover pair $(\mathscr{W}, \mathscr{Q})$ with dynamics as in (3.11), if their system matrices are generated following Algorithm 2, with $n_u = 1$, the systems are GUAS and input-state stable (ISS) under arbitrary switching. Furthermore, $\mathscr{Q}(\theta_i) = \mathscr{W}(\theta_i)^{-1}, \forall i \in \mathscr{N}$.* □

Proof. To prove that $\mathscr{W}$ is GUAS stable, it is sufficient to show that the autonomous systems

$$x_w[k+1] = A_{w,i} x_w[k] \tag{5.37}$$

[3] The dynamics of $\mathscr{Q}$ are analogous to (5.34), substituting subscript w with q, and changing the input from $y_p[k]$ to $y_w[k]$, and defining the system matrices following (3.11).

Algorithm 2 Generate GUAS $\mathscr{W}$ and $\mathscr{Q}$

Input: $n_w \geq 1, N \geq 1$
Output: $\theta_i, i \in \mathscr{N}$

1: Randomly generate diagonal matrices $\bar{A}^-_{w,i}$, $i \in \mathscr{N}$, such that $\rho(\bar{A}^-_{w,i}) < 1$, and an orthonormal matrix $\bar{T}$
2: Define $A^-_{w,i} = \bar{T}^\top \bar{A}^-_{w,i} \bar{T}$.
3: Randomly generate $B^-_{w,i}$ such that $(A^-_{w,i}, B^-_{w,i})$ are controllable;
4: Design K_i such that (5.36) is jointly satisfied for all $i \in \mathscr{N}$;
5: Randomly generate $D_{w,i}$ and define $C_{w,i} = D_{w,i} K_i$.
6: Randomly generate $a_w \in \mathbb{R} \quad \& \quad |a_w| \leq \sqrt{0.5}, b_w^\top \in \mathbb{R}^p$, set $A_{w,u} = a_w$ and define $A_{w,i}, B_{w,i}, C_{w,i}, D_{w,i}$ solving (5.35).
7: Define $A_{q,i}$, $B_{q,i}$, $C_{q,i}$, $D_{q,i}$, corresponding to $A_{w,i}$, $B_{w,i}$, $C_{w,i}$, $D_{w,i}$, solving (3.11);
8: **if** $n_w > 1$, **for** $t = 2 : n_w$
9: Define:

$$A^-_{w,i} = A_{w,i}\,, \quad B^-_{w,i} = B_{w,i}\,,$$
$$C^-_{w,i} = C_{w,i}\,, \quad D^-_{w,i} = D_{w,i}\,;$$

10: Repeat Step 6
11: **endif endfor**

admit a common Lyapunov function for all $i \in \mathscr{N}$. We define a candidate Lyapunov function $V_w : \mathbb{R}^{n_w} \to \mathbb{R}$, $V_w(x) = x^\top P_w x$, where $P_w \succ 0, P_w \in \mathbb{S}^{n_w}$. Thus, it is sufficient that

$$A^\top_{w,i} P_w A_{w,i} - P_w \prec 0\,, \quad \forall\, i \in \mathscr{N}. \tag{5.38}$$

Note that, given the definition of $A_{w,i}$ in (5.35), a transformation $T = \mathrm{diag}(\bar{T}, I_{n_u})$ can be defined such that $A_{w,i} = T^\top \bar{A}_{w,i} T$, with $\bar{A}_{w,i} = \mathrm{diag}(\bar{A}^-_{w,i}, A_{w,u})$ and T common to all modes. We therefore rewrite (5.38) as:

$$T^\top \bar{A}_{w,i} T P_w T^\top \bar{A}^-_{w,i} T - P_w \prec 0\,, \quad \forall\, i \in \mathscr{N}. \tag{5.39}$$

We now pre- and post-multiply (5.39) by T and $T^\top$, respectively, and define $\bar{P}_w = T P_w T^\top \in \mathbb{S}^{n_w}$. Note that, because $P_w \succ 0$, $\bar{P}_w \succ 0$ as well. Thus, if

$$\bar{A}^\top_{w,i} \bar{P}_w \bar{A}_{w,i} - \bar{P}_w \prec 0 \tag{5.40}$$

holds, so does (5.38). Given $\bar{A}_{w,i}, \forall\, i \in \mathscr{N}$ by design is a diagonal matrix with $\rho(\bar{A}_{w,i}) < 1$, there exists a positive definite $\bar{P}_w$ such that (5.40) holds for all $i \in \mathscr{N}$. This proves that $\mathscr{W}$ is GUAS under arbitrary switching, including the switching function in (5.34).

Similarly, we prove $\mathscr{Q}$ is GUAS, by supposing that there exists a symmetric $P_q \succ 0$ such that the candidate Lyapunov function $V_q : \mathbb{R}^{n_w} \to \mathbb{R}$, $V_q(x) = x^\top P_q x$ is suitable for all modes $i \in \mathscr{N}$. We prove this by construction. Firstly, note that given definition of K_i in (5.36), each $A^-_{q,i} = A^-_{w,i} - B^-_{w,i} D^{-1}_{w,i} C^-_{w,i} = A^-_{w,i} - B^-_{w,i} K_i$ is Schur stable, and $V^-_q : \mathbb{R}^{n_w - n_u} \to \mathbb{R}$, $V^-_q(x) = x^\top P^-_q x$ is a common Lyapunov function for all $i \in \mathscr{N}$, with $P^-_q = X^{-1}$, where $X \succ 0$ solves (5.36). Furthermore, from definition of $A_{q,i}$ in (3.11) and matrices in (5.35), $A_{q,i}$ can be written as:

$$\begin{aligned} A_{q,i} &= \begin{bmatrix} A^-_{w.i} - B^-_{w,i} D^{-1}_{w,i} C^-_{w,i} & 0 \\ -b_w D^{-1}_{w,i} C^-_{w,i} & a_w \end{bmatrix} \\ &\triangleq \begin{bmatrix} A_{q,i,1} & 0 \\ A_{q,i,3} & A_{q,i,4} \end{bmatrix} \end{aligned}, \tag{5.41}$$

with $a_w = A_{w,u} \in \mathbb{R}$, given $n_u = 1$ by assumption. Let us now introduce $p_q > 0, p_q \in \mathbb{R}$, and define $P_q = \mathrm{diag}(P^-_q, p_q)$. For V_q to be an appropriate Lyapunov function, it is sufficient that

$$A^\top_{q,i} P_q A_{q,i} - P_q \prec 0 \tag{5.42}$$

holds for all $i \in \mathscr{N}$. By considering the decomposition of $A_{q,i}$ defined in (5.41), we rewrite (5.42) as:

$$\begin{bmatrix} \Phi & \Xi \\ \Xi^\top & \Psi \end{bmatrix} \prec 0, \tag{5.43}$$

where

$$\Phi \triangleq A_{q,i,1}^{\top} P_q^{-} A_{q,i,1} - P_q^{-} + A_{q,i,3}^{\top} p_q A_{q,i,3}$$
$$\Psi \triangleq A_{q,i,4}^{\top} p_q A_{q,i,4} - p_q$$
$$\Xi \triangleq A_{q,i,3} p_q A_{q,i,4}$$

Thus, by applying the Schur complement, (5.42) holds iff

$$\Psi \prec 0; \tag{5.44a}$$
$$\Phi - \Xi \Psi^{-1} \Xi^{\top} \prec 0. \tag{5.44b}$$

By substituting the definition of Ψ and $A_{q,i,4}$, (5.44a) is equivalent to $(1-a_w^2) < 0$, and therefore holds if and only if $|a_w| < 1$, which holds by construction. Furthermore, the design procedure in Algorithm 2 guarantees that (5.44b) holds: indeed, after some algebraic manipulations, recalling that $a_w, p_q \in \mathbb{R}$, we rewrite (5.44b) as:

$$P_q^{-} - A_{q,i}^{-\top} P_q^{-} A_{q,i} - A_{q,i,3}^{\top} A_{q,i,3} \frac{1-2a_w^2}{1-a_w^2} p_q \prec 0. \tag{5.45}$$

Therefore, given that $(1-a_w^2) < 0$, it is necessary for $(1-2a_w^2) \geq 0$, which holds if $|a_w| \leq \sqrt{0.5}$, corresponding to the constraint set in Step 6 in Algorithm 2. This completes the proof that $\mathscr{Q}$ is GUAS.

Furthermore, $\mathscr{W}$ and $\mathscr{Q}$ are ISS, as V_w and V_q are contiunous uniform strict Lyapunov functions [77].

Finally, we complete the proof by pointing out that $\mathscr{Q}(\theta_i) = \mathscr{W}(\theta_i)^{-1}, \forall i \in \mathscr{N}$ holds by construction, via Step 7 in Algorithm 2. ■

Corollary 1. *The results of Theorem 12 hold for $n_u > 1$.* □

Proof. Suppose that $n_u = 2$. Following the procedure indicated in Step 8 in Algorithm 2, note that the value taken by $A_{w,i}^{-}$ in this proof is the same as that taken by $A_{w,i}$ in the proof of Theorem 12. The same goes for all other matrices. Because $A_{w,i}$ is block diagonal, it is sufficient to define a new positive definite matrix $P_w = \text{diag}(P_w^{-}, p_w) \succ 0$, where $P_w^{-} \succ 0$ is the matrix defining the Lyapunov function for $\mathscr{W}$ in the proof of Theorem 12, and $p_w > 0, p_w \in \mathbb{R}$. Given that $|a_w| < \sqrt{0.5} < 1$ by its definition in Step 6,

$$A_{w,i}^{\top} P_w A_{w,i} - P_w \prec 0$$

holds for any $p_w > 0$. Thus $\mathscr{W}$ GUAS and ISS under arbitrary switching, for $n_u = 2$. On the other hand, following the same reasoning in the proof of Theorem 12, given $|a_{w,2}| < \sqrt{0.5}$, GUAS and ISS of $\mathscr{Q}$ is proven. The proof for $n_u > 2$ follows by induction and recursive definition of $A_{w,u}$ in Algorithm 2. ■

5.2.3. Detectability Design

In this section, several theorems are proposed and proved to give the design guideline of the parameters of the HMWM Theory coding method. The detectability design of parameters is very complex under the HMWM structure for replay attacks and sensor-signal-injection zero-dynamics attacks. For the replay attack, we will only provide detection performance theorem proof for a simplified version that the HMWM parameter only has a single set of parameters.

Replay Attack

Theorem 13. *Assume there is only a single set of multiplicative watermarking parameters. If the coding matrix H is designed to make $\Phi_H = (A_p - B_p K - LC_p + LHK)$ has at least one unstable eigenvalue and $(C_p + HK, \Phi_H)$ is made observable, the replay attack will be detected.* □

Proof. Assume $\Delta x_q[k+1] = x_q[k+1] - x_q[k+1-\Delta k]$ and $\Delta y_q[k] = y_q[k] - y_q[k-\Delta k]$

$$\begin{aligned} \Delta x_q[k+1] &= A_q x_q[k] + B_q y_w[k] - A_q x_q[k-\Delta k] - B_q\, y_w[k-\Delta k] \\ &= A_q \Delta x_q[k] \\ \Delta y_q[k] &= C_q x_q[k] + D_q y_w[k] - C_q x_q[k-\Delta k] - D_q y_w[k-\Delta k] - H(u[k] - u[k-\Delta k]) \\ &= C_q \Delta x_q[k] - H(u[k] - u[k-\Delta k]) \end{aligned} \tag{5.46}$$

Then we can expand $r[k]$ as follows:

$$\begin{aligned} r[k] &= y_q[k] - \hat{y}[k] = (y_q[k] - y_q[k-\Delta k]) - (\hat{y}[k] - y_q[k-\Delta k]) \\ &= C_q \Delta x_q[k] - H(u[k] - u[k-\Delta k]) - \big(\hat{y}[k] - \hat{y}[k-\Delta k] + \hat{y}[k-\Delta k] - y_q[k-\Delta k]\big) \\ &= C_q \Delta x_q[k] - H(u[k] - u[k-\Delta k]) + r[k-\Delta k] - \big(\hat{y}[k] - \hat{y}[k-\Delta k]\big) \\ &= r[k-\Delta k] + C_q \Delta x_q[k] - (HK + C_p)\big(\hat{x}_p[k] - \hat{x}_p[k-\Delta k]\big) \end{aligned} \tag{5.47}$$

