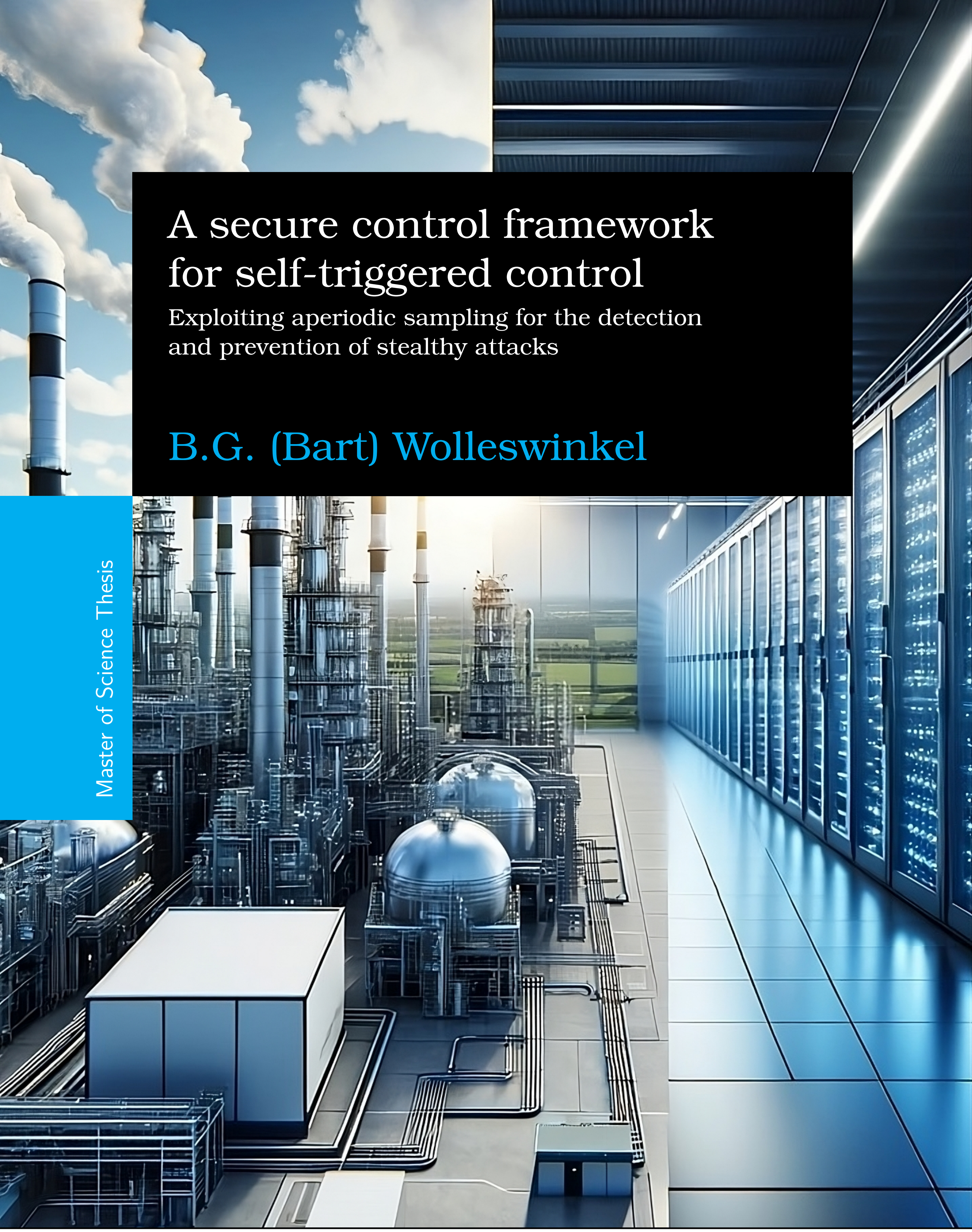


A secure control framework for self-triggered control

Exploiting aperiodic sampling for the detection and prevention of stealthy attacks

B.G. (Bart) Wolleswinkel

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

A secure control framework for self-triggered control

Exploiting aperiodic sampling for the detection and prevention of stealthy attacks

Master of Science Thesis

For the degree of Master of Science in Systems and Control
at Delft University of Technology

B.G. (Bart) Wolleswinkel

March 10, 2024

Faculty of Mechanical Engineering (ME) · Delft University of Technology

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical Engineering (ME) for acceptance a thesis entitled

A secure control framework
for self-triggered control

by

B.G. (Bart) Wolleswinkel

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: March 10, 2024

Supervisor(s): ____________________
dr.ir. M. (Manuel) Mazo Espinosa

Reader(s): ____________________
dr.ir. R.M.G. (Riccardo) Ferrari

____
dr.ir. G. (Gabriel) de Albuquerque Gleizer

____
dr.ir. L. (Laura) Ferranti

Abstract

This thesis addresses the detection and prevention of adversarial attacks on self-triggered control (STC) systems and demonstrates how manipulating the sampling times can be used to provide a secure control framework. Secure control is vital to guarantee both the safety and security of critical infrastructure, which has been the target of malicious attacks over the past decade.

First, a novel watermarking scheme is proposed based on the early triggering of a well-designed STC policy, such that stability is guaranteed to be preserved. We show that this watermarking scheme, together with an event-triggered χ^2 detector we design, is able to detect replay attacks. An online heuristic for obtaining an optimal early triggering policy is provided. If certain assumptions hold we show that the early triggering policy is a discrete uniform one. Through an illustrative example, both a quantitative and qualitative comparison between two other watermarking schemes are provided. We conclude that none of the watermarking schemes can claim absolute superiority, and trade-offs between all considered schemes exist.

Next, we propose a new type of attack called a switched zero dynamic attack (ZDA), and provide an algorithm on how to construct these switched ZDAs. We show that certain STC systems are susceptible to such attacks, and demonstrate that by tuning the triggering parameters there exist sufficient conditions such that these attacks are no longer disruptive. The effect of additive perturbations and a non-zero initial condition, as well as the proposed tuning method, are shown in a numerical example. We provide a qualitative comparison between several other countermeasures in the literature, which we extend for aperiodic sampling when needed. Finally, shortcomings and future directions are discussed.

Table of Contents

Acknowledgements **viii**

1 Introduction **1**

1-1 Introduction to secure control 1

1-2 Current developments in secure control 2

1-3 Outline 2

1-4 Contributions 3

2 Networked cyber-physical systems **5**

2-1 Architecture 6

3 Event and self-triggered control **9**

3-1 Time-triggered control 9

3-2 Event-triggered control 10

3-2-1 Continuous event-triggered control 12

3-3 Periodic event-triggered control 13

3-4 Self-triggered control 15

3-5 Stability and performance 17

3-6 Triggering condition 19

3-7 Observer design 21

4 Adversary model **23**

4-1 Types of attacks 25

4-2 Objective and constraints 26

5 A self-triggered control watermarking scheme 28

5-1 Replay attacks . . . 28

5-1-1 Existing countermeasures . . . 30

5-2 Event-triggered χ^2 detector . . . 31

5-3 Augmented triggering design . . . 33

5-3-1 Optimal watermarking scheme . . . 35

5-3-2 Offline procedure . . . 37

5-4 Control system design guidelines . . . 38

5-5 Illustrative examples . . . 40

5-5-1 Quantitative comparison . . . 44

5-5-2 Qualitative comparison . . . 45

5-6 Shortcomings . . . 46

6 Switched zero dynamics attacks 50

6-1 Zero dynamics attacks . . . 50

6-1-1 Sampling zeros . . . 51

6-2 Construction of a ZDA . . . 52

6-2-1 Controller invariant subspace . . . 53

6-2-2 Existing countermeasures . . . 53

6-3 Switched zero dynamic attack . . . 55

6-4 Extending inter-event times . . . 56

6-4-1 Resilience to switched ZDAs . . . 59

6-5 Illustrative examples . . . 60

6-5-1 Influence of imperfections . . . 61

6-5-2 Influence of large IETs . . . 62

6-5-3 Qualitative comparison . . . 66

7 Conclusion 67

7-1 Discussion . . . 67

7-2 Future work . . . 68

A Supplementary material for Chapter 5 70

A-1 Linear impulsive system model . . . 70

A-2 Offline static Kalman filter design . . . 71

A-3 Sketches of several proofs . . . 72

A-3-1 MSS of PETC policies with quadratic triggering conditions . . . 72

A-3-2 ISS of STC policies with dynamic controllers . . . 72

A-4 Influence of the controller state . . . 72

A-5 Different early triggering procedures in steady state . . . 73

B Supplementary material for Chapter 6 76

B-1 Extension of ZDAa on MIMO systems 76

B-2 Relative degree for MIMO systems 76

B-3 Masking attacks 76

B-4 Algorithm for obtaining the Byrnes-Isisdori normal form 77

B-5 Algorithm for deploying additional measurements 78

B-6 Additional dynamics 79

Bibliography 80

Glossary 92

List of Acronyms 92

List of Figures

2-1 Illustrative overview and relations between control system classifications 5

2-2 Overview of the typical structure of an networked cyber-physical system (NCPS) 6

3-1 Different sampling schemes, adapted from [68, Figure 1.4]. Note that in this section we assume full-state feedback such that $\boldsymbol{\gamma}(t) = \boldsymbol{\chi}(t)$. 12

3-2 The considered NCPS architecture. Novel contributions are highlighted in blue. . 17

4-1 Considered adversary model, adapted from [108, Figure 4] 23

4-2 Attack space with common and proposed attacks, adapted from [124, Figure 1] . 25

5-1 Overview of **(a)** additive watermarking and **(b)** multiplicative watermarking . . . 30

5-2 Illustrative examples of the optimal triggering policy under the proposed heuristic with $\bar{\kappa}_i = 5$. Note the asymmetry of the Kullback-Leibler (KL) divergence. . . . 37

5-3 Illustrative examples of the optimal triggering policy under the proposed heuristic with $\bar{\kappa}_i = 5$ in two specific scenarios, the latter being problematic for attack detection. 39

5-4 Effect of a replay attack when no watermarking is present (*baseline*) 41

5-5 Effect of a replay attack with the proposed STC watermarking policy 41

5-6 Effect of a replay attack with additive watermarking (*benchmark*) 42

5-7 Effect of a replay attack with multiplicative watermarking (*benchmark*) 43

5-8 Comparison of the inter-event times (IETs) for the four scenarios considered . . . 43

5-9 Boxplot of the squared tracking error $\|\boldsymbol{r}[k] - \boldsymbol{y}[k]\|^2$ 44

5-10 Effect of a replay attack during steady state regulation with the proposed method with $\sigma = 0.32$, $\epsilon = 0$, $\bar{\kappa} = 10$ and online watermarking 47

5-11 Effect of a replay attack during steady state with additive watermarking 48

5-12 Effect of a replay attack during steady state regulation with the proposed method with $\sigma = 0.32$, $\epsilon = 1$ and $\bar{\kappa} = 20$ 48

6-1 Root locus plot of both intrinsic zeros (green) and sampling zeros (blue) for different sampling periods $h \in \mathbb{R}_{>0}$ 58

6-2 Effect of a ZDA on a noise-free system with initial condition $\boldsymbol{x}_0 = \mathbf{0}$ 62

6-3 Effect of a ZDA on a noise-free system with non-zero initial condition $\boldsymbol{x}_0 = \mathbf{1}$ . . 63

6-4 Effect of a ZDA on a system with load disturbance ($\boldsymbol{\Sigma}_{\mathrm{w}} \approx 10^{-3} \cdot \boldsymbol{I}$) and initial condition $\boldsymbol{x}_0 = \mathbf{0}$ 63

6-5 Phase space with trajectories and the effect of a ZDA. Note that the trajectories are clipped outside the box $[-1.5, 1.5]^3$. 64

6-6 Effect of a ZDA on a noise-free system with large IETs 65

6-7 Effect of a ZDA on a system with load disturbances and large IETs 65

6-8 Generalized hold function 66

A-1 Effect of the dynamic controller state on the detection of replay attacks 73

A-3 Effect of a replay attack during steady state regulation with the proposed method with $\sigma = 0.32$, $\epsilon = 0$ and $\bar{\kappa} = 10$ offline procedure 74

A-2 Hypothesis over time. From the topmost figure (no watermarking) the raised alarm at the initialization of the attack can be mistaken for a false alarm. From the bottom figure (proposed method) the attack is distinguishable. 74

A-4 Effect of a replay attack during steady state regulation with the proposed method with $\sigma = 0.32$, $\epsilon = 0$, $\bar{\kappa} = 10$ and maximal difference 75

List of Tables

4-1 Categorization of attacks, adapted from [124] 27

5-1 Summary of the trade-offs between the different watermarking schemes 45

5-2 Estimated false alarm rate $\hat{p}_{\mathrm{fp}}$ and reported deviations in standard error (s.e.) . . 46

B-1 List of symbols 93

Acknowledgements

To begin, I would like to thank my supervisor dr.ir. M. (Manuel) Mazo Espinosa for his support, understanding, and assistance over the past months in writing this thesis. Secondly, I would like to thank dr.ir. R.M.G. (Riccardo) Ferrari for the interest shown in my work, as well as the time made to answer my questions. Thirdly, I thank dr.ir. G. (Gabriel) de Albuquerque Gleizer for the several lively discussions and insight provided during the later stages of my thesis. Finally, I would like to thank dr.ir. L. (Laura) Ferranti, particularly for joining the committee on such short notice.