The dynamics of $\hat{x}_p[k]$ can be written to:

$$\begin{aligned} \hat{x}_p[k] &= A_p \hat{x}_p[k-1] + B_p u[k-1] + L\big(y_q[k-1] - \hat{y}_p[k-1]\big) \\ &= (A_p + B_p K)\hat{x}_p[k-1] - LC_p \hat{x}_p[k-1] + L y_q[k-1] \\ &= (A_p + B_p K - LC_p)\hat{x}_p[k-1] + L y_q[k-1] \end{aligned} \tag{5.48}$$

Assume $\Delta \hat{x}_p[k] = \hat{x}_p[k] - \hat{x}_p[k-\Delta K]$, we have:

$$\begin{aligned} \Delta \hat{x}_p[k] &= (A_p + B_p K - LC_p)\Delta \hat{x}_p[k-1] + L(y_q[k-1] - y_q[k-\Delta k-1]) \\ &= (A_p + B_p K - LC_p)\Delta \hat{x}_p[k-1] + L(\Delta y_q[k-1]) \\ &= \big((A_p + B_p K - LC_p) + LHK\big)\Delta \hat{x}_p[k-1] + LC_q \Delta x_q[k-1] \end{aligned} \tag{5.49}$$

Then we can return to the dynamics of $r[k]$ as follows:

$$\begin{aligned} r[k] &= r[k-\Delta k] + C_q \Delta x_q[k] - HK\big(x_p[k] - \hat{x}_p[k-\Delta k]\big) - C_p\big(\hat{x}_p[k] - \hat{x}_p[k-\Delta k]\big) \\ &= r[k-\Delta k] + C_q \Delta x_q[k] - (HK + C_p)\Delta \hat{x}_p[k] \\ &= r[k-\Delta k] + C_q \Delta x_q[k] - (HK + C_p)\big((A_p + B_p K - LC_p + LHK)\Delta \hat{x}_p[k-1] + LM[k-1]\big) \\ &= r[k-\Delta k] + \big(C_q A_q - (HK + C_p)LC_q\big)\Delta x_q[k-1] + (HK + C_p)(A_p + B_p K - LC_p + LHK)\Delta \hat{x}_p[k-1] \\ &= r[k-\Delta k] + \big(C_q A_q - (HK + C_p)LC_q\big) A_q^{k-1} \Delta x_q[0] + (HK + C_p)\underbrace{(A_p + B_p K - LC_p + LHK)}_{\Phi_H}\Delta \hat{x}_p[k-1] \end{aligned} \tag{5.50}$$

Because A_q is a stable matrix, then the term $\big(C_q A_q - (HK + C_p)LC_q\big) A_q^{k-1} \Delta x_q[0] \to 0$ as time goes by. Therefore, with an unstable Φ_H, the $r[k]$ will go to infinite.

■

Remark 25. The theorem 13 proves that when there is only a single set of multiplicative watermarking parameters. We can design a coding matrix H such that $\Phi_H = (A_p - B_p K - LC_p + LHK)$ has at least one unstable eigenvalue to detect a replay attack will be detected. This means, that by introducing suitable H, the multiplicative watermarking method will be able to detect replay attacks with probability 1 even without any switching happening. ◁

Remark 26. When there are multiple sets of parameters, the scenario will be much more complicated. The $m[k]$ term will be $m[k] = C_q(\theta_k)x_q^a[k] + D_q(\theta_k)y_w^a[k] - C_q(\theta_{k-\Delta k})x_q^a[k-\Delta k] - D_q(\theta_{k-\Delta k})y_w^a[k-\Delta k]$ under that scenario. The following proposition may propose some clues to the detection performance. ◁

Proposition 5.2.1. *When there are multiple sets of parameters, define $\Delta \hat{x}_p[k] = \hat{x}_p[k] - \hat{x}_p[k-\Delta K]$ and unstable $\Phi_H = A_p + B_p K - LC_p + LHK$, with $r[k] = y_p'[k] - \hat{y}_p[k]$ we have the following dynamics:*

$$\begin{aligned} \Delta \hat{x}_p[k+1] &= \Phi_H \Delta \hat{x}_p[k] + Lm[k] \\ r[k] &= -(HK + C_p)\Delta \hat{x}_p[k] + m[k] \end{aligned} \tag{5.51}$$

when Φ_H is unstable and the $Lm[k]$ does not stabilize the unstable dynamics. □

Proof. Assume $\Delta x_q[k+1] = x_q[k+1] - x_q[k+1-\Delta k]$ and $\Delta y_q[k] = y_q[k] - y_q[k-\Delta k]$

$$\begin{aligned} \Delta x_q[k+1] &= A_q(\theta_k)x_q[k] + B_q(\theta_k)y_w[k] - A_q(\theta_{k-\Delta k})x_q[k-\Delta k] - B_q(\theta_{k-\Delta k})y_w[k-\Delta k] \\ \Delta y_q[k] &= \underbrace{C_q(\theta_k)x_q[k] + D_q(\theta_k)y_w[k] - C_q(\theta_{k-\Delta k})x_q[k-\Delta k] - D_q(\theta_{k-\Delta k})y_w[k-\Delta k]}_{m[k]} - H(u[k] - u[k-\Delta k]) \end{aligned} \tag{5.52}$$

Then we can expand $r[k]$ as follows:

$$\begin{aligned} r[k] &= y_q[k] - \hat{y}[k] = (y_q[k] - y_q[k-\Delta k]) - (\hat{y}[k] - y_q[k-\Delta k]) \\ &= m[k] - H(u[k] - u[k-\Delta k]) - \big(\hat{y}[k] - \hat{y}[k-\Delta k] + \hat{y}[k-\Delta k] - y_q[k-\Delta k]\big) \\ &= m[k] - H(u[k] - u[k-\Delta k]) + r[k-\Delta k] - \big(\hat{y}[k] - \hat{y}[k-\Delta k]\big) \\ &= r[k-\Delta k] + m[k] - (HK + C_p)\big(\hat{x}_p[k] - \hat{x}_p[k-\Delta k]\big) \end{aligned} \tag{5.53}$$

The dynamics of $\hat{x}_p[k]$ can be written to:

$$\begin{aligned} \hat{x}_p[k] &= A_p\hat{x}_p[k-1] + B_p u[k-1] + L\big(y_q[k-1] - \hat{y}_p[k-1]\big) \\ &= (A_p + B_pK)\hat{x}_p[k-1] - LC_p\hat{x}_p[k-1] + Ly_q[k-1] \\ &= (A_p + B_pK - LC_p)\hat{x}_p[k-1] + Ly_q[k-1] \end{aligned} \tag{5.54}$$

Assume $\Delta\hat{x}_p[k] = \hat{x}_p[k] - \hat{x}_p[k-\Delta K]$, we have:

$$\begin{aligned} \Delta\hat{x}_p[k] &= (A_p + B_pK - LC_p)\Delta\hat{x}_p[k-1] + L(y_q[k-1] - y_q[k-\Delta k-1]) \\ &= (A_p + B_pK - LC_p)\Delta\hat{x}_p[k-1] + L(\Delta y_q[k-1]) \\ &= \big((A_p + B_pK - LC_p) + LHK\big)\Delta\hat{x}_p[k-1] + LM[k-1] \end{aligned} \tag{5.55}$$

Then we can return to the dynamics of $r[k]$ as follows:

$$\begin{aligned} r[k] &= m[k] - HK\big(x[k] - \hat{x}_p[k-\Delta k]\big) - C_p\big(\hat{x}_p[k] - \hat{x}_p[k-\Delta k]\big) \\ &= m[k] - (HK + C_p)\Delta\hat{x}_p[k] \\ &= m[k] - (HK + C_p)\big((A_p + B_pK - LC_p + LHK)\Delta\hat{x}_p[k-1] + LM[k-1]\big) \\ &= \big(m[k] - (HK + C_p)LM[k-1]\big) + (HK + C_p)\underbrace{(A_p + B_pK - LC_p + LHK)}_{\Phi_H}\Delta\hat{x}_p[k-1] \end{aligned} \tag{5.56}$$

So the joint dynamics of $r[k]$ and $\Delta\hat{x}_p[k]$ will then become:

$$\begin{aligned} \Delta\hat{x}_p[k+1] &= \Phi_H\Delta\hat{x}_p[k] + Lm[k] \\ r[k] &= -(HK + C_p)\Delta\hat{x}_p[k] + m[k] \end{aligned} \tag{5.57}$$

$m[k]$ can be further expanded as follows:

$$\begin{aligned} m[k] &= C_q(\theta_k)x_q[k] + D_q(\theta_k)y_w[k] - C_q(\theta_{k-\Delta k})x_q[k-\Delta k] - D_q(\theta_{k-\Delta k})y_w[k-\Delta k] \\ &= C_q(\theta_k)x_q[k] - C_q(\theta_{k-\Delta k})x_q[k-\Delta k] + (D_q(\theta_k) - D_q(\theta_{k-\Delta k}))y_w[k-\Delta k] \end{aligned} \tag{5.58}$$

The $x_q[k]$ term related to $x_q[k-1], x_w[k-1], x_p[k-1]$ and $\hat{x}_p[k-1]$. The $x_q[k-\Delta k]$ and $y_w[k-\Delta k]$ terms only relate to state or variables prior to timestamp $k-\Delta k$. This means, in $LM[k-1]$ there is no feedback relationship related to the $\hat{x}_p[k]$ term in $\Delta\hat{x}_p[k]$. Then the $LM[k-1]$ term in equation (5.58) cannot provide a feedback pole placement to Φ_H, which makes equation (5.58) an unstable dynamics.

■

Remark 27. Proposition 5.2.1 shows that with unstable Φ_H we can also build unstable dynamics and the $r[k]$ is the output of the dynamics. However, because the $m[k]$ part is very complex, there are some special cases in which $m[k]$ may partly compensate for the unstable dynamics. ◁

Control-Signal-Injection Zero Dynamics Attack

For control-signal-injection zero-dynamics attack, similar to the theorem 2, the following condition holds:

Theorem 14. *The original control-signal-injection zero-dynamics attack, i.e. with the g value meeting equation (2.3), will keep being a zero-dynamics attack if and only if there exists $x_1 \in \mathbb{R}^n$, such that:*

$$\begin{bmatrix} \nu I - A_p & -B_p \\ C_p & H \end{bmatrix}\begin{bmatrix} x_1 \\ g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \exists x_1 \in \mathbb{R}^n \tag{5.59}$$

□

Proof. For a control-signal-injection zero-dynamics attack, the attack will inject an attack sequence as follows:

$$\begin{bmatrix} vI - A_p & -B_p \\ C_p & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$\Delta u^a[k] = v^k g \tag{5.60}$$

With the HMWM, After including $Hu[k]$, define $e[k] = x_p[k] - \hat{x}_p[k]$, the dynamics of $e[k], r[k]$ will change to:

$$\begin{aligned} e[k+1] &= x_p[k+1] - \hat{x}_p[k+1] \\ &= A_p x_p[k] + B_p u'[k] + w[k] - \left[A_p \hat{x}_p[k] + B_p u[k] + L(y'_p[k] - \hat{y}_p[k])\right] \\ &= A_p(x_p[k] - \hat{x}_p[k]) + B_p \Delta u[k] + w[k] - L\left(C_p x[k] + v[k] + H\Delta u[k] - C_p \hat{x}_p[k]\right) \\ &= (A_p - LC_p)e[k] + (B_p - LH)\Delta u[k] + (w[k] - Lv[k]) \\ r[k] &= y'_p[k] - \hat{y}_p[k] \\ &= C_p e[k] + v[k] + H\Delta u[k] \end{aligned} \tag{5.61}$$

So now we have

$$\begin{aligned} e[k+1] &= (A_p - LC_p)e[k] + (B_p - LH)\Delta u[k] + (w[k] - Lv[k]) \\ r[k] &= C_p e[k] + v[k] + H\Delta u[k] \end{aligned} \tag{5.62}$$

If the attack sequence $\Delta u^a[k] = v^k g$ still wants to be a malicious zero-dynamics attack sequence, it should meet the following condition:

$$\begin{bmatrix} vI - (A_p - LC_p) & LH - B_p \\ C_p & H \end{bmatrix} \begin{bmatrix} x_1 \\ g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \exists x_1 \in \mathbb{R}^n \tag{5.63}$$

The condition equals to:

$$\begin{bmatrix} vI - A_p & -B_p \\ C_p & H \end{bmatrix} \begin{bmatrix} x_1 \\ g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \exists x_1 \in \mathbb{R}^n \tag{5.64}$$

This is the same as equation (5.63) and finishes the proof.