I hope the topic of this thesis interests you, and I look forward to future collaboration.

Delft, University of Technology
March 10, 2024

B.G. (Bart) Wolleswinkel

Please consider the environment before printing.
This thesis is optimized for digital viewing.

“If I had more time, I would have written a shorter letter.”
— *Mark Twain*

Chapter 1

Introduction

1-1 Introduction to secure control

Over the last decades control systems have become increasingly digitized as well as ever more reliant on the use of wireless communication [87]. Amid the large-scale transition to industry 4.0, with the promising benefits of internet of things (IoT), cloud computing, and data sharing, we have seen the emergence of cyber-physical systems (CPSs), control systems where physical and software components are tightly interconnected and distributed over larger geographical areas [115, 39]. Within CPS, a networked control system (NCS) is characterized by the utilization of a (shared) wireless communication network, differentiating them from traditional wired point-to-point connections. They offer a flexible architecture where sensors and actuators can be conveniently removed, contributing to their lower deployment and maintenance cost [139, 46]. This makes them an attractive option in practice and as such, reports have shown that the use of wireless communication within CPS is expected to continue to grow at an unprecedented rate [119].

The use of wireless communication networks in NCSs, however, also poses additional challenges. Two main drawbacks are tight constraints on the available bandwidth of the communication network as well as energy constraints on the nodes in the communication networks, which are often battery-powered [1, 46]. One solution, first proposed in [122], to somewhat mitigate these effects is that of an event-triggered control (ETC) policy. Here, data is only transmitted upon the occurrence of a significant event, therefore hopefully reducing the network traffic and in turn expanding the lifespan of the components. Later, self-triggered control (STC) was introduced [86], where the next data transmission is determined based on a prediction model located on the controller side instead of at the sensors. This has the added benefit of the sensors being able to idle in between transmission times as well as having the event triggering mechanism where more computational resources are usually available [6].

Apart from the drawbacks mentioned above, another consideration in using (networked) CPS is that of cyber security. The vulnerability of CPSs to these types of attacks has become abundantly clear since the STUXNET worm infiltrated an Iranian nuclear facility in 2010

[20]. Since then, numerous occurrences of malicious attacks on control systems have been recorded, one of the most recent ones being the third blackout of the Ukrainian power grid, which was caused by the Russia-linked threat actor Sandworm [55]. This attack, which was immediately followed by a missile strike, demonstrated that cyberattacks can be used as a digital augmentation to warfare. As a large portion of our critical infrastructure (including but not limited to electric power, oil/gas, and water distribution [15, 47]) consists of CPSs, protecting these systems from cyberattacks is essential to prevent economical cost, societal cost or even loss of life [56, 47]. Yet experts indicate that there is a cyber risk gap due to a lack of awareness of the threat potential, but a growing severity and frequency of cyberattacks [102]. Whilst the number of publications on secure control has grown steadily over the last few years, relatively few implementations are used in the industry, for various reasons.

CPSs and NCSs in particular could potentially benefit substantially from employing STC policies. However, the majority of the ETC literature does not consider malicious cyberattacks [30]. This is in direct contradiction with ETC systems being designed to negate some of the drawbacks of NCSs, which itself are particularly vulnerable due to increased reliance on remote (and wireless) operation [102].

1-2 Current developments in secure control

Due to the aforementioned STUXNET worm as well as other examples such as the ransomware attack on a United States fuel pipeline in 2021 [34], the Maroochy sewage spill in 2000 [117, 56] and the Sandworm infiltration leading to the Ukraine power outage in 2015 [80, 77], security of control systems has become an active topic of research [114]. These attacks indicate that security is of fundamental importance to ensure the safe operation of CPSs [56] and that the risk is not a possibility, but a reality [114].

As mentioned in [9], whilst ETC policies have received attention both in fault detection and vulnerability to denial-of-service (DoS) attacks [120, 30], their work appears to be the first to consider replay attacks in a ETC framework. Furthermore, there seems to be a lack of literature on zero dynamic attacks (ZDAs) on aperiodically sampled systems. The vast majority of literature on secure control only considers periodic sampling, and as such, there is potentially much to gain by looking into the secure control of ETC systems, which appears to be a relatively novel direction.

1-3 Outline

This thesis contains seven chapters, including this introduction. The main matter comprises Chapters 2-6 and can be subdivided into three parts. In Chapter 7 we summarize the results and shortcomings and discuss future research directions.

The first part, consisting of Chapters 2-4, contains an introduction of both ETC and STC as well as an overview of attacks considered in secure control. We construct the general framework including relevant assumptions on e.g. the plant dynamics, network architecture, and intent of the adversary. In the second part, Chapter 5, we introduce the notion of a replay attack and propose a novel watermarking STC scheme as well as a procedure to construct an optimal strategy, both online and offline. In the third part, Chapter 6, we consider an adversary with different capabilities and introduce the notion of a switched ZDA. Then, we show that a control system with an STC policy can be susceptible to such attacks, and we propose possible countermeasures to prevent them. Note that the second and third parts are disjoint, and whilst they share the same framework as introduced in part one, the obtained results (which rely on different sets of assumptions) stand on their own.

1-4 Contributions

Our contributions can be summarized as follows:

- We propose a novel watermarking scheme based on modulating the inter-event times as dictated by an STC policy. To the best of the author's knowledge, this is the first time a watermarking strategy for STC systems has been proposed, particularly for non-deterministic systems.
- We construct a modified event-based χ^2 detector for the detection of replay attacks in STC systems, as well as introducing a new quadratic triggering condition suited for (periodic) reference tracking.
- We demonstrate that certain systems using an STC policy are susceptible to disruptive switched ZDAs. To the best of the author's knowledge, this is the first time ZDAs on aperiodically sampled systems have been considered.
- We suggest several countermeasures that can be taken to prevent ZDAs from becoming disruptive, employing recent results on the abstraction of traffic models of periodic event-triggered control (PETC) systems.
- We provide a rich simulation environment called NCSim which can be used to simulate an NCS in Python. Apart from the functionalities discussed in the result sections of this thesis a plethora of other features are available, such as DoS attacks, quantization, PETC, and sampled-data simulation.

Notation

Let $\mathbb{N} = \{1, 2, \ldots\}$ denote the set of natural numbers and $\mathbb{N}_0 \doteq \mathbb{N}\backslash\{0\}$ denote the set of non-negative integers. Let $\mathbb{R}_{\geqslant 0} = [0, \infty)$ denote the set of non-negative real numbers and $\mathbb{R}_{>0} = (0, \infty)$ denote the set of positive real numbers. Capitalized boldface letters $\boldsymbol{A}$ denote matrices and lowercase boldface letters $\boldsymbol{v}$ denote column vectors. Let $\boldsymbol{I}$ denote the identity matrix, $\mathbf{1}$ a vector with all elements equal to 1 and $\mathbf{0}$ a matrix with all elements equal to 0, all of the appropriate size. Let $\lambda(\boldsymbol{A}) = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ (accounting for multiplicity) denote the set of all n eigenvalues of the square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$. The operator $\mathrm{col}(\bullet)$ concatenates its operands vertically such that $\mathrm{col}(\boldsymbol{v}_1, \boldsymbol{v}_2) = [\begin{array}{cc} \boldsymbol{v}_1^{\mathrm{T}} & \boldsymbol{v}_2^{\mathrm{T}} \end{array}]^{\mathrm{T}}$. A real, symmetric positive (semi)-definite matrix $\boldsymbol{A} = \boldsymbol{A}^{\mathrm{T}}$ is denoted by $\boldsymbol{A} \succ 0$ ($\boldsymbol{A} \succcurlyeq 0$), and a negative (semi)-definite matrix by $\boldsymbol{A} \prec 0$ ($\boldsymbol{A} \preccurlyeq 0$). Let $\|\boldsymbol{v}\|$ ($\|\boldsymbol{A}\|$) denote the Euclidean 2-norm (operator norm) of the vector $\boldsymbol{x}$ (matrix $\boldsymbol{A}$), and $\|\boldsymbol{v}\|_{\boldsymbol{W}} = \sqrt{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{v}}$ denote the $\boldsymbol{W}$-weighted 2-norm of $\boldsymbol{v}$, with $\boldsymbol{W} \succcurlyeq 0$. For a symmetric matrix $\boldsymbol{A}$ described in blocks, we may use $\star$ to denote blocks that can be induced by symmetry. The trace of a square matrix $\boldsymbol{A}$ is denoted by $\mathrm{tr}(\boldsymbol{A})$. We use $\underset{\sim}{\mathcal{P}}$ ($\underset{\sim}{\boldsymbol{A}}$) to denote continuous-time systems (dynamics) to differentiate them from their discrete-time counterparts $\mathcal{P}$ ($\boldsymbol{A}$). A continuous function $\alpha : \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{\geqslant 0}$ belongs to class $\mathcal{K}$ if and only if $\alpha(0) = 0$ and $\alpha(r) < \alpha(r')$ for all $r < r'$. A continuous function $\beta : \mathbb{R}_{\geqslant 0} \times \mathbb{R}_{\geqslant 0} \to \mathbb{R}$ belongs to class $\mathcal{KL}$ if and only if $\beta(r, s)$ belongs to class $\mathcal{K}$ for any fixed s, $\beta(r, s) \geqslant \beta(r, s')$ for all $s < s'$ and any fixed r, and $\lim_{s \to \infty} \beta(r, s) = 0$ fixed r. The real (imaginary) part of a complex variable $z \in \mathbb{C}$ is denoted $\mathfrak{Re}\{z\}$ ($\mathfrak{Im}\{z\}$). The symbol $\mathbb{1}_{x_i \leqslant a}$ denotes the indicator function defined as $\mathbb{1}_{x_i \leqslant a} = 1$ if $x_i \leqslant a$ and 0 otherwise. The transfer function $\boldsymbol{H}(z)$ of a state-space representation is denoted by

$$\boldsymbol{H}(z) = \left(\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{C} & \boldsymbol{D} \end{array}\right). \tag{1-1}$$

Chapter 2

Networked cyber-physical systems

Cyber-physical systems (CPSs) are control systems where physical components and software are deeply intertwined and tightly interconnected [115]. In a CPS, the plant, actuators, and sensors are in the physical domain whilst the network layer and controller reside in cyberspace [107]. The real-time dynamical system is an essential component in the physical layer of the CPSs [140]. Nowadays, CPSs are widely used among industrial control systems (ICSs). Archetypical examples of ICSs include but are not limited to, chemical plants, manufacturing facilities, and power distribution (see Figure 2-1). The larger systems of such systems are usually implemented by means of supervisory control and data acquisition (SCADA) systems. They consist of multiple control and data acquisition systems, which might operate on entirely different time and spatial scales [115]. Finally, SCADA systems are often subject to strict real-time constraints [118], and as such demand the use of information technology (IT) solutions specifically catered to control systems.