■

Proposition 5.2.2. *If $g \notin \ker(H)$, equation 5.63 can only be met if:*

1. *v is both a transmission error of the system and an eigenvalue of A_p.*
2. *Assume $\mathbb{E}_v$ is the eigenspace of A_p that corresponding the eigenvalue of v. Then there exists $x \in \mathbb{E}_v$, such that $Hg = C_p x$.* □

Proof. The theorem 2 and 14 have the same structure and the same condition. The proof of this proposition is the same as the proof of proposition 4.1.2 ■

sensor-signal-injection zero-dynamics attack

The residual dynamic under the sensor-signal-injection zero-dynamics attack is very complex. So far, this residual does not have a similar result as in Theorem 3, Theorem 13 or Theorem 14.

In [24], the authors show the sensor-data-injection zero-dynamics attack for the original system will cause an extra residual on a system with multiplicative watermarking when the generator and remover have matched parameters. In [30], the authors further show that the sensor-data-injection zero-dynamics attack will cause extra residual when the explicit switching happens. Intuitively, because hybrid multiplicative watermarking switches fast and the $Hu[k]$ part will itself cause extra residual, the hybrid multiplicative watermarking methods will also be able to detect sensor-data-injection zero-dynamics attacks.

5.2.4. Switching Rule Design

The switching rule in Section 5.2.1 meets the requirements outlined in Section 3.4.2. Indeed, **R1** is met because, although exact quantification of the dwell time between switching events is challenging, the boundaries of each region $\mathscr{P}_i$ can be defined such that the probability of $x_{w,u}[k] \in \mathscr{P}_i$ is uniform across all $i \in \mathscr{N}$, given knowledge of the probability distributions of $w[k]$ and $v[k]$; **R2** is met, as the switching can be seen as being "truly random": the dynamics of $x_{w,u}$ depend on w and v, which are the result of physical processes,

and are not generated by a pseudo-random number generator[4]. Therefore, it is not possible to define the trajectory of $x_{w,u}[k]$ *a priori*. Finally, **R3** holds, as State and parameter synchronization is proven in Proposition 5.2.3.

Proposition 5.2.3. *Suppose a CPS as in (3.1)-(3.2) is equipped with the HMWM scheme (5.34). If $x_w[0] = x_q[0]$, and $\mathscr{W}$ and $\mathscr{Q}$ share the same $\mathscr{P}_i, \forall i \in \mathscr{N}$, then $\theta_w[k] = \theta_q[k], \forall k \geq 0$.* □

Proof. Let us start this proof by supposing that, at some time $k = \kappa$, $x_w[\kappa] = x_q[\kappa]$; thus, by definition of the hybrid multiplicative watermarking scheme in (5.34), $\theta_w[\kappa] = \theta_q[\kappa]$, as $x_w[\kappa] = x_q[\kappa] \in \mathscr{P}_i$. Dropping explicit dependence on the watermarking parameters, as they are matched, we write the one time-step difference equation of $x_{wq} = x_w - x_q$:

$$\begin{aligned} x_{wq}[\kappa+1] &\overset{(a)}{=} (A_{w,i} - B_{q,i}C_{q,i})x_w[\kappa] - A_{q,i}x_q[\kappa] \\ &\quad + (B_{w,i} - B_{q,i}D_{w,i})y_p[\kappa] \\ &\overset{(b)}{=} A_{q,i}x_{wq}[\kappa] \end{aligned} \tag{5.65}$$

where (a) holds by definition of the dynamics of x_w and x_q, and (b) holds by definition of the watermarking system matrices (3.11). Thus, $x_{wq}[\kappa+1] = x_{wq}[\kappa] = 0$, which in turn implies that $x_w[\kappa+1] = x_q[\kappa+1]$, and that $\theta_w[\kappa+1] = \theta_q[\kappa+1]$. The proposition's statement then holds by induction. ■

Remark 28. Let us note here that the switching law we present in this paper is different to the one presented in [30] in one fundamental aspect. Indeed, here the switching law at time k depends on information available in $\mathscr{I}_w[k-1]$ and $\mathscr{I}_q[k-1]$. Instead, it [30], the authors propose an event-triggered switching law, and the watermark remover must first *decode* $y_w[k]$, then evaluate whether there has been a parameter jump in the watermark generator, and if that is the case, update its own parameters and recompute $y_q[k]$. ◁

Proposition 5.2.4. *The closed-loop of the CPS with watermarking pair $(\mathscr{W}, \mathscr{Q})$ designed following Algorithm 2 is stable, and its performance remains unchanged if $x_w[0] = x_q[0]$.* □

Proof. The proof follows from the fact that, if designed following Algorithm 2, $(\mathscr{W}, \mathscr{Q})$ satisfy Definition 3.3.1, as shown in Theorem 12, and their parameters match for all $k \in \mathbb{Z}_+$, as proven in Proposition 5.2.3. ■

Example Design of Switching Region

Let us now propose a possible definition of the non-overlapping partitions $\mathscr{P}_i, i \in \mathscr{N}$. Specifically, we propose a partitioning of $\mathbb{R}^{n_u}$ such that, when the system reaches a steady state, the probability of $x_{w,u}[k] \in \mathscr{P}_i$, at any k, is uniform across $i \in \mathscr{N}$. Let us start by characterizing the statistical properties of $x_{w,u}[k] \sim \mathscr{N}(\mu_{x_{w,u}}[k], \Sigma_{x_{w,u}}[k])$ from the following process.

We define $e[k+1] \triangleq x_p[k+1] - \hat{x}_p[k+1]$. When no attacks happen, we have $y_q[k] = y_p[k]$, the $e[k+1]$ can be expanded as follows:

$$\begin{aligned} e[k+1] &= x_p[k+1] - \hat{x}_p[k+1] \\ &= (A_p - LC_p)e[k] + w[k] - Lv[k] \end{aligned} \tag{5.66}$$

The dynamics of $x_p[k]$ can be expanded as follows:

$$\begin{aligned} x_p[k+1] &= A_p x_p[k] + B_p u[k] + w[k] \\ &= A_p x_p[k] + B_p(K(\hat{x}_p[k] - x_{ref}) + u_{ref}) + w[k] \\ &= (A_p + B_pK)x_p[k] - B_pKe[k] + w[k] + B_p \underbrace{(-Kx_{ref} + u_{ref})}_{\phi_u} \end{aligned} \tag{5.67}$$

The joint dynamics of $x_p[k]$ and $e[k]$ can be rewitten as:

$$\underbrace{\begin{bmatrix} x_p[k+1] \\ e[k+1] \end{bmatrix}}_{\bar{x}[k+1]} = \underbrace{\begin{bmatrix} (A_p + B_pK) & -B_pK \\ 0 & (A_p - LC_p) \end{bmatrix}}_{\bar{A}} \underbrace{\begin{bmatrix} x_p[k] \\ e[k] \end{bmatrix}}_{\bar{x}[k]} + \underbrace{\begin{bmatrix} I & 0 \\ I & -L \end{bmatrix}}_{\bar{E}} \underbrace{\begin{bmatrix} w[k] \\ v[K] \end{bmatrix}}_{\bar{v}[k]} + \underbrace{\begin{bmatrix} B_p \\ 0 \end{bmatrix}}_{\bar{G}} \phi_u \tag{5.68}$$

[4] Note that at design stage random-number generators are necessary for the definition of the system parameters; this is done offline and does not clash with our statement here.

The relationships between $u[k]$ and $\bar{x}[k]$ can be presented as follows:

$$u[k] = K(\hat{x}_p[k] - x_{ref}) + u_{ref} = K(x_p[k] - e[k]) + \phi_u = \underbrace{\begin{bmatrix} K & -K \end{bmatrix}}_{\bar{K}} \bar{x}[k] + \phi_u \tag{5.69}$$

The joint dynamics of the $y[k]$ will then be

$$\begin{aligned} y[k] &= C_p x_p[k] + v[k] \\ &= \underbrace{\begin{bmatrix} C_p & 0 \end{bmatrix}}_{\bar{C}} \bar{x}[k] + \underbrace{\begin{bmatrix} 0 & I \end{bmatrix}}_{\bar{F}} \bar{v}[k] \end{aligned} \tag{5.70}$$

The joint dynamics of the $\bar{x}[k]$ and $y[k]$ can be represented as:

$$\begin{aligned} \bar{x}[k+1] &= \bar{A}\bar{x}[k] + \bar{E}\bar{v}[k] + \bar{G}\phi_u \\ y[k] &= \bar{C}\bar{x}[k] + \bar{F}\bar{v}[k] \end{aligned} \tag{5.71}$$

Because $\bar{x}[k]$ and $\bar{v}[k]$ are uncorrelated, the dynamics of the mean and covariance of $\bar{x}$ will be:

$$\begin{aligned} \mu_{\bar{x}}[k+1] &= \bar{A}\mu_{\bar{x}}[k] + \bar{E}\mu_{\bar{v}}[k] + \bar{G}\phi_u \\ \Sigma_{\bar{x}}[k+1] &= \bar{A}\Sigma_{\bar{x}}[k]\bar{A}^\top + \bar{E}\Sigma_{\bar{v}}[k]\bar{E}^\top \end{aligned} \tag{5.72}$$

Remember $\bar{A}$ is stable, there will be a steady state $\Sigma_{\bar{x}}$ and $\mu_{\bar{x}}$ which meets the following equations:

$$\begin{aligned} \mu_{\bar{x}} &= \bar{A}\mu_{\bar{x}} + \bar{E}\mu_{\bar{v}} + \bar{G}\phi_u \\ \Sigma_{\bar{x}} &= \bar{A}\Sigma_{\bar{x}}\bar{A}^\top + \bar{E}\Sigma_{\bar{v}}\bar{E}^\top \end{aligned} \tag{5.73}$$

Take the dynamics of the watermarking generator into consideration:

$$\begin{aligned} x_w[k+1] &= A_w x_w[k] + B_w(y_p[k] + Hu[k]) \\ y_w[k+1] &= C_w x_w[k] + B_w(y_p[k] + Hu[k]) \end{aligned} \tag{5.74}$$

Define $\bar{u}[k] = y_p[k] + Hu[k]$, the $\bar{u}[k]$ can be expanded as:

$$\begin{aligned} \bar{u}[k] &= y_p[k] + Hu[k] \\ &= (\bar{C} + H\bar{K})\bar{x}[k] + \bar{F}\bar{v}[k] + H\phi_u \end{aligned} \tag{5.75}$$

Then the dynamics of the mean $\mu_{\bar{u}}$ and variance $\Sigma_{\bar{u}}$ of $\bar{u}$ is as follows:

$$\begin{aligned} \mu_{\bar{u}} &= (\bar{C} + H\bar{K})\mu_{\bar{x}} + \bar{F}\mu_{\bar{v}} + H\phi_u \\ \Sigma_{\bar{u}} &= (\bar{C} + H\bar{K})\Sigma_{\bar{x}}(\bar{C} + H\bar{K})^T + \bar{F}\Sigma_{\bar{v}}\bar{F}^T \end{aligned} \tag{5.76}$$

So the mean and the covariance of $x_{w,u}[k]$ can be calculated as follows:

$$\begin{aligned} \mu_{x_{w,u}}[k+1] &= A_{w,u}\mu_{x_{w,u}}[k] + B_{w,u}\mu_{\bar{u}}[k] \\ \Sigma_{x_{w,u}}[k+1] &= A_{w,u}\Sigma_{x_{w,u}}[k]A_{w,u}^\top + B_{w,u}\Sigma_{\bar{u}}[k]B_{w,u}^\top \\ &\quad + \operatorname{cov}(A_{w,u}x_{w,u}[k], B_{w,u}\bar{u}[k]) + \operatorname{cov}(A_{w,u}x_{w,u}[k], B_{w,u}\bar{u}[k])^\top \end{aligned} \tag{5.77}$$