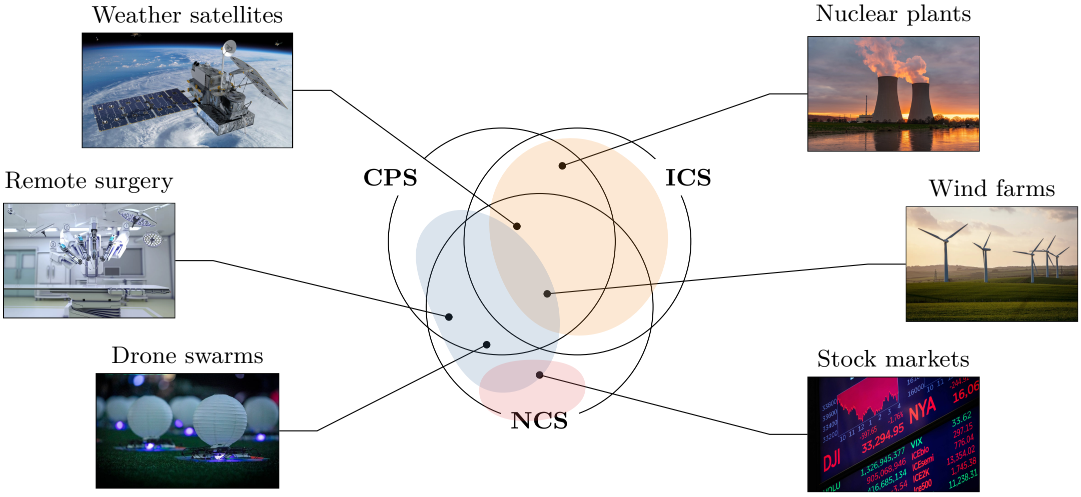


Figure 2-1: Illustrative overview and relations between control system classifications

Networked control systems (NCSs) are control systems where a (possibly shared, wireless) communications network is employed to facilitate the transmission of data from and to the controller. This differentiates them from traditional wired point-to-point connections where the plant and controller are collocated. Many SCADA systems employ (wireless) communication networks, and a key feature of SCADA systems is that they operate over multiple geographical locations and, as such, their communication networks need to span over large distances [39].

Note that all three definitions as mentioned above have considerable overlap and are therefore sometimes interchangeably used in the literature. To avoid ambiguity, here we will refer to them under the common denominator of networked cyber-physical system (NCPS), which we define as CPS employing a digital communication network similar to that in NCSs. Our proposed methods will be primarily focused on, but not limited to, application in ICSs. A simplified overview can be seen in Figure 2-2.

2-1 Architecture

We employ a *sampled-data* approach where the output of a continuous-time plant $\utilde{\mathcal{P}}$ is sampled and a digital controller $\mathcal{C}$ computes the actuation input which is held constant by the actuators. We restrict ourselves to linear time-invariant (LTI) dynamics of the form

$$\utilde{\mathcal{P}} \quad : \qquad \begin{aligned} \dot{\boldsymbol{\chi}}(t) &= \utilde{\boldsymbol{A}}\boldsymbol{\chi}(t) + \utilde{\boldsymbol{B}}\boldsymbol{\upsilon}(t) + \utilde{\boldsymbol{E}}\dot{\boldsymbol{\omega}}(t), && \text{(2-1a)} \\ \boldsymbol{\gamma}(t) &= \boldsymbol{C}\boldsymbol{\chi}(t) + \boldsymbol{\nu}(t), && \text{(2-1b)} \end{aligned}$$

where $\utilde{\boldsymbol{A}} \in \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{x}}$, $\utilde{\boldsymbol{B}} \in \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{u}}$, $\boldsymbol{C} \in \mathbb{R}^{n_\mathrm{y} \times n_\mathrm{u}}$ (note the absence of a feedforward matrix $\boldsymbol{D}$[1]), $\utilde{\boldsymbol{E}} \in \mathbb{R}^{n_\mathrm{x} \times \mathrm{x}}$, and $\dot{\boldsymbol{\omega}}(t)$ is the derivative in the generalized mean square sense of an n_x-dimensional Wiener process with incremental unit covariance $\boldsymbol{I}\mathrm{d}t$ [68, 53]. Furthermore, $\boldsymbol{\nu}(t)$ is an n_y-dimensional Wiener process with incremental covariance $\utilde{\boldsymbol{\Sigma}}_\nu \mathrm{d}t$. The load disturbance and measurement noise are independent of one another and independent of the current and previous state, which implies $\mathbb{E}\left[\boldsymbol{\omega}(t')\boldsymbol{\chi}^\mathrm{T}(t)\right] = \boldsymbol{0}$ and $\mathbb{E}\left[\boldsymbol{\nu}(t')\boldsymbol{\chi}^\mathrm{T}(t)\right] = \boldsymbol{0}$ for all $t' \geqslant t$ and

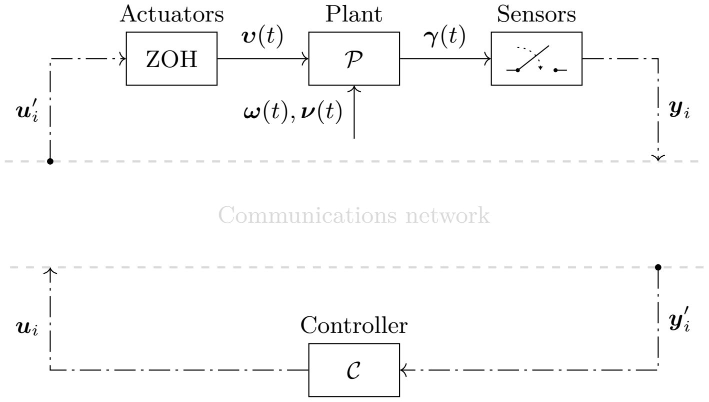


Figure 2-2: Overview of the typical structure of an NCPS

$\mathbb{E}[\boldsymbol{\omega}(t')\boldsymbol{\nu}^{\mathrm{T}}(t)] = \mathbf{0}$ for all t', t [131]. We make the following standard assumptions.

Assumption 1. *The pairs* $(\underset{\sim}{\boldsymbol{A}}, \underset{\sim}{\boldsymbol{B}})$ *and* $(\underset{\sim}{\boldsymbol{A}}, \underset{\sim}{\boldsymbol{E}})$ *are both controllable. Furthermore, the pair* $(\underset{\sim}{\boldsymbol{A}}, \boldsymbol{C})$ *is observable.* ◊

As can be seen in Figure 3-2 the output $\boldsymbol{\gamma}(t)$ of the continuous-time plant is sampled at the sensors such that $\boldsymbol{\gamma}(t_i) = \boldsymbol{y}_i$. Then, the measurement is sent to the controller over a communications network. The (dynamic) digital controller $\mathcal{C}$ is given by

$$\mathcal{C} \quad : \quad \begin{aligned} \boldsymbol{c}[k+1] &= \boldsymbol{A}_{\mathrm{c}}\boldsymbol{c}[k] + \boldsymbol{B}_{\mathrm{c}}\boldsymbol{y}[k], && \text{(2-2a)} \\ \boldsymbol{u}[k] &= \boldsymbol{C}_{\mathrm{c}}\boldsymbol{c}[k] + \boldsymbol{D}_{\mathrm{c}}\boldsymbol{y}[k], && \text{(2-2b)} \end{aligned}$$

where $\boldsymbol{y}[k] = \boldsymbol{y}_i$ for $k_i \leqslant k < k_{i+1}$ and similarly $\boldsymbol{u}_i = \boldsymbol{u}[k_i]$. Here, k_i denotes the i-th discrete-time event index (see Chapter 3). Note that static full-state feedback and output feedback controllers are a special case of (2-2). The controller input $\boldsymbol{u}_i$ is held constant at the actuators through a zero-order hold (ZOH) mechanism to produce $\boldsymbol{\upsilon}(t)$. Although we are considering a system architecture in which a communication network is present, we make the following assumption as is common in event-triggered control (ETC) literature.

Assumption 2. *The communications network has no communication delays, no packet drops, no quantization errors* [8]*, and all computations can be done in zero time* [64]*.* ◊

Relaxation of Assumption 2 is possible for explicit consideration of computation times (see e.g. [122]) and network induces phenomena such as communication delays and packet drops (see e.g. [103, 78]). To summarize, a simplified overview of the physical continuous-time plant $\underset{\sim}{\mathcal{P}}$ and digital discrete-time controller $\mathcal{C}$, separated by a communications network, can be seen in Figure 2-2.

Noise modeling

The dynamics as described by (2-1) with load disturbance $\dot{\boldsymbol{\omega}}(t)$ and measurement noise $\boldsymbol{\nu}(t)$ constitute a stochastic framework, where the distributions of the random vectors are assumed to be known. This form of modeling differs fundamentally from additive perturbations $\boldsymbol{\delta}(k)$, where the dynamics are given by

$$\underset{\sim}{\mathcal{P}} \quad : \quad \begin{aligned} \dot{\boldsymbol{\chi}}(t) &= \underset{\sim}{\boldsymbol{A}}\boldsymbol{\chi}(t) + \underset{\sim}{\boldsymbol{B}}\boldsymbol{\upsilon}(t) + \underset{\sim}{\boldsymbol{E}}\boldsymbol{\delta}(t), && \text{(2-3a)} \\ \boldsymbol{\gamma}(t) &= \boldsymbol{C}\boldsymbol{\chi}(t). && \text{(2-3b)} \end{aligned}$$

Here, the perturbation $\boldsymbol{\delta}(t)$ is assumed to be bounded ($\boldsymbol{\delta}(k) \in \mathcal{L}_\infty$) and possibly square-integrable ($\boldsymbol{\delta}(k) \in \mathcal{L}_2$) but do not admit any (known) probability distribution, and might even be deterministic. Note that this form of modeling constitutes the overwhelming majority of ETC literature (see §3-2), with notable exceptions being [26, 83]. The probabilistic framework

[1] The choice for $\boldsymbol{D} = \mathbf{0}$ is in part for the convenience of implementation in NCSim, as a value other than $\mathbf{0}$ creates an algebraic loop, for which integrated solvers are available in e.g. Simulink but have not been implemented in Python.

of stochastic systems is naturally less strict than the deterministic one, as it takes into account the disturbances' probability distribution, instead of being bound by worst-case scenarios [28]. However, note that for Gaussian noise the notion of $\mathcal{L}_2$-gain, $\mathcal{L}_\infty$-gain, and input-to-state stability (ISS) are not applicable, as Gaussian noise is not bounded nor square integrable. This does hinder the direct applicability of previous results on self-triggered control (STC), e.g. [50, 21]. As such, extensions of the methods proposed here to a framework with bounded perturbations are an interesting future direction.

Chapter 3

Event and self-triggered control

The closed-loop system as in Figure 2-2 in the absence of a communications network can be regarded as a conventional sampled-data system when the plant, sensors, actuators, and controller are all collocated, for which analysis has been well established since the late 1950s [139]. The existence of a (possibly shared) communications network in an networked cyber-physical system (NCPS) poses fundamentally new challenges and opportunities. To alleviate some of the drawbacks of networked control systems (NCSs) alternative sampling policies such as event-triggered control (ETC) have been proposed. We will first discuss the case of conventional periodic sampling, from here on referred to as time-triggered control (TTC).

3-1 Time-triggered control

In TTC the output $\boldsymbol{\gamma}(t)$ is sampled periodically with sampling period $h \in \mathbb{R}_{>0}$, and the continuous-time plant $\underset{\sim}{\mathcal{P}}$ as in (2-1) can be discretized as

$$\mathcal{P} \quad : \quad \begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{B}\boldsymbol{u}[k] + \boldsymbol{E}\boldsymbol{w}[k], && \text{(3-1a)} \\ \boldsymbol{y}[k] &= \boldsymbol{C}\boldsymbol{x}[k] + \boldsymbol{v}[k], && \text{(3-1b)} \end{aligned}$$

such that $\boldsymbol{x}[k] = \boldsymbol{\chi}(h \cdot k)$ with $k \in \mathbb{N}_0$. Given that the actuators perform a zero-order hold (ZOH) (also called *sample-and-hold* [64]) the matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{E}$ can be written as

$$\boldsymbol{A} = e^{\boldsymbol{A} \cdot h}, \qquad \boldsymbol{B} = \int_0^h e^{\boldsymbol{A} \cdot t} \, \mathrm{d}t \underset{\sim}{\boldsymbol{B}}, \qquad \boldsymbol{E} = \int_0^h e^{\boldsymbol{A} \cdot t} \, \mathrm{d}t \underset{\sim}{\boldsymbol{E}}. \tag{3-2}$$

Under Assumption 1 the pair $(\boldsymbol{A}, \boldsymbol{E})$ is controllable which implies the load disturbance $\boldsymbol{w}[k] \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_w)$ is a non-degenerate independent and identically distributed (i.i.d.) Gaussian random vector [28]. The covariance matrices $\boldsymbol{\Sigma}_\mathrm{w}$ and $\boldsymbol{\Sigma}_\mathrm{v}$ are then given by [131, 103]

$$\boldsymbol{\Sigma}_{\mathrm{w}} = \int_0^h e^{\utilde{\boldsymbol{A}}t} \utilde{\boldsymbol{E}}\utilde{\boldsymbol{E}}^{\mathrm{T}} e^{\utilde{\boldsymbol{A}}^{\mathrm{T}}t} \,\mathrm{d}t, \qquad \boldsymbol{\Sigma}_{\mathrm{w}} \approx \utilde{\boldsymbol{\Sigma}}_{\nu}/h, \tag{3-3}$$

where the latter becomes a better approximation for small values of h [131]. We see that for TTC the analysis of the sampled-data system becomes rather straightforward. Of particular interest is how to choose the sampling period h. To ensure stabilizability is maintained for the sampled-data system, h must be non-pathological, which we define next.

Definition 3.1 (Non-pathological sampling [79])**.** *Consider the continuous-time plant* $\utilde{\mathcal{P}}$ *as in* (2-1) *and suppose* h *is chosen such that for all* $\lambda_p, \lambda_q \in \lambda(\utilde{\boldsymbol{A}})$ *with* $\mathfrak{Re}\{\lambda_p\} = \mathfrak{Re}\{\lambda_q\}$ *we have* $\mathfrak{Im}\{\lambda_p\} \neq \mathfrak{Im}\{\lambda_q\} \cdot (2\pi \cdot \ell/h)$ *for all* $\ell \in \mathbb{N}$*. Then, the sampling period* h *is* non-pathological*.*

Our interest in non-pathological sampling stems from the fact that the discretization $\mathcal{P}$ inherits some of the beneficial properties of the original continuous-time system as is captured in the following lemma.

Lemma 3.1 ([17, Theorem 3.2.1])**.** *Consider the continuous-time plant* $\utilde{\mathcal{P}}$ *as in* (2-1) *and suppose* Assumption 1 *holds. Then, if the sampling period* h *is non-pathological the pairs* $(\boldsymbol{A}, \boldsymbol{B})$ *and* $(\boldsymbol{A}, \boldsymbol{C})$ *corresponding to the discretized dynamics* $\mathcal{P}$ *as in* (3-1) *are controllable and observable, respectively.*

In light of Lemma 3.1, we will make the following assumption.

Assumption 3. *The sampling period* h *is non-pathological.* ◊

Note that for a sufficiently small sampling period h Assumption 3 is always satisfied [65]. Furthermore, if $\utilde{\boldsymbol{A}}$ only has real eigenvalues, then all sampling periods h are non-pathological [4]. Although some detailed analytical results exist, in practice *ad hoc* rules are applied to determine stabilizing sampling periods h (for instance, 20 times the bandwidth of the system [45]) and whether Assumption 3 holds is checked *a posteriori* [86].

Whilst TTC has the benefit of making analysis simple it does have drawbacks, specifically in an NCPS as discussed here. Since sampling happens at a fixed rate regardless of whether it is really necessary or not, a TTC sampling policy is clearly a waste of communication resources [61]. As such, one of the disadvantages of TTC is that it unavoidably brings heavy (traffic) loads into the network, possibly leading to network congestion [139]. Furthermore, frequent changes in the actuator state, and consequently in the control input, lead to unnecessary energy consumption as well as actuator attrition [82]. To mitigate some of these drawbacks an ETC sampling policy has been proposed.

3-2 Event-triggered control

ETC (first mentioned in [143] under the name *Lebesgue sampling*) is a sampling policy where data is only transmitted upon the occurrence of an event. Such an event usually depicts an intolerable deviation of the state (or output) with respect to the last sample. As such, the transmission times are determined online by means of well-designed triggering rules [30]. Several variants and classifications of ETC exist and an overview is shown in Figure 3-1.

We first introduce the relevant notation to describe ETC policies. Let $t_i \in \mathbb{R}_{\geqslant 0}$ denote the time at which the i-th event occurs where, without loss of generality, we define $t_0 = 0$ and $i \in \mathbb{N}_0$. The i-th inter-event time (IET) $\tau_i = t_i - t_{i-1}$ denotes the time between successive events, where we similarly define $\tau_0 = 0$. Note that for TTC we have $\tau_i = h$ for all $i \neq 0$ as the occurrence of an 'event' is solely time-based (see Figure 3-1a). However, for ETC this is in general not the case, and sampling occurs aperiodically. Finally, the elapsed time $s(t) = \min_{t_i \leqslant t} t - t_i \in [0, \tau_{i+1})$ depicts the time since the last event instance, where we will often omit the explicit dependence on t for brevity.

In ETC, whether or not to transmit a new measurement over the communications network is controlled by a triggering function $\phi : \mathbb{R} \times \mathbb{R}^{2 \cdot n_\mathrm{x}} \to \{0, 1\}$, where $\phi(s, \boldsymbol{\xi}(t)) = 1$ denotes a new transmission (see Figure 3-1b and Figure 3-1c) and $\boldsymbol{\xi}(t) = \mathrm{col}(\boldsymbol{\chi}(t), \boldsymbol{x}_i)$ for $t \in [t_i, t_{i+1})$ denotes the augmented state vector. Here, $\boldsymbol{x}_i = \boldsymbol{\chi}(t_i)$ denotes the last sampled state instance. In the following, we present the results as if the full-state $\boldsymbol{\chi}(t)$ is available corresponding to the case $\boldsymbol{C} = \boldsymbol{I}$. Later we will discuss the extension to direct output feedback and observer-based output feedback (see §3-7).

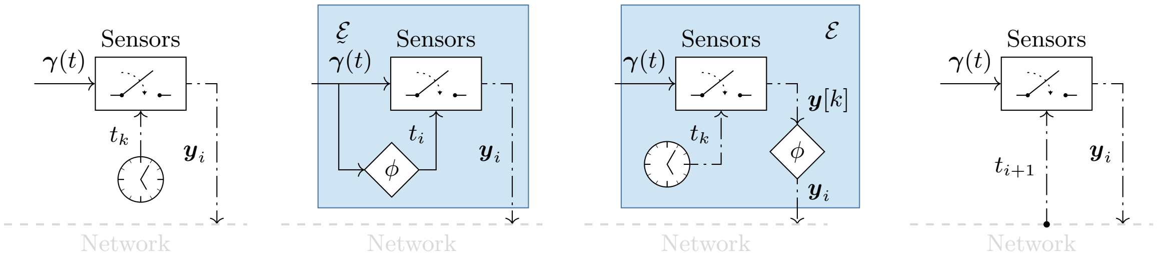
The triggering function ϕ must be appropriately constructed to ensure closed-loop stability and adequate performance. In principle, the triggering condition can be explicitly dependent on the elapsed time s. This form of triggering condition is called *dynamic* ETC (see §5-6). Here, we consider *static* triggering conditions (also called *time-invariant* [132]) such that $\phi : \mathbb{R}^{2 \cdot n_\mathrm{x}} \to \{0, 1\}$, where extensions to dynamic triggering are left to future work.

A common type of static triggering condition is a quadratic type of the form

$$\phi(\boldsymbol{\xi}(t)) = \begin{cases} 0 & \boldsymbol{\xi}^\mathrm{T}(t)\boldsymbol{Q}(\sigma)\boldsymbol{\xi}(t) \leqslant \epsilon, \\ 1 & \boldsymbol{\xi}^\mathrm{T}(t)\boldsymbol{Q}(\sigma)\boldsymbol{\xi}(t) > \epsilon. \end{cases} \tag{3-4}$$

Here, $\boldsymbol{Q}(\sigma) \in \mathbb{R}^{2 \cdot n_\mathrm{x} \times 2 \cdot n_\mathrm{x}}$ is a symmetric matrix to be designed, where we will often omit the parametric dependence on σ for brevity. The triggering parameter $\sigma \in \mathbb{R}_{\geqslant 0}$ is a tuneable parameter that determines how often events occur. As discussed in §3-5, the value of σ will be decisive in guaranteeing both stability and performance of the closed-loop system. Furthermore, the margin parameter $\epsilon \in \mathbb{R}_{\geqslant 0}$ is often beneficial in the presence of disturbances to avoid excessive triggering [50]. In most cases, $\epsilon = 0$ unless explicitly specified differently (see §5-6). Note that many forms of triggering conditions in the literature can be written as quadratic triggering conditions (see e.g. [61, 52]).

As is common in ETC literature, we introduce an upper bound $\bar{\tau} \in \mathbb{R}_{>0}$ to the IETs, which is a tunable design parameter. There are several reasons for the introduction of this upper bound. For one, it enforces robustness of the implementation by establishing a heartbeat of how frequently the plant state is desired to be monitored on the controller side [16, 85, 52]. Furthermore, as will become evident in §3-4 an upper bound is necessary for any self-triggered control (STC) implementation to have a guaranteed finite search space. It is important to note that an upper bound $\bar{\tau}$ does not adversely affect stability or performance in any way (see Theorem 5.1). A sufficiently large choice for $\bar{\tau}$ often has no influence on an ETC policy



(a) TTC **(b)** CETC **(c)** PETC **(d)** STC

Figure 3-1: Different sampling schemes, adapted from [68, Figure 1.4]. Note that in this section we assume full-state feedback such that $\gamma(t) = \boldsymbol{\chi}(t)$.

at all, as a largest IET often occurs naturally (e.g. for linear time-invariant (LTI) systems without disturbances [79]).

As such, ETC poses the following two advantages compared to TTC:

- **Less congestion**: Fewer transmissions which lead to less network-induced phenomena such as delay and packet drops [32, 48], as the available bandwidth in NCPSs is often very limited [82, 138].
- **Power saving**: Due to the reduction in transmissions less energy is consumed which maximizes network lifespan [54], as many components (sensors, actuators) are often battery-powered [54, 32, 82].
- **Reduced actuator attrition**: Less frequent changes in control input lead to less actuator attrition [82], due to reduced mechanical wear of the actuator [29].

In conclusion, the type of ETC policy considered here is summarized in the following assumption.

Assumption 4. *The triggering function* ϕ *is a* static, quadratic *triggering condition as in* (3-4) *with upperbound on the* IET $\bar{\tau}$*.* $\Diamond$

As discussed, the construction of ϕ determines the closed-loop behavior of the system. Whilst we can classify these triggering conditions as being *static* or *dynamic*, we can make a further distinction based on whether the triggering function continuously monitors the plant's output or only does so periodically, respectively called continuous event-triggered control (CETC) and periodic event-triggered control (PETC). We will briefly discuss CETC next after which we will discuss PETC and some of the advantages compared to CETC.