Notice that $\operatorname{cov}(A_{w,u}x_{w,u}[k], B_{w,u}\bar{u}[k]) = A_{w,u}\operatorname{cov}(x_{w,u}[k], \bar{u}[k])B_{w,u}$In order to calculate $\Sigma_{x_{w,u}}[k+1]$, we need to find $\operatorname{cov}(x_{w,u}[k], \bar{u}[k])$. It can be expanded as follows:

$$\begin{aligned} \operatorname{cov}(x_{w,u}[k], \bar{u}[k]) &= \operatorname{cov}(x_{w,u}[k], (\bar{C} + H\bar{K})\bar{x}[k] + \bar{F}\bar{v}[k] + H\phi_u) \\ &= \operatorname{cov}(x_{w,u}[k], \bar{x}[k])(\bar{C} + H\bar{K})^\top \end{aligned} \tag{5.78}$$

From equation (5.78), we need to find the covariance between $x_{w,u}[k+1]$ and $\bar{x}[k+1]$. It can be expanded:

$$\begin{aligned} \operatorname{cov}(x_{w,u}[k+1], \bar{x}[k+1]) &= \operatorname{cov}(A_{w,u}x_{w,u}[k] + B_{w,u}\bar{u}[k], \bar{A}\bar{x}[k] + \bar{E}\bar{v}[k] + G\phi_u) \\ &= \operatorname{cov}(A_{w,u}x_{w,u}[k], \bar{A}\bar{x}[k]) + \operatorname{cov}(B_{w,u}\bar{u}[k], \bar{A}\bar{x}[k]) + \operatorname{cov}(B_{w,u}\bar{u}[k], \bar{E}\bar{v}[k]) \\ &= A_{w,u}\operatorname{cov}(x_{w,u}[k], \bar{x}[k])\bar{A}^\top + B_{w,u}(\bar{C} + H\bar{K})\operatorname{var}(\bar{x}[k])\bar{A}^\top + B_{w,u}\bar{F}\operatorname{var}(\bar{v}[k])\bar{E}^\top \end{aligned} \tag{5.79}$$

Because $A_{w,u}$ is time-invariant and stable, then the $\text{cov}(x_{w,u}[k], \bar{x}[k])$ have a steady-state value is the solution of a Sylvester equation with form $AXB-X+C=0$. With the value of $\text{cov}(x_{w,u}[k], \bar{x}[k])$, the $\text{cov}\left(x_{w,u}[k], \bar{u}[k]\right)$ can then be calculated from Equation (5.78). Then we have the mean and the covariance of $x_{w,u}[k]$ as follows:

$$\begin{aligned}\mu_{x_{w,u}} &= A_{w,u}\mu_{x_{w,u}} + B_{w,u}\mu_{\bar{u}} \\ \Sigma_{x_{w,u}} &= A_{w,u}\Sigma_{x_{w,u}}A_{w,u}^{\top} + B_{w,u}\Sigma_{\bar{u}}B_{w,u}^{\top} \\ &\quad + A_{w,u}\,\text{cov}(x_{w,u},\bar{x})(\bar{C}+H\bar{K})^{\top}B_{w,u}^{\top} + (A_{w,u}\,\text{cov}(x_{w,u},\bar{x})(\bar{C}+H\bar{K})^{\top}B_{w,u}^{\top})^{\top}\end{aligned} \tag{5.80}$$

Because $A_{w,u}$ is time-invariant and stable, then we can also calculate the converged $\mu_{x_{w,u}}$ and $\Sigma_{x_{w,u}}$, based on the discrete-time Lyapunov Function. Then it is possible to define the steady state values of $\mu_{x_{w,u}}$ and $\Sigma_{x_{w,u}}$. The steady-state statistics of $x_{w,u}$ can then be used to partition $\mathbb{R}^{n_u}$ into N polyhedra, each having the same probability, using, e.g., the cumulative distribution function of the multiparametric Gaussian distribution.

Remark 29. The procedure outlined in this section only considers using $x_{w,u}[k]$ as the decision variable of mode selection. This is, of course, only one possible solution, as mode selection can also depend on $x_w[k]$ as a whole, or $u[k]$. The evaluation of whether there are any (dis)advantages in making one choice instead of another is left for further work. ◁

Remark 30. Note that the procedure proposed in this section to define $\mathscr{P}_i, i \in \mathscr{N}$ depend on the references $x_{p,ref}, u_{ref}$, as it biases the unobservable state's mean. As such, it is necessary to change $\mathscr{P}_i$ whenever $x_{p,ref}$ changes, which requires $x_{p,ref}$ to be transmitted between $\mathscr{C}$ and $\mathscr{P}$, and for $\mathscr{W}$ to have sufficient computational resources to execute the computation. We leave the development of a definition of the partitioning $\mathscr{P}_i$ that is time-invariant as future work. ◁

5.3. Identification Resistance

Having presented our proposed design strategy for the HMWM in Section 5.2, and having thus addressed Problem 3.2.a. and Problem 3.2.b., we can now evaluate our scheme against an adversarial eavesdropper attack, as the one defined in Section 3.2. Before providing details on this, note that, if seen from the cryptography perspective, $\mathscr{W}$ and $\mathscr{Q}$ can be seen as procedures that encode and decode the transmitted data. From this viewpoint, $\theta_w[k]$ and $\theta_q[k]$ can be considered as secret keys, guaranteeing security. In assessing the security of cryptographic algorithms, the computational complexity required to *break* them is evaluated, which often takes the form of evaluating the complexity of solving inverse problems over the field of integers modulo a prime [67]. The techniques for evaluating the security of cryptographic algorithms inspire our evaluation of our proposed methodology, which relies on three metrics:

1. the computational complexity of identifying the system parameters;
2. the amount of memory required to perform identification;
3. an evaluation of the theoretical difficulties associated with identifying the model of PWA hybrid system dynamics with unobservable states.

In the remainder of this section, we demonstrate how, by designing the dynamics of $\mathscr{W}$ and $\mathscr{Q}$ according to Algorithm 2, the obtained result is hard to identify.

Theorem 15. *Considering multiplicative watermarking systems $\mathscr{W}$ and $\mathscr{Q}$, designed following Algorithm 2, the computational complexity of exactly identifying $\theta_w[k]$ and $\theta_q[k]$ from $\mathscr{I}_a[k]$ is $\mathscr{NP}$-hard.* □

Proof. The proof follows directly from the computational complexity of solving exact identification of problems 7 8 and 9, i.e. theorems 8, 9, 10. ■

Remark 31. As mentioned in [68, Ch.5], analysis of the complexity of different bounded-error identification strategies for switched systems is conducted by restricting solutions to the set of rational numbers rather than the reals. We apply these results here without loss of generality, as in practice, the solution we propose is to be applied to a digital control system, and for matching parameters to be guaranteed, a fixed point representation is likely to be necessary. ◁

Although Theorem 15 gives a result for the computational complexity of *exact* identification of the system parameters, there are some other methods to find some approximate solutions for input-output models or state-space models of the system. Some are introduced in Section 5.1.3.

Table 5.1: IO Identification Complexity

n_m	n_h	IO	IO dimension	Sample Complexity
s	ν	s^ν	$(n_y+n_u)\nu$	$\frac{((n_y+n_u)\nu-1)s^\nu+((p+m)\nu+1)s^\nu}{2}$

One possible way is to use some heuristic way to identify parameters from the state-space model perspective. However, as stated in drawbacks **D 16.1** and **D 16.1**, these methods assume either a minimum dwell time or pathwise-observable property. The scheme presented in this paper does not satisfy these properties.

Another possible method is to use some heuristic way to identify parameters from the input-output model perspective, i.e. PWARX perspectives. The following result pertains to the difficulty of identifying PWA systems with unobservable outputs.

Theorem 16. *Consider a multiplicative watermarking scheme for which $\mathscr{W}$ and $\mathscr{Q}$ are designed following Algorithm 2, which does not admit a PWARX model.* □

Proof. Based on proposition 5.1.1 Given that $\mathscr{W}$ and $\mathscr{Q}$ are defined as unobservable, they can only be observed in infinite time. The theorem's statement follows directly. ■

Finally, let us comment on the storage space which may be necessary to compute an approximate solution to the parameters of the HMWM scheme in the input-output form. Theorem 11 relates to the storage space.

Furthermore, in Table 5.1, we include a characterization of the sample complexity required to perform identification using an approximate IO model based on theorem 11, which truncates the input-output data at a horizon length of n_h while identifying a state-space model with n_m modes. Table 6.1 and Table 6.1 show how this grows intractable as the horizon length increases.

Instead of identifying an infinite-dimension IO model, a learning attacker can use a finite-dimension IO model to approximate the switching dynamics. According to [71, 79] for a state-space model with s submodels ($n_m = s$) and use ν horizons ($n_h = \nu$) to approximate it, the dimension of the IO model and sample complexity is shown in the Table 5.1. The complexity grows intractable as the horizon and the number of modes grows.

5.4. Conclusion

Inspired by the computational hardness problems in the cryptography analysis, in this chapter, we studied Problem 3.1, Problem 3.2 and Problem 3.3. First, we extended the structure of the multiplicative watermarking method by designing it to be a fast-switching PWA system switching based on some unobservable dynamics and additively merging the control signal information. We mathematically showed the detection performance of the replay attack and certain types of the control-signal-injection zero-dynamics attack. Besides, We provided some parameter design guidelines to improve the detection performance and guarantee stability, synchronization and fast-switching properties. Finally, we illustrated the hardness of the attacker to execute a malicious parameter identification.

In the next chapter, we will show the detection and identification resistance performance by numerical examples simulated on testbench 1 and testbench 2.

6

Hybrid Multiplicative Watermarking: Simulation Study

6.1. Detection Performance

This section will show the simulation result of the detection performance of the HMWM Theory coding method from two perspectives. We will also use numerical examples to verify theorem 13, theorem 14.

6.1.1. Testbench 1

For testbench 1, We designed $\mathscr{W}$ and $\mathscr{Q}$ with 5 states ($n_w = 5$), of which 2 are unobservable states ($n_u = 2$). The number of modes is 6 ($N = 6$). The following parameter is designed to detect the replay attack and control-signal-injection zero-dynamics attack. The parameter is generated randomly and meets the detectability condition in the theorem 13 (maximum absolute value of eigenvalue is 1.01) and theorem 14.

$$H = \begin{bmatrix} -3.2575 & 5.8857 \\ -6.7564 & -3.7757 \end{bmatrix}, \quad A_{w,u} = \begin{bmatrix} 0.3908 & 0 \\ 0 & 0.6076 \end{bmatrix}, \quad B_{w,u} = \begin{bmatrix} 0.1299 & 0.4694 \\ 0.5688 & 0.0119 \end{bmatrix}, \tag{6.1}$$

Following the example design procedure in section 5.2.4, the mean and variance of the steady-state $x_{w,u}$ are as follows. Figure 6.1a shows the switching partition of the region.

$$\mu_{x_{w,u}} = [-5.2162, 7.7169]^\top, \Sigma_{x_{w,u}} = \begin{bmatrix} 0.5346 & -0.0823 \\ -0.0823 & 0.1106 \end{bmatrix}. \tag{6.2}$$

Figures 6.2a and 6.2b show the detection performance of the HMWM Theory coding method under the replay attack and the control-signal-injection zero-dynamics attack on testbench 1. Figure 6.2a shows the detection performance of the replay attack. Different from Figure 3.3, the residual value exceeds $\hat{r}$ soon after the attack starts and successfully triggers an alarm. Figure 6.2b shows the detection performance of the control-signal-injection zero-dynamics attack. Different from Figure 3.4, the residual value exceeds $\hat{r}$ soon after the attack starts and successfully triggers an alarm.

6.1.2. Testbench 2

For testbench 2, We designed $\mathscr{W}$ and $\mathscr{Q}$ with 5 states ($n_w = 5$), of which 2 are unobservable states ($n_u = 2$). The number of modes is 6 ($N = 6$). The following parameter is designed to detect the replay attack and control-signal-injection zero-dynamics attack. The parameter is generated randomly and meets the detectability condition in the theorem 13 (maximum absolute value of eigenvalue is 1.01).

$$H = \begin{bmatrix} 5.4982 \\ 6.3461 \end{bmatrix}, \quad A_{w,u} = \begin{bmatrix} -0.6245 & 0 \\ 0 & 0.0431 \end{bmatrix}, \quad B_{w,u} = \begin{bmatrix} 0.7792 & 0.1299 \\ 0.9340 & 0.5688 \end{bmatrix}, \tag{6.3}$$

Following the example design procedure in section 5.2.4, the mean and variance of the steady-state $x_{w,u}$ are as follows. Figure 6.1c shows the switching partition of the region.

$$\mu_{x_{w,u}} = [0.5544, 0.9843]^\top, \Sigma_{x_{w,u}} = \begin{bmatrix} 0.2353 & 0.2082 \\ 0.1659 & 0.2743 \end{bmatrix}. \tag{6.4}$$

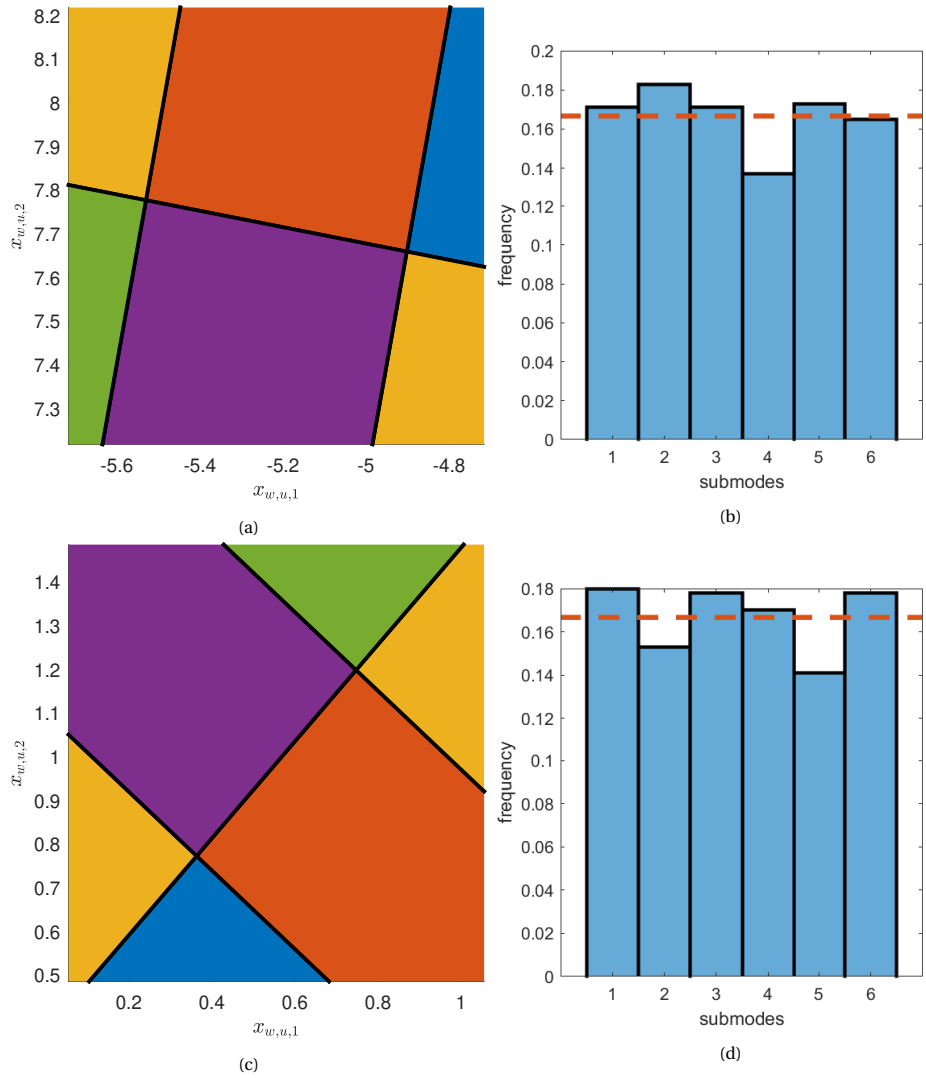


Figure 6.1: switching rule: (a) testbench 1: the switching partition with $N = 6$ modes. (b) testbench 1: relative frequency of each of the $N = 6$ modes over 1000 time steps. (c) testbench 2: the switching partition with $N = 6$ modes. (d) testbench 2: relative frequency of each of the $N = 6$ modes over 1000 time steps.

Figures 6.2c and 6.2d show the detection performance of the HMWM Theory coding method under the replay attack and the sensor-signal-injection zero-dynamics attack on testbench 2. Figure 6.2c shows the detection performance of the replay attack. Different from Figure 3.6, the residual value exceeds $\hat{r}$ soon after the attack starts and successfully triggers an alarm. Figure 6.2d shows the detection performance of the control-signal-injection zero-dynamics attack. Different from Figure 3.7, the residual value exceeds $\hat{r}$ soon after the attack starts and successfully triggers an alarm.

6.1.3. Theorem Verification

To verify theorem 13, we use testbench 1 and only use a single set of parameters of multiplicative watermarking. We randomly generate different values of H and make sure that the maximum absolute value of eigenvalue $\max|v|$ of $Phi_H = (A_p - B_pK - LC_p + LHK)$ locates in different regions: $[0,1), [1,1.2), [1.2,1.4)$. Figure 6.3 shows the verification result. From the figure, we can see that when $H = 0$ or $\max|v| \in [0,1)$, the system cannot detect replay attacks. When $\max|v| \geq 1$, the larger the $\max|v|$, the faster the residual exceeds the $\hat{r}$ and the attack is detected. This matches the theorem 13.

Figure 6.4 shows the detection performance when we have multiple sets of multiplicative watermarking. We still randomly generate different Hs and make sure that the maximum absolute value of eigenvalue $\max|v|$ of $\Phi_H = (A_p - B_pK - LC_p + LHK)$ located in different regions: $[0,1), [1,1.2), [1.2,1.4)$. From the figure, we can

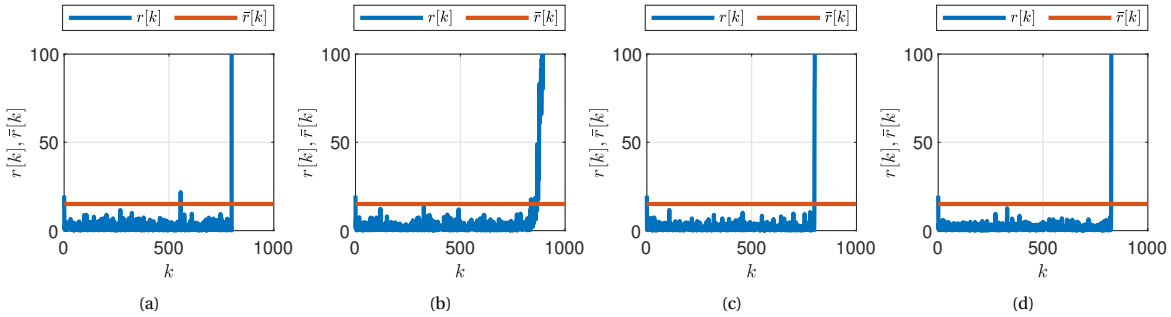


Figure 6.2: Detection performance of the hybrid multiplicative watermarking method, attacks start at $k_a = 800$. (a) residual value of testbench 1 under replay attack. (b) residual value of testbench 1 under control-signal-injection zero-dynamics attack. (c) residual value of testbench 2 under replay attack. (d) residual value of testbench 2 under sensor-signal-injection zero-dynamics attack

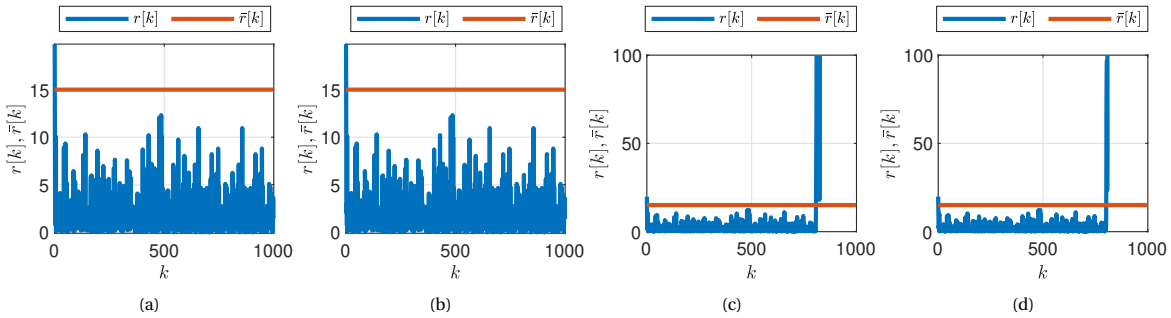


Figure 6.3: Test theorem of a replay attack with a single set of parameters and different values of H, attacks start at $k_a = 800$. (a) residual value when $H = 0$. (b) residual value when max eigenvalue $\max|v| \in [0,1)$. (c) residual value when max eigenvalue $\max|v| \in [1,1.2)$. (d) residual value when max eigenvalue $\max|v| \in [1.2,1.4)$

see that when $H = 0$, the system can still detect replay attacks, but the residual value fluctuates seriously. The main reason is that at that time the detection of replay attack is provided by the mismatching between replayed generator parameter and the true remover parameter, not depending on H. When $\max|v| \geq 1$, the attack can also be detected with a diverged residual value.

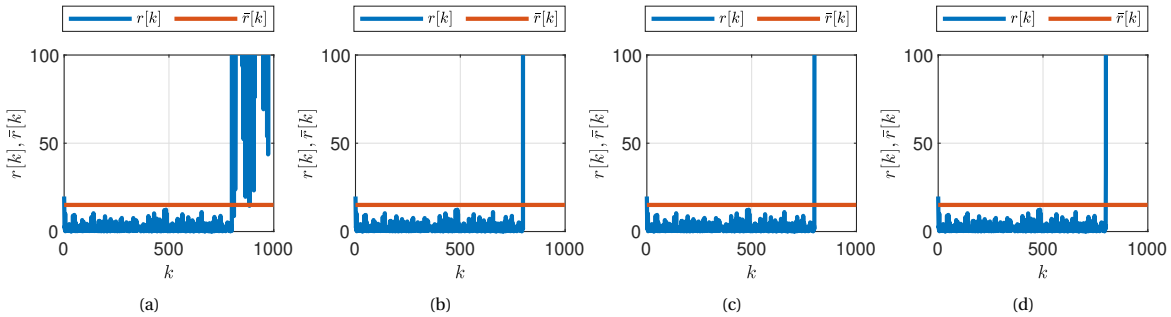
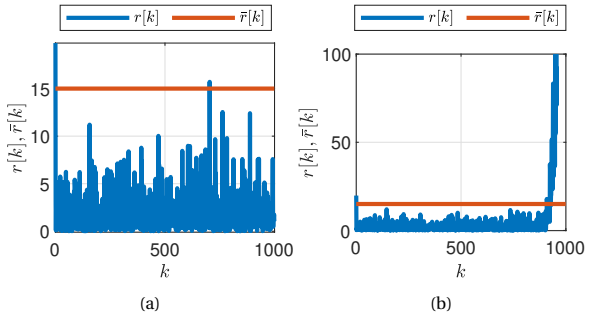


Figure 6.4: Test theorem of a replay attack with multiple sets of parameters and different values of H, attacks start at $k_a = 800$. (a) residual value when $H = 0$. (b) residual value when max eigenvalue $\max|v| \in [0,1)$. (c) residual value when max eigenvalue $\max|v| \in [1,1.2)$. (d) residual value when max eigenvalue $\max|v| \in [1.2,1.4)$

To verify theorem 14, we use testbench 1. The v value of the control-signal-injection zero-dynamics attack of testbench 1 is not the eigenvalue of A_p, which does not meet the condition in Proposition 5.2.2. We prepare two Hs as follows. H_1 is very large, with $H_1 g = 0$. H_2 is small, but $H_2 g \neq 0$. The verification result is in Figure 6.5. From the figure, we can see that, if $Hg = 0$, no alarm is triggered even with a very large H. If $Hg \neq 0$, the alarm will be successfully triggered even if H is small. This matches the theorem 14.

$$H_1 = \begin{bmatrix} 300 & 260 \\ 300 & 260 \end{bmatrix}, \quad H_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.5}$$



(a) (b)

Figure 6.5: Test theorem of control-signal-injection zero-dynamics attack with different values of H, attacks start at $k_a = 800$. (a) residual value when $H_1 g = 0$. (b) residual value when $H_2 g \neq 0$

6.2. Switching Rule Analysis

This section will analyze the switching rule based on the simulated data on Testbench 1 with parameters shown in equation (6.1) and Testbench 2 with parameters shown in equation (6.3). We will show the performance of hybrid multiplicative watermarking which is designed from the procedure in Section 5.2.4 from 4 perspectives:

1. We will show that our design of HMWM meets the requirements in **R1**, **R2** and **R3**. Which makes it hard to identify the parameters from state-space form.
2. We will show the numerical example of the sample complexity needed to identify the input-output model with approximately different horizons.
3. We tried to cluster IO data to analyze the mode of the input-output model with different clustering methods that correspond to the clustering functions `clusterdata()`, `knn()` and `dbscan()` in MATLAB. Since all regression methods finally match each data point to specific modes, we will show the result of comparing the actual modes and labels estimated by different methods. The label is estimated based on IO data, and different horizons' results are presented.
4. We tried to cluster state-space data to analyze the mode of the input-output model with different clustering methods that correspond to the clustering functions `clusterdata()`, `knn()` and `dbscan()` in MATLAB.. Since all regression methods finally match each data point to specific modes, we will show the result of comparing the actual modes and labels estimated by different methods.