3-2-1 Continuous event-triggered control

In CETC, the triggering function ϕ is presumed to be able to continuously monitor the state $\boldsymbol{\chi}(t)$ of the plant (see Figure 3-1b). Under Assumption 4 the sequence of triggering times $t_1, t_2, \ldots, t_i$ can be formally defined as [64]

$$\underset{\sim}{\mathcal{E}} \quad : \qquad t_i = \inf_t \{ t \in \mathbb{R}_{>0}, t > t_{i-1} \,|\, \boldsymbol{\xi}^{\mathrm{T}}(t)\boldsymbol{Q}\boldsymbol{\xi}(t) > \epsilon \vee t = t_{i-1} + \bar{\tau} \}. \tag{3-5}$$

Note that these event times are only known at execution time, which makes analysis of ETC systems significantly more challenging than TTC. From (3-5) we find that the IETs τ_i can be any non-negative real number. For a more detailed overview of CETC we refer the reader to the work of e.g. [64] and the references therein.

Whilst CETC provides a more straightforward framework to analyze closed-loop stability compared to PETC (see §3-3), because an event can occur exactly when the relaxed (bound on the) decrease of a continuous-time Lyaponuv function is violated, it poses two major drawbacks. First, the practical implementation of CETC is difficult since specialized analog event monitoring hardware is required to continuously monitor the system state $\boldsymbol{\chi}(t)$ [48, 31]. As such, in practice, it may be unreasonable or impractical to retrofit an existing system with such hardware [100, 127]. Second, the IET τ_i might be zero in CETC, leading to undesirable and unimplementable Zeno behavior, which often happens when the state approaches the origin [82]. Furthermore, it is generally difficult to prove the existence of a strictly positive lower bound on the IETs for CETC [48]. Whilst modifications to CETC policies to overcome some of these challenges have been proposed, in particular, *time-regularized* CETC (see e.g. [13]), in part due to these limitations an alternative sampling policy called PETC has been proposed.

3-3 Periodic event-triggered control

In PETC, rather than constantly monitoring the state $\boldsymbol{\chi}(t)$, the state is only sampled periodically with sampling period $h \in \mathbb{R}_{>0}$ to obtain $\boldsymbol{x}[k]$, similar to TTC. Different from TTC, the sampled state being transmitted is again determined by a triggering function ϕ, similar to that in CETC. As such, PETC can be considered as having traits of TTC, whilst benefiting from an event-driven policy similar to CETC. This poses three of the main advantages of PETC compared to CETC:

- **Zeno-freeness**: Due to the periodic sampling of the output, a PETC policy is guaranteed to have a smallest IET $h \in \mathbb{R}_{>0}$, which can be set as a design parameter [63, 48].
- **Digital platform**: The periodic sampling is better suited for practical implementations on more standard time-sliced embedded software architectures [100, 61].
- **STC conversion**: A PETC policy can be readily transformed into an STC policy, at least in the case the case of full-state feedback [61, 85].

As a drawback, stability analysis of PETC policies becomes arguably more involved, which will be discussed in §3-5. Under Assumption 4, and by introduction of $\boldsymbol{p}[k] = \mathrm{col}(\boldsymbol{\chi}(h \cdot k), \boldsymbol{x}_i)$ for $k_i \leqslant k < k_{i+1}$, the sequence of triggering times $t_1, t_2, \ldots, t_i$ can be formally defined as [52]

$$\mathcal{E} \quad : \qquad t_i = h \cdot \min_k \{ k \in \mathbb{N}, k > k_{i-1} \,|\, \boldsymbol{p}^{\mathrm{T}}[k]\boldsymbol{Q}\boldsymbol{p}[k] > \epsilon \vee k = k_{i-1} + \bar{\kappa} \}, \tag{3-6}$$

where $\bar{\kappa} = \lfloor \bar{\tau}/h \rfloor$ is an upper bound on the largest inter-event index. Because the state is sampled periodically in PETC, the event times $t_i = h \cdot k_i$, with $k_i \in \mathbb{N}_0$. Similarly, $\tau_i = h \cdot \kappa_i$ where $\kappa_i \in \mathbb{N}$ for $i \neq 0$ denotes the i-th inter-event index, with $\kappa_0 = 0$. As such, using (3-1) we can write the dynamics of the sampled state $\boldsymbol{x}_i$ under an PETC policy as a switched linear (SL) system given by

$$\boldsymbol{x}_{i+1} = \boldsymbol{A}_{\kappa_{i+1}} \boldsymbol{x}_i + \boldsymbol{B}_{\kappa_{i+1}} \boldsymbol{u}_i + \boldsymbol{E} \boldsymbol{w}_{i+1}, \tag{3-7}$$

where the inter-event indices $\kappa_i = (t_i - t_{i-1})/h$ are determined as in (3-6) at execution time. Furthermore, the matrices $\boldsymbol{A}_\kappa$, $\boldsymbol{B}_\kappa$ are given by

$$\boldsymbol{A}_\kappa = \boldsymbol{A}^\kappa, \qquad \boldsymbol{B}_\kappa = \sum_{\ell=0}^{\kappa-1} \boldsymbol{A}^\ell \boldsymbol{B}. \tag{3-8}$$

Because $\boldsymbol{w}[k]$ is an i.i.d. Gaussian random vector with covariance matrix $\boldsymbol{\Sigma}_{\mathrm{w}}$, the load disturbance $\boldsymbol{w}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\mathrm{w},\kappa_i})$, where

$$\boldsymbol{\Sigma}_{\mathrm{w},\kappa} = \sum_{\ell=0}^{\kappa-1} \boldsymbol{A}^\ell \boldsymbol{\Sigma}_{\mathrm{w}} (\boldsymbol{A}^{\mathrm{T}})^\ell. \tag{3-9}$$

Note that the random vectors $\boldsymbol{w}_i$ are no longer identically distributed for all i. However, they remain independent and as such $\mathbb{E}[\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{w}_{i'}] = \boldsymbol{0}$ for all $i \neq i'$. We will revisit (3-7)-(3-9) when constructing our proposed detector $\mathcal{D}$ in §5-2.

Let us first introduce the notation $1 \leqslant \kappa_{\min}$ and $\kappa_{\max} \leqslant \bar{\kappa}$ as the smallest and largest inter-event index for all pairs $(\boldsymbol{x}_0, \boldsymbol{u}_0) \in \mathbb{R}^{n_{\mathrm{x}}} \times \mathbb{R}^{n_{\mathrm{u}}}$, respectively, provided that $\boldsymbol{u}_0$ is the output of the controller $\mathcal{C}$ with state $\boldsymbol{c}[0] \in \mathbb{R}^{n_{\mathrm{c}}}$ to the input $\boldsymbol{y}_0$. Note that these are not tunable design parameters but rather consequences of the choice of PETC policy and controller design $\mathcal{C}$. Whilst they can be hard to obtain *a priori* and are usually only known at execution time, in the case of full-state feedback $\boldsymbol{K}$ the smallest IET $\kappa_{\min}$ can be computed exactly as [32, 61]

$$\kappa_{\min} = \min_{\kappa} \{\kappa \in \mathbb{N} \,|\, \max\{\lambda((\boldsymbol{J}_1^{\kappa-1} \boldsymbol{J}_0)^{\mathrm{T}} \boldsymbol{Q} (\boldsymbol{J}_1^{\kappa-1} \boldsymbol{J}_0))\} > 0\}, \tag{3-10}$$

with $\boldsymbol{J}_0$, $\boldsymbol{J}_1$ as in (A-2b). Furthermore, an upper bound on the largest inter-event index $\bar{\kappa}_{\max}$ can be computed (provided it exist) as [51]

$$\bar{\kappa}_{\max} = \min_{\kappa} \{\kappa \in \mathbb{N} \,|\, \boldsymbol{N}(\kappa) \succ 0\}, \qquad \boldsymbol{N}(\kappa) = \begin{bmatrix} \boldsymbol{A}_\kappa - \boldsymbol{B}_\kappa \boldsymbol{K} \\ \boldsymbol{I} \end{bmatrix}^{\mathrm{T}} \boldsymbol{Q} \begin{bmatrix} \boldsymbol{A}_\kappa - \boldsymbol{B}_\kappa \boldsymbol{K} \\ \boldsymbol{I} \end{bmatrix}. \tag{3-11}$$

This then can be used to set the upper bound $\bar{\kappa}$ without affecting the PETC in any way (see §3-2). This upper bound can, in general, be quite conservative (i.e. $\kappa_{\max} < \bar{\kappa}_{\max}$). Note that there exists a $\kappa' = \bar{\kappa}_{\max} - 1$ for which $\boldsymbol{N}(\kappa')$ is not positive-definite (PD) (meaning for those states $\boldsymbol{q}[k]$ for which $\boldsymbol{q}^{\mathrm{T}}[k] \boldsymbol{N}(\kappa') \boldsymbol{q}[k] \leqslant 0$ the mechanism $\mathcal{E}$ would not have triggered). However, it can be the case that for all those states which satisfy the former, there exists a

$\kappa'' < \kappa'$ for which $\boldsymbol{q}^{\mathrm{T}}[k]\boldsymbol{N}(\kappa'')\boldsymbol{q}[k] > 0$ (meaning that for those states the PETC policy would have already triggered).

It must be noted that any PETC implementation can always be chosen to arbitrarily closely resemble the performance properties of a standard TTC periodic implementation as $\sigma \to 0$ [60]. For this case, the PETC policy reduces (except possibly for a set with zero measure) to TTC policy [68]. This motivates the introduction of the following definition.

Definition 3.2 (Nontrivial PETC, adapted from [36])**.** *Consider the dynamics as in* (3-7) *and suppose* $\boldsymbol{w}_i = \boldsymbol{0}$ *for all* i*. Then, if* σ *and* $\mathcal{C}$ *are constructed such that* $\kappa_{\max} > 1$ *the PETC policy as in* (3-6) *is* nontrivial.

Apart from the choice of (quadratic) triggering condition, and assuming an emulation-based design where h has already been decided, both schemes come with three tunable parameters σ, ϵ, and $\bar{\tau}$ (or $\bar{\kappa}$). The latter of the three usually doesn't impact performance in any way (under Assumption 2), provided it is chosen sufficiently large. The triggering parameter σ must be chosen with considerable care, as it does not only affect the rate of sampling but can lead to a loss of stability if chosen too large (see §3-5). Finally, the margin parameter $\epsilon \in \mathbb{R}_{\geqslant 0}$ can freely be chosen [21, Theorem 2] to trade-off network utilization with loss of performance, and often leads to better results as the state trajectory approaches the origin and disturbances are present [51] (see §5-6). As a final note, $\epsilon > 0$ is often called *mixed* triggering in the literature [14, 68], whilst $\sigma = 0$ is referred to as *Lebesgue* sampling[1] [28, 143].