Remark 32. For clustering results, We choose the random index (RI), the Fowlkes–Mallows index (FMI), and the Jaccard index (JI) to measure the performance. The closer these indices are to 1, the better the clustering result.

Remark 33. For clustering results, we show the result under both known-plaintext assumption and non-known-plaintext assumption.

6.2.1. Testbench 1

The simulation result shows that the switching rule meets the requirements in section 5.2.1:

1. **R1** in Figure 6.1b we show that, for $N = 6$ modes, each partition (shown in Fig. 6.1a) each mode is active approximately the same amount of time; furthermore during this simulation, 536 switching events occur, the median dwell time is 1, with a maximum dwell time of 11.
2. **R2**: The randomness of the mode sequence is guaranteed by design.
3. **R3**: in Figure 6.6a we show synchronization error of the watermarking systems' states, outputs and parameters; although there is a small error in the states of $\mathscr{W}$ and $\mathscr{Q}$, as well as between y_p and y_q (cfr. Figure 6.6a.a-Figure 6.6a.b) this does not impact the mode selection. These errors can be ascribed to numerical errors in MATLAB.

For the given watermarking parameter, the IO model dimension and the minimum number of samples needed to meet the PE requirement for different horizon numbers are shown in Table 6.1. The number of IO models and samples needed becomes intractable as the horizon number grows.

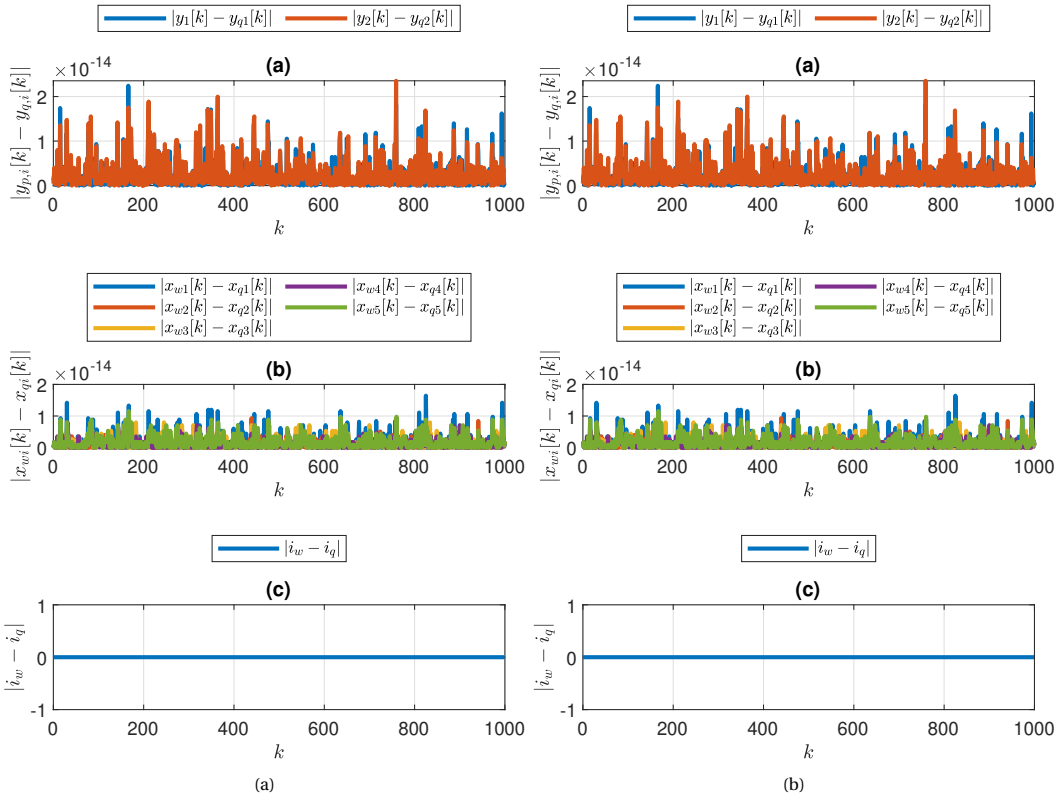


Figure 6.6: Hybrid Multiplicative Watermarking synchronization result. (a) Testbench 1 result (b) Testbench 2 result

Table 6.1: Testbench 1: Numerical IO Identification Complexity

Horizon Number	IO modes	number of samples
1	6	69
5	7776	5.7451×10^8
10	6.0466×10^7	7.1295×10^{16}
15	4.7018×10^{11}	6.5217×10^{24}

Figure 6.7 shows the clustering result based on input-output data to analyze an input-output model. Figure 6.7b assumes the attacker tries to cluster data under the known-plaintext attack assumption, while figure 6.7a is under the non-known-plaintext attack assumption. Table 6.2 shows the clustering result based on input-output data to analyze a state-space model from both known-plaintext and non-known-plaintext attacks. It can be seen that all these clustering results are not good, no matter whether it is a known-plaintext attack or not.

6.2.2. Testbench 2

The simulation result shows that the switching rule meets the requirements in section 5.2.1:

1. **R1** in Figure 6.1d we show that, for $N = 6$ modes, each partition (shown in Fig. 6.1c) each mode is active approximately the same amount of time; furthermore during this simulation, 894 switching events occur, the median dwell time is 1, with a maximum dwell time of 5.
2. **R2**: The randomness of the mode sequence is guaranteed by design.
3. **R3**: in Figure 6.6b we show synchronization error of the watermarking systems' states, outputs and parameters; although there is a small error in the states of $\mathscr{W}$ and $\mathscr{Q}$, as well as between y_p and y_q (cfr. Figure 6.6b.a-Figure 6.6b.b) this does not impact the mode selection. These errors can be ascribed to numerical errors in MATLAB.

For the given watermarking parameter, the IO model dimension and the minimum number of samples

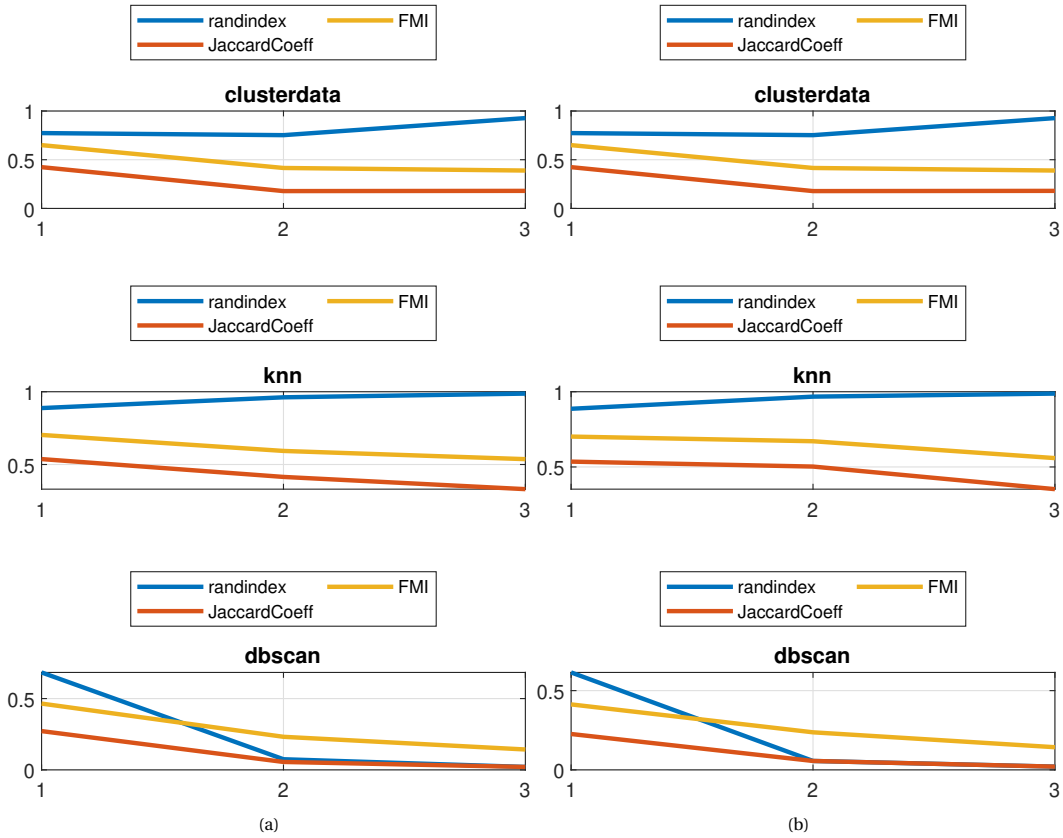


Figure 6.7: Testbench 1: clustering input-output model with different methods under different scenarios. (a) under non-known-plaintext attack. (b) under a known-plaintext attack.

Table 6.2: Testbench 1: clustering state-space model

	non-known-plaintext			known-plaintext		
methods	RI	JC	FMI	RI	JC	FMI
clusterdata	0.1771	0.1684	0.4084	0.1771	0.1685	0.4085
knn	0.7367	0.1451	0.2536	0.7203	0.0982	0.1789
dbscan	0.1691	0.1687	0.4108	0.1691	0.1687	0.4108

needed to meet the PE requirement for different horizon numbers are shown in Table 6.3. The number of IO models and samples needed becomes intractable as the horizon number grows.

Figure 6.8 shows the clustering result based on input-output data to analyze an input-output model. Figure 6.8b assumes the attacker tries to cluster data under the known-plaintext attack assumption, while figure 6.8a is under the non-known-plaintext attack assumption. Table 6.4 shows the clustering result based on input-output data to analyze a state-space model from known and non-known-plaintext attacks. It can be seen that all these clustering results are not good, no matter whether it is a known-plaintext attack or not.