3-4 Self-triggered control

So far, we have shown that ETC transmits a new measurement once an event has occurred. One disadvantage of both CETC and PETC is that the state of the system still needs to be monitored in between event times, be it continuously or periodically. Furthermore, the sensors need to have sufficient computational capacity to implement the triggering function ϕ. Aimed at achieving the same benefits as ETC, in self-triggered control (STC) the occurrence of the next event time t_{i+1} is predicted at the controller side and transmitted to the sensors, meaning the sensors can idle between sampling instances. This makes STC a *proactive* sampling policy whereas ETC is a *reactive* one. Note that STC is a model-based approach, which often perform better in NCPS [64]. The sequence of triggering times $t_1, t_2, \ldots, t_i$ can be formally defined as [64]

$$\mathcal{S} \quad : \qquad t_{i+1} = t_i + \Gamma(\boldsymbol{x}_i, \boldsymbol{u}_i), \tag{3-12}$$

where $\Gamma : \mathbb{R}^{n_{\mathrm{x}}} \times \mathbb{R}^{n_{\mathrm{u}}} \to \mathbb{R}_{>0}$ is a to be designed event predictor function. A common strategy, also considered here, is to choose Γ to emulate a PETC policy. This then leads to

$$\Gamma(\boldsymbol{x}_i, \boldsymbol{u}_i) = h \cdot \max_{\kappa}\{\kappa \in \mathbb{N}, \kappa \leqslant \bar{\kappa} \,|\, \mathrm{col}(\boldsymbol{\Phi}(\bullet), \boldsymbol{x}_i)^{\mathrm{T}}\boldsymbol{Q}\mathrm{col}(\boldsymbol{\Phi}(\bullet), \boldsymbol{x}_i) \leqslant 0\}, \tag{3-13}$$

[1] Lebesgue sampling is sometimes also called *send-on-delta* [116], although here we reserve that term for triggering conditions of the form $\|\boldsymbol{\chi}(t)\| \geqslant \epsilon$ which are also referred to as *uniform* sampling [62].

where due to the inclusion of $\bar{\kappa}$ we guarantee a finite number of conditions to check. The state evolution function $\boldsymbol{\Phi}(\kappa\,; \boldsymbol{x}_i, \boldsymbol{u}_i)$ derived from (3-7) is given by

$$\boldsymbol{\Phi}(\kappa\,; \boldsymbol{x}_i, \boldsymbol{u}_i) = \boldsymbol{A}_\kappa \boldsymbol{x}_i + \boldsymbol{B}_\kappa \boldsymbol{u}_i. \tag{3-14}$$

An STC policy poses three main advantages compared to PETC:

- **Sparser sampling**: Since the next sampling instance t_{i+1} is calculated in advance, in between update times the sensor can idle potentially saving energy [32] (which is especially beneficial when the nodes are battery-powered [100]).
- **Computational complexity**: The triggering condition and event prediction are checked at the controller side where usually greater computational resources are available [6]. In some applications, it may be unreasonable or impractical to retrofit an existing system with event detectors, and as such an STC policy may be more appropriate [126].
- **Reference tracking**: As discussed in §3-6 STC might be better suited for tracking a reference since $\boldsymbol{r}[k]$ (and future values) are available at the controller side and can be incorporated into the triggering condition[2]. Furthermore, co-design methods might be easier as the next sampling time and the actuation input can be jointly synthesized [53].

Furthermore, an STC policy gives the control system direct control over when to sample next, which forms the basis for our proposed watermarking scheme (see §5-3). This comes with the added benefit that the event predictor function Γ can thus also be interchanged at execution time, which will be further discussed in §5-6.

We are now ready to introduce our considered architecture, which can be seen in Figure 3-2. The design of the detector $\mathcal{D}$ will be discussed in §5-2. Here, $[\![\bullet]\!]$ denotes a watermarked signal (see Chapter 5). The augmented STC watermarking strategy will be discussed in §5-3. The digital controller $\mathcal{C}$ as in (2-2), adapted for reference tracking, is given by

$$\mathcal{C} \quad : \quad \begin{aligned} \boldsymbol{c}[k+1] &= \boldsymbol{A}_\mathrm{c}\boldsymbol{c}[k] + \boldsymbol{B}_\mathrm{c}(\boldsymbol{r}[k_i] - \hat{\boldsymbol{x}}_i), \\ \boldsymbol{u}[k] &= \boldsymbol{C}_\mathrm{c}\boldsymbol{c}[k] + \boldsymbol{D}_\mathrm{c}(\boldsymbol{r}[k_i] - \hat{\boldsymbol{x}}_i). \end{aligned} \qquad k_i \leqslant k < k_{i+1}, \qquad \begin{aligned} &\text{(3-15a)} \\ &\text{(3-15b)} \end{aligned}$$

Here, $\boldsymbol{x}_i \in \mathbb{R}^{n_\mathrm{x}}$ denotes the state estimate (see §3-7) and $\boldsymbol{r}[k] \in \mathbb{R}^{n_\mathrm{x}}$ denotes the reference signal to be tracked. To summarize, the sampled outputs are only transmitted to the controller at the occurrence of an event with an in ETC policy, whilst for STC when to sample next is decided on the controller side. Therefore, STC also permits greater control of the sampling periods, at the cost of complicating the stability analysis (see §3-5). We will revisit this in §5-3.

[2] This property of STC bears some resemblance with the *preview capabilities* of model predictive control (MPC) [38].

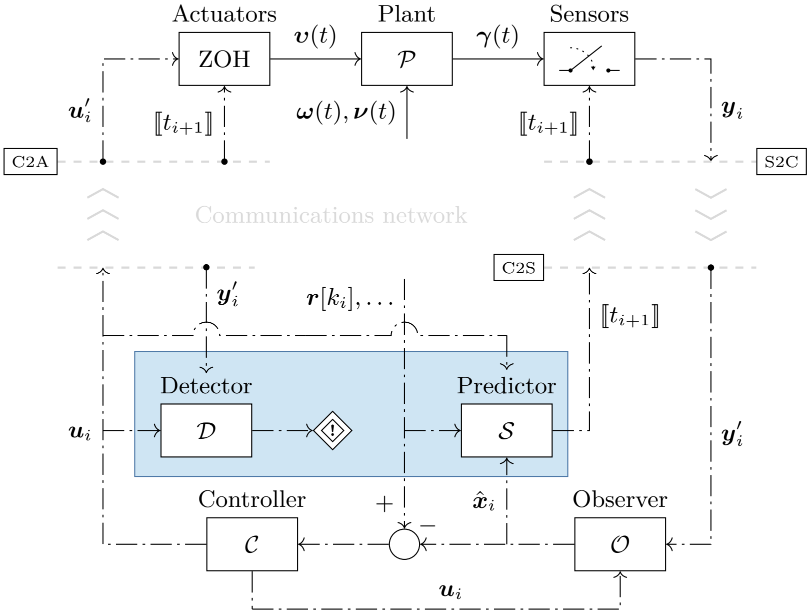


Figure 3-2: The considered NCPS architecture. Novel contributions are highlighted in blue.

3-5 Stability and performance

As with any control strategy, it is of vital importance whether the closed-loop implementation will be stable. Since we are dealing with a sampled-data system, we aim to give stability guarantees of the continuous-time closed-loop system [78]. Below, we reiterate the definition of globally exponentially stable (GES) for continuous-time systems with (augmented) state vector $\boldsymbol{\xi}(t)$. Note that since we are considering stability, this is equivalent to a zero reference $\boldsymbol{r}[k] = \boldsymbol{0}$, for all k (i.e. regulation).

Definition 3.3 (Globally exponentially stable (GES) [63, Definition II.2])**.** *The closed loop system as in* Figure 3-2 *is said to be* GES *if there exist constants* $\gamma, \rho \in \mathbb{R}_{>0}$ *such that*

$$\|\boldsymbol{\xi}(t)\| \leqslant \gamma \cdot \|\boldsymbol{\xi}(0)\| \cdot e^{-\rho \cdot t}, \qquad \forall t \geqslant 0 \tag{3-16}$$

holds for all $\boldsymbol{\xi}(0) \in \mathbb{R}^{n_\xi}$*. Here,* ρ *is called (a lower bound on) the* decay rate [58] *and* γ *is called the* gain [33].

If the load disturbances and measurement noise are absent, the STC policy as proposed here has an equivalent PETC implementation [32]. Then, borrowing from (3-1) we introduce the augmented state vector $\boldsymbol{q}[k]$ such that $\boldsymbol{q}[k] = \boldsymbol{\xi}(h \cdot k)$. Note that in this case the limits $\lim_{t\to\infty} \boldsymbol{\xi}(t) = \boldsymbol{0}$ and $\lim_{k\to\infty} \boldsymbol{q}[k] = \boldsymbol{0}$ are equivalent [65], as proven in the hybrid formulation in [44, Theorem 4], and GES of the PETC guarantees stability of the continuous-time closed-loop system [60].

The aim is to determine *a priori* whether a given triggering matrix $\boldsymbol{Q}$ (parameterized by a triggering parameter σ) provides stability. Of the several modeling choices available, a piecewise linear (PWL) model is the least conservative in providing stability guarantees [63]. We introduce the augmented state vector $\boldsymbol{q}[k+1] = \mathrm{col}(\boldsymbol{x}[k], \boldsymbol{c}[k], \boldsymbol{y}_i, \boldsymbol{u}_i)$, with $k_i \leqslant k < k_{i+1}$, following a similar approach as in [50] where we rewrite the centralized synchronous PETC policy as [32]. We do this by combining (3-1) and (2-2) which gives

$$\boldsymbol{q}[k+1] = \begin{cases} \boldsymbol{A}_0\boldsymbol{q}[k] + \boldsymbol{F}_0\begin{bmatrix} \boldsymbol{w}[k] \\ \boldsymbol{v}[k] \end{bmatrix} & \boldsymbol{q}^{\mathrm{T}}[k]\bar{\boldsymbol{Q}}\boldsymbol{q}[k] \leqslant 0, \\ \boldsymbol{A}_1\boldsymbol{q}[k] + \boldsymbol{F}_1\begin{bmatrix} \boldsymbol{w}[k] \\ \boldsymbol{v}[k] \end{bmatrix} & \boldsymbol{q}^{\mathrm{T}}[k]\bar{\boldsymbol{Q}}\boldsymbol{q}[k] > 0, \end{cases} \tag{3-17}$$

where the matrices $\boldsymbol{A}_0$, $\boldsymbol{A}_1$ are given by

$$\boldsymbol{A}_0 = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B} \\ \boldsymbol{0} & \boldsymbol{A}_{\mathrm{c}} & -\boldsymbol{B}_{\mathrm{c}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \qquad \boldsymbol{A}_1 = \begin{bmatrix} \boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}_{\mathrm{c}}\boldsymbol{C} & \boldsymbol{B}\boldsymbol{C}_{\mathrm{c}} & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{B}_{\mathrm{c}}\boldsymbol{C} & \boldsymbol{A}_{\mathrm{c}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{C} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{D}_{\mathrm{c}}\boldsymbol{C} & \boldsymbol{C}_{\mathrm{c}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \tag{3-18}$$

and the additive noise matrices $\boldsymbol{F}_0$, $\boldsymbol{F}_1$ and augmented triggering matrix $\bar{\boldsymbol{Q}}$ are given by

$$\boldsymbol{F}_0 = \begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}, \qquad \boldsymbol{F}_1 = \begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{B}_{\mathrm{c}} \\ \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & -\boldsymbol{D}_{\mathrm{c}} \end{bmatrix}, \qquad \boldsymbol{Q} = \begin{bmatrix} (1-\sigma^2)\cdot\boldsymbol{I} & \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}. \tag{3-19}$$

In case $\boldsymbol{w}[k] = \boldsymbol{0}$, $\boldsymbol{v}[k] = \boldsymbol{0}$ for all k, GES is guaranteed by the following theorem.

Theorem 3.2 (GES of PWL system, adapted from [58, Theorem 6.2])**.** *Consider the* PWL *system in* (3-17) *and suppose* $\boldsymbol{P}_0, \boldsymbol{P}_1 \succ 0, \alpha_{ij}, \beta_{ij}, \pi_i, \rho \in \mathbb{R}_{>0}$ *with* $i, j \in \{0, 1\}$ *are the optimal solution to*[a]

$$\max \rho \qquad \text{s.t.} \tag{3-20a}$$

$$e^{-2\cdot\rho\cdot h}\cdot\boldsymbol{P}_i - \boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{P}_j\boldsymbol{A}_i + (-1)^i\cdot\alpha_{ij}\cdot\bar{\boldsymbol{Q}} + (-1)^j\cdot\beta_{ij}\cdot\boldsymbol{A}_i^{\mathrm{T}}\bar{\boldsymbol{Q}}\boldsymbol{A}_i \succcurlyeq 0, \tag{3-20b}$$

$$\boldsymbol{P}_i + (-1)^i\cdot\pi_i\cdot\bar{\boldsymbol{Q}} \succ 0. \tag{3-20c}$$

Then, the closed-loop system as in Figure 3-2 *is* GES *with (lower bound on the) decay rate* ρ*.*

[a] Note that (3-20b), (3-20c) are bilinear matrix inequalities (BMIs), which are normally intractable, but can be computed employing a line search or bisection method over ρ.