6.3. Conclusion

In this Chapter, we illustrated the detection and identification resistance performance of the hybrid multiplicative watermarking method using the simulation results of testbench 1 and testbench 2. The simulation result shows that:

1. With suitably designed parameters, the hybrid multiplicative watermarking can have good detection performance. We also numerically verified the theorem related to the replay attack, i.e. theorem 13 and the control-signal-injection zero-dynamics attack, i.e. theorem 14.
2. Following the design procedure in Section 5.2.4, the hybrid multiplicative watermarking can meet re-

Table 6.3: Testbench 1: Numerical IO Identification Complexity

Horizon Number	IO modes	number of samples
1	6	69
5	7776	5.7451×10^{8}
10	6.0466×10^{7}	7.1295×10^{16}
15	4.7018×10^{11}	6.5217×10^{24}

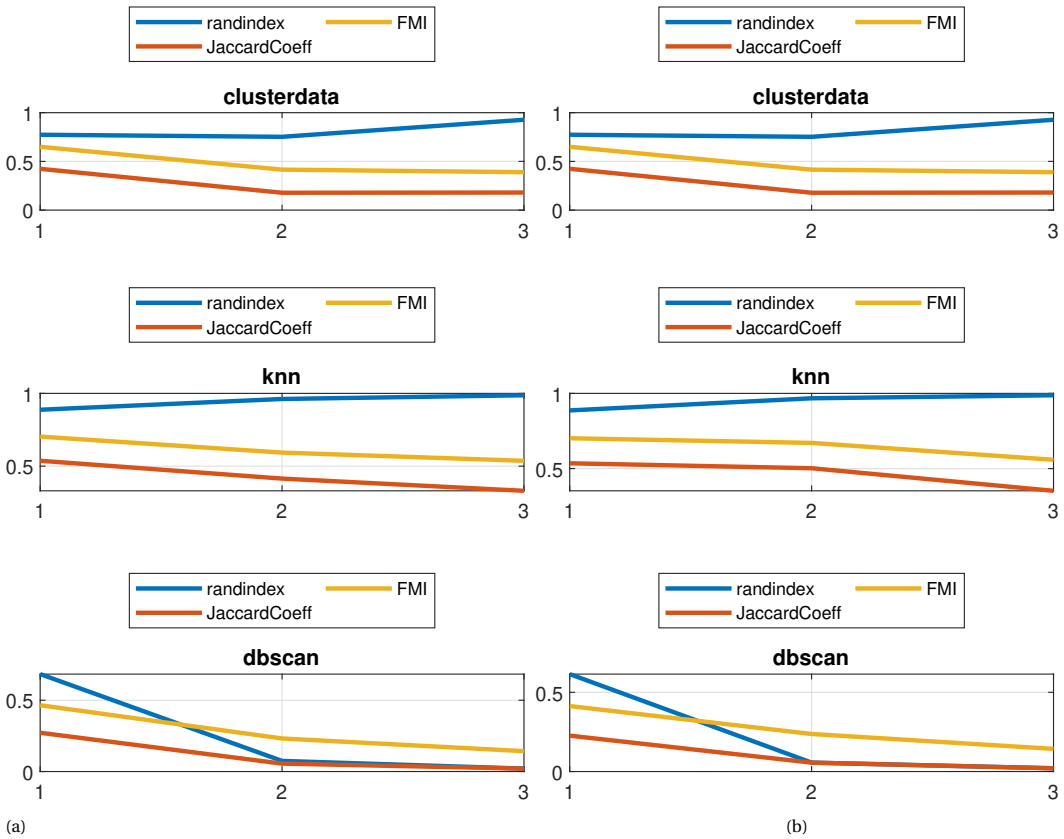


Figure 6.8: Testbench 2: clustering input-output model with different methods under different scenarios. (a) under non-known-plaintext attack. (b) under a known-plaintext attack.

quirements in **R1**, **R2** and **R3**.

3. By using the clustering result and the numerical result of the minimum sample, we illustrate that the hybrid multiplicative watermarking method can defend malicious parameter identification, even under the known-plaintext attack.

Table 6.4: Testbench 2: clustering state-space model

	non-known-plaintext			know-plaintext		
methods	RI	JC	FMI	RI	JC	FMI
clusterdata	0.1756	0.1684	0.4086	0.1771	0.1684	0.4084
knn	0.2369	0.1655	0.3890	0.7171	0.1011	0.1837
dbscan	0.2595	0.1649	0.3829	0.1691	0.1688	0.4108

7

Conclusion

Starting from introducing the current progress of active detection methods for cyber-physical systems, this thesis work addresses the problem of malicious parameter identification in current active detection methods. To address the problem, we propose two novel approaches: the *system immersion coding* method and the *hybrid multiplicative watermarking* method. Their design has a primal focus on disturbing the identification of attackers and defending against malicious parameter identification. Besides, as active detection methods, both of them are capable of detecting multiple attacks.

The system immersion method is a privacy solution used in federated learning. We enhance this method to detect malicious attacks by providing design guidelines to identify replay attacks, certain types of control-signal-injection zero-dynamics attacks, and certain types of sensor-signal-injection zero-dynamics attacks. However, we identify its vulnerability to *known-plaintext attacks*, which can compromise its effectiveness.

Motivated by the computation hardness problem in cryptography analysis, we propose a hybrid multiplicative watermarking scheme. In this approach, watermark parameters are periodically updated based on the dynamics of unobservable states in specially designed piecewise affine (PWA) hybrid systems. We provide design guidelines to detect replay attacks and certain types of control-signal-injection zero-dynamics attacks. Additionally, we demonstrate that our method enhances the computational complexity and systems-theoretic perspective of reconstructing watermarking parameters, making it challenging for eavesdroppers.

7.1. Conclusion and Answer of the Research Questions

This work focuses on addressing the malicious parameter identification problem in current active detection methods for cyber-physical systems. We begin by introducing the security and privacy issues in such systems and reviewing existing solutions. We highlight the vulnerability of most active detection methods to malicious parameter identification and propose two potential solutions: the *system immersion coding* method and the *hybrid multiplicative watermarking* method.

In Section 1.4, we proposed our general research questions 1 as *upgrading existing active detection methods or propose a new active detection method that can detect multiple attacks and defend malicious parameter identification*—then based on the system immersion method [36], the output-coding method [26] and the multiplicative watermarking method [30], we proposed two possible solutions for problem 1: one is combining the system-immersion coding method and the output-coding method to propose a new active detection method, the second is extending the structure of the multiplicative watermarking method.

System Immersion Coding Method

Throughout the thesis work, we answer the research questions in Problems 2, 2.1, 2.2, 2.3 as follows:

1. We propose a novel active detection method called the *system-immersion coding method*, which combines the system immersion method [36] and the output-coding method [26].
2. Our method can detect replay attacks, control-signal-injection zero-dynamics attacks, and sensor-signal-injection zero-dynamics attacks under some conditions. We provide design guidelines for selecting appropriate parameters to detect these attacks.

3. The proposed method effectively disrupts the attacker's accuracy in estimating design parameters without sacrificing performance.
4. However, we identify a vulnerability to known-plaintext attacks, where an attacker with estimated parameters can inject stealthy malicious attack sequences into the system.

Hybrid Multiplicative Watermarking Method

Throughout the thesis work, we answer the research questions in Problems 3, 3.1, 3.2, 3.3 as follows:

1. We propose a hybrid multiplicative watermarking (HMWM) based on the multiplicative watermarking method [30]. The watermark parameters are periodically updated, following the dynamics of the unobservable states of specifically designed piecewise affine (PWA) hybrid systems.
2. Our method improves the detection performance of the multiplicative watermarking method. Our proposed method can detect replay attacks and certain types of control-signal-injection zero-dynamics attacks under some conditions in the original multiplicative watermarking method.
3. We conduct a theoretical analysis to demonstrate the effects of the proposed scheme on closed-loop performance and stability preservation.
4. The HMWM approach enhances the complexity of reconstructing watermarking parameters, providing robustness from computational complexity and systems theoretic perspective.

7.2. Limitations and Future Work

While this work contributes valuable insights, it has some limitations. One limitation is that we only consider the stability under the scenario that the parameters of the generator and remover match. The system's behaviour when the parameters are mismatched is still missing but is important for real-life applications for the proposed method. Another limitation is that we only consider time-invariant reference points when designing the switching rule. However, in real-life scenarios, a system may have multiple or even time-varying reference points. Besides, the structures we proposed for the input information now only works on certain types of control-signal-injection zero-dynamics attack and sensor-signal-injection zero-dynamics attack, not all of them.

To expand on this research, we recommend exploring the following areas. The first is considering the robustness under poor network conditions. It is important to investigate the behaviour of the proposed methods when the generator and remover parameters are mismatched or when network conditions result in packet loss. Developing robustness in real-life scenarios with network protocols like UDP is crucial. The second is dealing with the time-varying reference point. That addresses the challenge of designing a switching rule for the hybrid multiplicative watermarking method when the reference point is time-varying.

Bibliography

[1] P. Derler, E. A. Lee, and A. Sangiovanni Vincentelli, "Modeling cyber–physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 13–28, Jan. 2012.

[2] R. Alguliyev, Y. Imamverdiyev, and L. Sukhostat, "Cyber-physical systems and their security issues," *Computers in Industry*, vol. 100, pp. 212–223, Sep. 2018.

[3] Y. Xu, Y. Yang, T. Li, J. Ju, and Q. Wang, "Review on cyber vulnerabilities of communication protocols in industrial control systems," in *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*, Nov. 2017, pp. 1–6.

[4] J. P. Farwell and R. Rohozinski, "Stuxnet and the future of cyber war," *Survival*, vol. 53, no. 1, pp. 23–40, Feb. 2011.

[5] S. Tan, J. M. Guerrero, P. Xie, R. Han, and J. C. Vasquez, "Brief survey on attack detection methods for cyber-physical systems," *IEEE Systems Journal*, vol. 14, no. 4, pp. 5329–5339, Dec. 2020.

[6] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "A secure control framework for resource-limited adversaries," *Automatica*, vol. 51, pp. 135–148, Jan. 2015.

[7] A. A. Cardenas, S. Amin, and S. Sastry, "Secure control: Towards survivable cyber-physical systems," in *2008 The 28th International Conference on Distributed Computing Systems Workshops*, Jun. 2008, pp. 495–500.

[8] S. M. Dibaji, M. Pirani, D. B. Flamholz, A. M. Annaswamy, K. H. Johansson, and A. Chakrabortty, "A systems and control perspective of cps security," *Annual Reviews in Control*, vol. 47, pp. 394–411, Jan. 2019.

[9] Q. Dinh Vu, R. Tan, and D. K. Y. Yau, "On applying fault detectors against false data injection attacks in cyber-physical control systems," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, Apr. 2016, pp. 1–9.

[10] L. Gao, B. Chen, and L. Yu, "Fusion-based fdi attack detection in cyber-physical systems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 8, pp. 1487–1491, Aug. 2020.

[11] D. Ye and T.-Y. Zhang, "Summation detector for false data-injection attack in cyber-physical systems," *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2338–2345, Jun. 2020.

[12] K. Manandhar, X. Cao, F. Hu, and Y. Liu, "Detection of faults and attacks including false data injection attack in smart grid using kalman filter," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 370–379, Dec. 2014.

[13] B. M. Sanandaji, E. Bitar, K. Poolla, and T. L. Vincent, "An abrupt change detection heuristic with applications to cyber data attacks on power systems," in *2014 American Control Conference*, Jun. 2014, pp. 5056–5061.

[14] Y. Shoukry and P. Tabuada, "Event-triggered state observers for sparse sensor noise/attacks," *IEEE Transactions on Automatic Control*, vol. 61, no. 8, pp. 2079–2091, Aug. 2016.

[15] S. Mishra, Y. Shoukry, N. Karamchandani, S. Diggavi, and P. Tabuada, "Secure state estimation: Optimal guarantees against sensor attacks in the presence of noise," in *2015 IEEE International Symposium on Information Theory (ISIT)*, Jun. 2015, pp. 2929–2933.

[16] H. Li, X. He, Y. Zhang, and W. Guan, "Attack detection in cyber-physical systems using particle filter: An illustration on three-tank system," in *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, Jul. 2018, pp. 504–509.

[17] A.-Y. Lu and G.-H. Yang, "Secure luenberger-like observers for cyber–physical systems under sparse actuator and sensor attacks," *Automatica*, vol. 98, pp. 124–129, Dec. 2018.

[18] X. Niu, J. Li, J. Sun, and K. Tomsovic, "Dynamic detection of false data injection attack in smart grid using deep learning," in *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Feb. 2019, pp. 1–6.

[19] Y. Li, L. Shi, and T. Chen, "Detection against linear deception attacks on multi-sensor remote state estimation," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 846–856, Sep. 2018.

[20] J. Yan, B. Tang, and H. He, "Detection of false data attacks in smart grid with supervised learning," in *2016 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2016, pp. 1395–1402.

[21] Y. He, G. J. Mendis, and J. Wei, "Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2505–2516, Sep. 2017.