In the case of additive perturbations (i.e. $\boldsymbol{\delta}(t) \neq \mathbf{0}$) one can use a (hybrid) linear impulsive (LI) system model as in [50]. Such a system formulation for the architecture considered in Figure 3-2, as well as the corresponding linear matrix inequality (LMI) conditions that need to be checked, can be found in Appendix A-1.

Ideally, when additive Gaussian noise is present we would also like to confirm a type of stochastic stability *a priori*. A natural choice of stability metric is that of mean square stable (MSS) which we define below.

Definition 3.4 (Mean square stable (MSS), adapted from [26, Definition 6.C.3])**.** *The closed-loop control system as in* Figure 3-2 *with extended state vector* $\boldsymbol{q}[k] = \operatorname{col}(\boldsymbol{x}[k], \boldsymbol{c}[k], \boldsymbol{y}_i, \boldsymbol{u}_i)$ *for* $k_i \leqslant k < k_{i+1}$ *is said to be* mean square stable (MSS) *if their exist a class* $\mathcal{KL}$ *function* β *and constant* $\delta \in \mathbb{R}_{>0}$ *such that*

$$\mathbb{E}[\,\|\boldsymbol{q}[k]\|^2] \leqslant \beta\Big(\mathbb{E}[\,\|\boldsymbol{q}[0]\|^2], k\Big) + \delta, \qquad k \geqslant 0 \tag{3-21}$$

holds for all $\boldsymbol{q}[0] \in \mathbb{R}^{n_\mathrm{x}+n_\mathrm{c}+n_\mathrm{y}+n_\mathrm{u}}$.

The implication of Definition 3.4 is that the covariance of the state vector is finite. To the best of the author's knowledge, no results on MSS are available for the architecture considered here, that being an STC policy and a dynamic controller. The results presented in [26, 6.C.2] do provide a proof for sufficient conditions of MSS of PETC systems, but only for the case of full-state feedback $\boldsymbol{K}$ and triggering condition $\|\boldsymbol{x}_i - \boldsymbol{\chi}(t)\|_\infty > \epsilon$ (i.e. modified *Lebesgue* sampling). Possibly, these results could be extended to dynamic controllers and general quadratic triggering conditions, and maybe under additional (more restrictive) assumptions to STC, but no proof is provided at this time. A sketch of a proof for tackling the more general case, but only for PETC, is provided in Appendix A-3-1. Therefore, we must introduce the following assumption.

Assumption 5. *The closed-loop system as in* Figure 3-2 *is* MSS*.* ◊

It must be noted that most of the results, as presented above, do not apply to the closed-loop architecture 3-2 in its entirety. As such, trade-offs must be made in the form of simplified architectures to provide the relevant guarantees *a priori*. As such, in Chapter 5 we will consider a dynamic controller but with full-state feedback $\boldsymbol{C} = \boldsymbol{I}$ (meaning no observer is present), whilst in Chapter 6 we will consider a static controller but with output-feedback and an observer $\mathcal{O}$, the design of which will be elaborated on in §3-7.

3-6 Triggering condition

In this section, we propose a new triggering quadratic triggering condition suitable for reference tracking. The relevance of quadratic triggering conditions has been argued in [63], and a (non-comprehensive) overview is given in e.g. [61, 100]. The reference signal $\boldsymbol{r}[k]$ is assumed to be periodic (see §5-1) and known in advance.

It is well known that many signals in real processes are periodic and as such, periodic reference tracking is a common task for many practical systems [84]. Examples of applications are power supplies, robot manipulators, mechatronic rotary systems, and piezoelectric actuators [106]. The branch of control theory dealing with such periodic references (and periodic disturbances) is named *repetitive control*.

In §3-2 a quadratic triggering condition of described by (3-4) was introduced. Here, we propose a triggering condition suitable for reference tracking, given by

$$\|\boldsymbol{x}_i - \boldsymbol{x}[k]\| > \sigma \cdot \|\boldsymbol{r}[k] - \boldsymbol{x}[k]\| + \epsilon. \tag{3-22}$$

Note that both σ and ϵ can in principle be both time-varying and state-dependent (see §5-6). We can show that (3-22) is indeed a quadratic triggering condition by the introduction of the augmented state vector $\boldsymbol{q}[k] = \text{col}(\boldsymbol{x}[k], \boldsymbol{x}_i, \boldsymbol{r}[k], \epsilon)$ for $k_i \leqslant k < k_{i+1}$. The triggering condition (3-22) can be written as

$$\boldsymbol{q}^{\mathrm{T}}[k]\boldsymbol{Q}\boldsymbol{q}[k] > 0, \qquad \boldsymbol{Q} = \begin{bmatrix} (1-\sigma^2)\cdot\boldsymbol{I} & -\boldsymbol{I} & \sigma\cdot\boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{I} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \sigma\cdot\boldsymbol{I} & \boldsymbol{0} & -\sigma\cdot\boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & 1 \end{bmatrix}. \tag{3-23}$$

The rationale behind this triggering condition is its resemblance to the widely used *trigger-on-relative-error* triggering condition given by $\|\boldsymbol{x}[k] - \boldsymbol{x}_i\| > \sigma \cdot \|\boldsymbol{x}[k]\|$ [61]. In fact, when $\boldsymbol{r}[k] = \boldsymbol{0}$, for all k (regulation) and $\epsilon = 0$ the triggering condition (3-22) becomes identical to the former. Other similarities include that when the state vector $\boldsymbol{x}[k]$ is far away from the reference, the <u>r</u>ight-<u>h</u>and <u>s</u>ide (RHS) of (3-22) will be larger, leading to sparser triggering. This is desirable as the trajectory mainly needs to get closer to the reference and thus coarse adjustments suffice. As the error decreases, more subtle adjustments in the actuation input are required, which is achieved through a diminishing RHS.

The proposed triggering condition is in parts similar to the one proposed in [111], albeit their triggering condition (and neglecting the disturbance dynamics) can be written as

$$\|\boldsymbol{x}_i - \boldsymbol{x}[k]\|_{\boldsymbol{W}_{\mathrm{x}}} > \|\boldsymbol{r}[k] - \boldsymbol{r}[k_i]\|_{\boldsymbol{W}_{\mathrm{r}}}, \tag{3-24}$$

where $\boldsymbol{W}_{\mathrm{r}} \succ 0$ and $\boldsymbol{W}_{\mathrm{x}} \succ 0$ are weighting matrices to be designed. Finally, note that whilst the numerical results as in §5-5 use the triggering condition as in (3-22) our main findings extend to general quadratic triggering conditions as well. To summarize, the STC controller mechanism is given by (3-12), with Γ defined as in (3-13) and quadratic triggering condition $\boldsymbol{Q}$ as in (3-23). For the stability analysis as performed in §3-5, one can take $\epsilon = 0$, $\boldsymbol{r}[k] = 0$ for all k and proceed with an identical analysis to conclude GES from there.

In this work, we are considering synchronous STC where, in conjunction with Assumption 2, the measurement output and actuation input are updated at the same time. The triggering condition $\boldsymbol{Q}$ as proposed in §3-6 is a modification of *trigger-on-relative-state-error*. Our framework is in general not as restrictive and by modifying the triggering matrix $\boldsymbol{Q}$, an STC policy which takes into account both state and actuation error[3] can easily be incorporated as well (see e.g. [61, 50]).

3-7 Observer design

Since STC predicts the evolution of the state vector, it relies on either full-state feedback or observer-based output feedback. In this thesis, both scenarios are explored. In the case of the latter, the full-order observer dynamics as shown in Figure 3-2 are given by

$$\mathcal{O} \quad : \qquad \hat{\boldsymbol{x}}_{i+1} = \boldsymbol{A}_{\kappa_{i+1}}\hat{\boldsymbol{x}}_i + \boldsymbol{B}_{\kappa_{i+1}}\boldsymbol{u}_i + \boldsymbol{L}_{\kappa_{i+1}}(\boldsymbol{y}_i - \boldsymbol{C}\hat{\boldsymbol{x}}_i), \tag{3-25}$$

where $\hat{\boldsymbol{x}}_i$ denotes the i-th state estimate. Similar to [2], here we are not concerned whether the observer as in Theorem 3.3 is optimal in any sense, as we are only interested in stability properties. More refined time-varying observers such as the one proposed in [4] can also be considered. Furthermore, in the presence of bounded perturbations $\boldsymbol{\delta}(t)$, extensions of the results as presented in §5-1 to observers such as the ellipsoidal-based design proposed in [21], and the special observer proposed in [50] are an interesting future direction.

The aim is to design an event-triggered observer, where we share the contention of [6] that it is better to place the observer on the controller node rather than a sensor node, since the former is where greater computational resources are usually available. For simplicity, we are considering a stationary filter as in [8]. Note that we are interested in preserving stability, which can be guaranteed by the following theorem.

Theorem 3.3 (Switch-invariant observer gain [2])**.** *Consider the closed-loop dynamics as in* Figure 3-2 *and suppose* Assumption 1-3 *hold and* $0 \prec \boldsymbol{W}_1 \in \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{x}}$, $\boldsymbol{W}_2 \in \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{y}}$ *are the optimal solutions to*

$$\min \operatorname{tr}(\boldsymbol{W}_1) \qquad s.t. \tag{3-26a}$$

$$\begin{bmatrix} \boldsymbol{W}_1 - \boldsymbol{I} & \boldsymbol{A}_\kappa^\mathrm{T}(\boldsymbol{W}_1 - \boldsymbol{W}_2\boldsymbol{C})^\mathrm{T} \\ \star & \boldsymbol{W}_1 \end{bmatrix} \succcurlyeq 0, \qquad 1 \leqslant \kappa \leqslant \bar{\kappa}. \tag{3-26b}$$

Then, $\boldsymbol{L}_\kappa = \boldsymbol{W}_1^{-1}\boldsymbol{W}_2$*, for all* κ *is a static observer gain which renders the observation errors* $\tilde{\boldsymbol{x}}_i = \hat{\boldsymbol{x}}_i - \boldsymbol{\chi}(t_i)$*, with* $\hat{\boldsymbol{x}}_i$ *as in* (3-25)*,* GES [4, Lemma 3]*.*

Note that (3-26) might be infeasible for the chosen $\bar{\kappa}$, which implies a smaller $\bar{\kappa}$ must be chosen. Following the approach as in [6, Assumption 1], we here propose an identical constraint. Let us define the subsystem $\underset{\sim}{\mathcal{G}}$ as the interconnection between the plant $\underset{\sim}{\mathcal{P}}$ (with $\boldsymbol{C} = \boldsymbol{I}$), the sensors and actuators, the digital controller $\mathcal{C}$ and the STC policy $\mathcal{S}$. We make the following assumption.

[3] This is in fact another benefit of STC, since the triggering is centralized and access to the current output $\boldsymbol{u}[k]$ is available (which is not necessarily the case for PETC).

Assumption 6. *The subsystem $\underset{\sim}{\mathcal{G}}$ is* input-to-state stable (ISS) *with respect to observation errors $\tilde{\boldsymbol{x}}$.* $\Diamond$

We postulate that the closed-loop system as in Figure 3-2 can be shown to be ISS to observation errors. A sketch of the proof is provided in Appendix A-3-2. However, at this time no complete proof is provided, which is left to future work. Note that in the case of a full-state feedback controller $\boldsymbol{K}$ the results in [5] can be used directly and Assumption 6 is guaranteed to hold. We are in general interested in dynamic controllers $\mathcal{C}$, as proportional–integral–derivative (PID) controllers are one of the most common types of controllers in industrial control systems (ICSs) [62, 133], especially for reference tracking. However, the majority of the literature on STC has been written for full-state feedback controllers and as such, stronger results are available for these types of systems. Nonetheless, reference tracking using a proportional gain is a form of full-state feedback (e.g. *proportional-only* controllers), which are still widely used in industry [35]. Therefore, results for full-state feedback (with application to reference tracking) are interesting in their own right.