[22] Y. Mo, S. Weerakkody, and B. Sinopoli, "Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs," *IEEE Control Systems Magazine*, vol. 35, no. 1, pp. 93–109, Feb. 2015.

[23] F. Miao, Q. Zhu, M. Pajic, and G. J. Pappas, "Coding sensor outputs for injection attacks detection," in *53rd IEEE Conference on Decision and Control*, Dec. 2014, pp. 5776–5781.

[24] R. M. Ferrari and A. M. Teixeira, "Detection and isolation of replay attacks through sensor watermarking * *this work has received funding from the european union seventh framework programme (fp7/2007-2013) under grant agreement no. 608224 and from h2020 programme under grant no. 707546 (sure)." *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7363–7368, Jul. 2017.

[25] P. Griffioen, S. Weerakkody, and B. Sinopoli, "A moving target defense for securing cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2016–2031, May 2021.

[26] H. Guo, Z.-H. Pang, J. Sun, and J. Li, "An output-coding-based detection scheme against replay attacks in cyber-physical systems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 10, pp. 3306–3310, Oct. 2021.

[27] M. Ghaderi, K. Gheitasi, and W. Lucia, "A blended active detection strategy for false data injection attacks in cyber-physical systems," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 1, pp. 168–176, Mar. 2021.

[28] F. Miao, Q. Zhu, M. Pajic, and G. J. Pappas, "Coding schemes for securing cyber-physical systems against stealthy data injection attacks," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 106–117, Mar. 2017.

[29] L. Zhai, K. G. Vamvoudakis, and J. Hugues, "Switching watermarking-based detection scheme against replay attacks," in *2021 60th IEEE Conference on Decision and Control (CDC)*, Dec. 2021, pp. 4200–4205.

[30] R. M. G. Ferrari and A. M. H. Teixeira, "A switching multiplicative watermarking scheme for detection of stealthy cyber-attacks," *IEEE Transactions on Automatic Control*, vol. 66, no. 6, pp. 2558–2573, Jun. 2021.

[31] A. J. Gallo and R. M. G. Ferrari, "Cryptographic switching functions for multiplicative watermarking in cyber-physical systems," Mar. 2022.

[32] G. Bottegal, F. Farokhi, and I. Shames, "Preserving privacy of finite impulse response systems," *IEEE Control Systems Letters*, vol. 1, no. 1, pp. 128–133, Jul. 2017.

[33] V. Katewa, A. Chakrabortty, and V. Gupta, "Differential privacy for network identification," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 1, pp. 266–277, Mar. 2020.

[34] H. Hayati, C. Murguia, and N. van de Wouw, "Privacy-preserving anomaly detection in stochastic dynamical systems: Synthesis of optimal gaussian mechanisms," Nov. 2022.

[35] P. Stobbe, T. Keijzer, and R. M. G. Ferrari, "A fully homomorphic encryption scheme for real-time safe control," Sep. 2022.

[36] H. Hayati, C. Murguia, and N. van de Wouw, "Privacy-preserving federated learning via system immersion and random matrix encryption," Apr. 2022.

[37] A. M. H. Teixeira and R. M. Ferrari, "Detection of sensor data injection attacks with multiplicative watermarking," in *2018 European Control Conference (ECC)*. Limassol: IEEE, Jun. 2018, pp. 338–343.

[38] J. Giraldo, A. Cardenas, and R. G. Sanfelice, "A moving target defense to detect stealthy attacks in cyber-physical systems," in *2019 American Control Conference (ACC)*, Jul. 2019, pp. 391–396.

[39] A. O. de Sá, L. F. R. d. C. Carmo, and R. C. S. Machado, "Covert attacks in cyber-physical control systems," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1641–1651, 2017.

[40] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. Monticello, IL, USA: IEEE, Sep. 2009, pp. 911–918.

[41] J. Tian, R. Tan, X. Guan, Z. Xu, and T. Liu, "Moving target defense approach to detecting stuxnet-like attacks," *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 291–300, Jan. 2020.

[42] R. S. Smith, "Covert misappropriation of networked control systems: Presenting a feedback structure," *IEEE Control Systems Magazine*, vol. 35, no. 1, pp. 82–92, Feb. 2015.

[43] K. Gheitasi and W. Lucia, "A finite-time stealthy covert attack against cyber-physical systems," in *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, vol. 1, Jun. 2020, pp. 347–352.

[44] S. Weerakkody, Y. Mo, and B. Sinopoli, "Detecting integrity attacks on control systems using robust physical watermarking," in *53rd IEEE Conference on Decision and Control*, Dec. 2014, pp. 3757–3764.

[45] B. Satchidanandan and P. R. Kumar, "Dynamic watermarking: Active defense of networked cyber–physical systems," *Proceedings of the IEEE*, vol. 105, no. 2, pp. 219–240, Feb. 2017.

[46] ——, "On the design of security-guaranteeing dynamic watermarks," *IEEE Control Systems Letters*, vol. 4, no. 2, pp. 307–312, Apr. 2020.

[47] J. Rubio-Hernán, L. De Cicco, and J. García-Alfaro, "Revisiting a watermark-based detection scheme to handle cyber-physical attacks," in *2016 11th International Conference on Availability, Reliability and Security (ARES)*, Aug. 2016, pp. 21–28.

[48] A. Naha, A. Teixeira, A. Ahlén, and S. Dey, "Deception attack detection using reduced watermarking," in *2021 European Control Conference (ECC)*, Jun. 2021, pp. 74–80.

[49] D. Du, C. Zhang, X. Li, M. Fei, T. Yang, and H. Zhou, "Secure control of networked control systems using dynamic watermarking," *IEEE Transactions on Cybernetics*, pp. 1–14, 2021.

[50] A. Khazraei, H. Kebriaei, and F. R. Salmasi, "A new watermarking approach for replay attack detection in lqg systems," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec. 2017, pp. 5143–5148.

[51] S. Weerakkody and B. Sinopoli, "A moving target approach for identifying malicious sensors in control systems," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sep. 2016, pp. 1149–1156.

[52] ——, "Detecting integrity attacks on control systems using a moving target approach," Jun. 2017.

[53] P. Griffioen, S. Weerakkody, and B. Sinopoli, "An optimal design of a moving target defense for attack detection in control systems," in *2019 American Control Conference (ACC)*, Jul. 2019, pp. 4527–4534.

[54] C. Schellenberger and P. Zhang, "Detection of covert attacks on cyber-physical systems by extending the system dynamics with an auxiliary system," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec. 2017, pp. 1374–1379.

[55] S. Weerakkody, O. Ozel, P. Griffioen, and B. Sinopoli, "Active detection for exposing intelligent attacks in control systems," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, Aug. 2017, pp. 1306–1312.

[56] S. Han and G. J. Pappas, "Privacy in control and dynamical systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 309–332, May 2018.

[57] A. Tsiamis, K. Gatsis, and G. J. Pappas, "State estimation codes for perfect secrecy," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec. 2017, pp. 176–181.

[58] ——, "State-secrecy codes for stable systems," in *2018 Annual American Control Conference (ACC)*, Jun. 2018, pp. 171–177.

[59] A. Tsiamis, A. B. Alexandru, and G. J. Pappas, "Motion planning with secrecy," in *2019 American Control Conference (ACC)*, Jul. 2019, pp. 784–791.

[60] A. Tsiamis, K. Gatsis, and G. J. Pappas, "State-secrecy codes for networked linear systems," *IEEE Transactions on Automatic Control*, vol. 65, no. 5, pp. 2001–2015, May 2020.

[61] W. Yang, D. Li, H. Zhang, Y. Tang, and W. X. Zheng, "An encoding mechanism for secrecy of remote state estimation," *Automatica*, vol. 120, p. 109116, Oct. 2020.

[62] M. Abdalmoaty, S. C. Anand, and A. M. H. Teixeira, "Privacy and security in network controlled systems via dynamic masking," Nov. 2022.

[63] A. Abdelwahab, W. Lucia, and A. Youssef, "Decoy-based moving target defense against cyber-physical attacks on smart grid," in *2020 IEEE Electric Power and Energy Conference (EPEC)*, Nov. 2020, pp. 1–5.

[64] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*, ser. Lecture Notes in Computer Science, M. Agrawal, D. Du, Z. Duan, and A. Li, Eds. Berlin, Heidelberg: Springer, 2008, pp. 1–19.

[65] L. Nandakumar, R. Ferrari, and T. Keviczky, "Privacy-preserving of system model with perturbed state trajectories using differential privacy: With application to a supply chain network," *IFAC-PapersOnLine*, vol. 52, no. 20, pp. 309–314, 2019.

[66] D. Umsonst and H. Sandberg, "Experimental evaluation of sensor attacks and defense mechanisms in feedback systems," *Control Engineering Practice*, vol. 124, p. 105178, Jul. 2022.

[67] J. Katz and Y. Lindell, *Introduction to modern cryptography*. CRC press, 2020.

[68] F. Lauer and G. Bloch, *Hybrid system identification*. Springer, 2019.

[69] S. Paoletti, J. Roll, A. Garulli, and A. Vicino, "Input-output realization of piecewise affine state space models," in *2007 46th IEEE Conference on Decision and Control*, Dec. 2007, pp. 3164–3169.

[70] S. Paoletti, A. Garulli, J. Roll, and A. Vicino, "A necessary and sufficient condition for input-output realization of switched affine state space models," in *2008 47th IEEE Conference on Decision and Control*, Dec. 2008, pp. 935–940.

[71] S. Paoletti, J. Roll, A. Garulli, and A. Vicino, "On the input-output representation of piecewise affine state space models," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 60–73, Jan. 2010.

[72] H. Lin and P. J. Antsaklis, "Stability and stabilizability of switched linear systems: A survey of recent results," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 308–322, Feb. 2009.

[73] D. Liberzon and A. Morse, "Basic problems in stability and design of switched systems," *IEEE Control Systems Magazine*, vol. 19, no. 5, pp. 59–70, Oct. 1999.

[74] H. Ye, A. N. Michel, and L. Hou, "Stability theory for hybrid dynamical systems," *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, vol. 43, no. 4, 1998.

[75] G. Feng, "Stability analysis of piecewise discrete-time linear systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 7, pp. 1108–1112, Jul. 2002.

[76] D. Mignone, G. Ferrari-Trecate, and M. Morari, "Stability and stabilization of piecewise affine and hybrid systems: An lmi approach," in *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*, vol. 1, Dec. 2000, pp. 504–509 vol.1.

[77] M. Lazar, W. Heemels, and A. Teel, "Subtleties in robust stability of discrete-time piecewise affine systems," in *2007 American Control Conference*, Jul. 2007, pp. 3464–3469.

[78] F. Lauer, "On the complexity of piecewise affine system identification," Sep. 2015.

[79] B. Mu, T. Chen, C. Cheng, and E.-w. Bai, "Persistence of excitation for identifying switched linear systems," *Automatica*, vol. 137, p. 110142, Mar. 2022.

[80] H. Nakada, K. Takaba, and T. Katayama, "Identification of piecewise affine systems based on statistical clustering technique," *Automatica*, vol. 41, no. 5, pp. 905–913, May 2005.

[81] M. G. Sefidmazgi, M. M. Kordmahalleh, A. Homaifar, and A. Karimoddini, "Switched linear system identification based on bounded-switching clustering," in *2015 American Control Conference (ACC)*, Jul. 2015, pp. 1806–1811.

[82] M. G. Sefidmazgi, M. M. Kordmahalleh, A. Homaifar, A. Karimoddini, and E. Tunstel, "A bounded switching approach for identification of switched mimo systems," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2016, pp. 004 743–004 748.

[83] R. V. Lopes, G. A. Borges, and J. Y. Ishihara, "New algorithm for identification of discrete-time switched linear systems," in *2013 American Control Conference*, Jun. 2013, pp. 6219–6224.

[84] K. Pekpe, G. Mourot, K. Gasso, and J. Ragot, "Identification of switching systems using change detection technique in the subspace framework," in *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*. Nassau, Bahamas: IEEE, 2004, pp. 3720–3725 Vol.4.

[85] L. Bako, G. Mercère, R. Vidal, and S. Lecoeuche, "Identification of switched linear state space models without minimum dwell time," *IFAC Proceedings Volumes*, vol. 42, no. 10, pp. 569–574, 2009.