Combining the results from Theorem 3.3 and Assumption 6 we can use the result of [5, Theorem 4], which utilizes the fact that the cascade of two ISS systems is itself ISS, and conclude that the closed-loop system as in Figure 3-2 is GES. As discussed in §3-5, GES stability in the absence of both load disturbances and measurement noise might imply MSS of the STC in the presence of noise. Here, we only demonstrate MSS qualitatively through a numerical simulation (see Chapter 5). Finally, in line with [21, Remark 6], these results can be extended to general quadratic triggering conditions $\boldsymbol{Q}$. The specific choice for (3-23) is mostly for convenience and demonstrative purposes.

Chapter 4

Adversary model

Recent years have shown increased vulnerability of networked cyber-physical systems (NCPSs) to malicious cyberattacks. The rise of malware and computer worms, with prominent examples being Stuxnet [37, 105], BlackEnergy [75, 128], and INDUSTROYER [20], specifically targeted at industrial control systems (ICSs) is worrisome [107]. Furthermore, since many cyber-physical systems (CPSs) use of-the-shelve information technology (IT) solutions, even for critical infrastructure, this can lead to additional vulnerabilities [109]. As such, in addition to providing safety in the control system, there is a necessity to also provide *security.*

Secure control borrows substantially from the field of fault-tolerant control. However, the detection of maleficent attacks is arguably more challenging than fault detection [37]. For one, different from fault diagnostics, attacks might not follow a (discernible) statistical pattern and can be either partly or fully deterministic [24] (see e.g. §6-3). Furthermore, the adversary is usually intelligent and can actively try to deceive any detector logic present in the control system [37] (see e.g. §5-1). Finally, the possibility of a network being compromised means the control system is dealing with a persistent *fog-of-war*, and can not rely on knowledge of

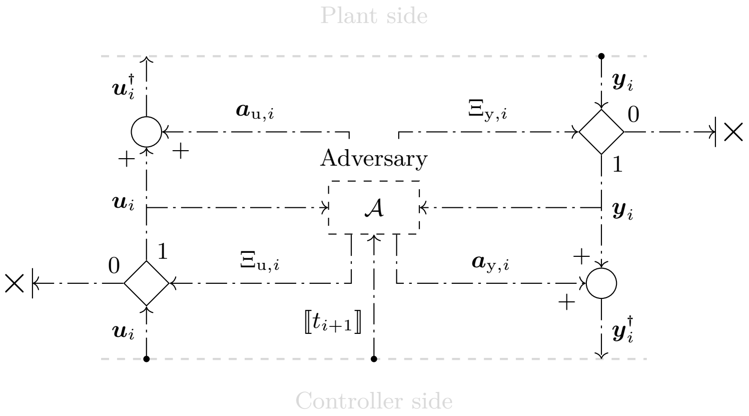


Figure 4-1: Considered adversary model, adapted from [108, Figure 4]

events at the plant side. As such, attack detection solely based on the received measurement is the only scheme with practical viability, which is not always taken into consideration in the literature.

As an NCPS contains a (wireless) communications network, its susceptibility to cyberattacks is heightened. In this section, we will discuss the capabilities at the disposal of an intelligent adversary and the restrictions imposed on them. In general, an adversary model is a way of restricting the scope of the problem and is necessary if any insight is hoped to be achieved [15]. Importantly, a NCPSs can thus only be secure as long as our adversary model and trust assumptions are satisfied in practice.

The adversary model $\underset{\sim}{\mathcal{A}}$ (in its most general form) considered here can be seen in Figure 4-1. The signals $\boldsymbol{a}_{\mathrm{u},i}$, $\boldsymbol{a}_{\mathrm{y},i}$ denote the attack vectors acting on the input and output respectively, which can be chosen by the adversary to construct an attack. The signals $\Xi_{\mathrm{u},i}, \Xi_{\mathrm{y},i} \in \{0,1\}$ denote jamming signals of the C2A and S2C channels respectively, and are usually subject to energy constraints [136]. Finally, note that the adversary $\underset{\sim}{\mathcal{A}}$ has access to the next sampling instance $\llbracket t_{i+1} \rrbracket$ as transmitted by the self-triggered control (STC) mechanism.

The model in Figure 4-1 captures the broadest class of possible adversaries. Here, the dashed line around the adversary denotes the uncertainty about the presence of the adversary. We refer to this as a Byzantine adversary model, the definition of which is given below. This indicates one of the fundamental challenges in secure control, namely the detection of an attack (and thereby confirming the presence of an adversary). Furthermore, together with assumptions on the capabilities of the adversary allow us to further classify said adversaries.

Definition 4.1 (Byzantine adversary model, adapted from [88])**.** *Consider the adversary $\underset{\sim}{\mathcal{A}}$ as in* Figure 4-1 *and suppose the adversary satisfies (some of) the following conditions:*

I. ***Complete system knowledge:*** *The adversary is omniscient and knows the exact plant dynamics and controller architecture $\underset{\sim}{\mathcal{P}}$, $\mathcal{C}$, $\mathcal{S}$, and $\mathcal{O}$* [95, 92].

II. ***Detection procedure:*** *The adversary is aware of the existence and model of any detector $\mathcal{D}$* [56].

III. ***Perfect disclosure:*** *The adversary has access to all real-time measurements and control inputs $\boldsymbol{u}_i$, $\boldsymbol{y}_i$* [80, 16, 56].

IV. ***Arbitrary modifications:*** *The adversary is able to modify the signals $\boldsymbol{u}_i^{\prime}$, $\boldsymbol{y}_i^{\prime}$ arbitrarily* [56, 92].

When $\underset{\sim}{\mathcal{A}}$ satisfies all of the former conditions we call it a strong Byzantine adversary*. If it only satisfies conditions III and IV we call $\underset{\sim}{\mathcal{A}}$ a* weak Byzantine adversary*.*

It might sound unreasonable to expect an adversary to have full model knowledge. However, the threat of strong Byzantine adversaries should not be taken lightly, as data breaches through not only phishing [19] but also through specialty-crafted malware (e.g. Havex [40]), are relatively common [114]. Furthermore, the insider threat is critical in large infrastructures,

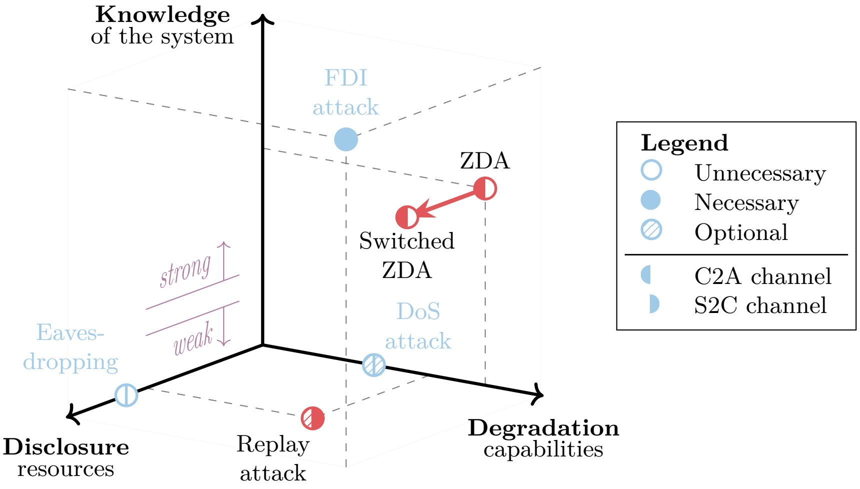


Figure 4-2: Attack space with common and proposed attacks, adapted from [124, Figure 1]

as these systems usually involve many employees [91]. This has been exemplified in the Maroochy sewage spill in 2000 [117, 56]. Problematically, such insiders do not only possess detailed knowledge of the control systems but often also have the engineering know-how on how to craft sophisticated attacks.

We model the attacks as occurring at the network level, but attacks might alternatively utilize compromised target endpoints (i.e. sensors, actuators, or even controllers). In practice, this scenario is actually more common as malware installed on system hardware is often where attacks originate [67]. Precisely these target endpoints are usually the weakest link in an NCPS due to the use of commodity IT product [118, 104]. The latter does not alter the applicability of our proposed method in any way, and therefore we focus on the former (and most common in CPS literature) modeling paradigm.

4-1 Types of attacks

With the distinction between adversaries made in Definition 4.1, common attack types as mentioned in the literature can be categorized, which are depicted in Figure 4-2. The two types of attacks highlighted in red denote a replay attack as discussed in Chapter 5, and a modified zero dynamic attack (ZDA), called a switched ZDA, which is elaborated on in Chapter 6. The threat of eavesdropping and possible countermeasures can be found in e.g. [140, 20]. Denial-of-services (DoSs) on event-triggered control (ETC) control systems are discussed in e.g. [30, 121], and more information on false data injection (FDI) is provided in e.g. [89, 47].

Inspired by [124], to classify both the system knowledge and disclosure resources of the adversary as depicted Figure 4-2, we borrow the concept of information $\mathbb{I}_{\mathrm{a}}(t)$ from a game theoretical framework. The set $\mathbb{I}_\mathrm{a}(t)$ represents all the information available to the adversary $\mathcal{A}$ at time instance t. Naturally, for a strong Byzantine adversary we have $\{\mathcal{P}, \mathcal{C}, \mathcal{D}\} \subset \mathbb{I}_\mathrm{a}(t)$ for all t. The information needed corresponding to respective attacks is described in Chapter 5 and Chapter 6.

4-2 Objective and constraints

With the assumptions on $\mathcal{A}$ in place, we must naturally introduce an objective and any possible constraints of an attack as constructed by the adversary. Here, we consider a similar objective to that in [57, 67] for which we introduce the definition of a safe region.

Definition 4.2 (Safe region)**.** *A* safe region $\mathbb{X}_\mathrm{s} \subset \mathbb{R}^{n_\mathrm{n}}$ *is a bounded set that ensures the safety. Under nominal system operations* $\boldsymbol{\chi}(t) \in \mathbb{X}_\mathrm{s}$ *for all* $t \geqslant t_0$[a]*.*

[a] Note that in the presence of load disturbances or measurement noise, the above definition cannot be guaranteed and a probabilistic model should be used instead, i.e. $\mathbb{P}[\boldsymbol{\chi}(t) \in \mathbb{X}_\mathrm{s}] \geqslant p_\mathrm{s}$.

The introduction of a safe region is natural in a practical control framework, as the complement of such a region might represent states in which, for example, the pressure of a holding vessel will exceed its pressure rating or the level of a liquid in a tank exceeds its capacity [67].

As is common in secure control literature [91], we assume the system has been in operation for a sufficiently long time such that we can model the system as having been initialized at $t = -\infty$, which is reasonable as control systems usually run for a long time [90]. Without loss of generality, we assume an attack occurs no earlier than $t_0 = 0$. Since we propose a residual-based detector $\mathcal{D}$ (see §5-2), the notion of an *ϵ-stealthy attack* is repeated here in light of the proposed framework.

Definition 4.3 (ϵ-stealthy attack, adapted from [123])**.** *An attack is said to be ϵ-stealthy with respect to a detector $\mathcal{D}$ if $\|\boldsymbol{z}_i\| \leqslant \epsilon$ for the entire attack duration.*

Here, $\boldsymbol{z}_i = \boldsymbol{y}_i - \boldsymbol{C}\boldsymbol{d}_i$ denotes the residual from the detector (see §5-2). Similar to the frameworks of [70, 69] we can state the goals of our adversary as follows:

- **Objective**: Construct a *disruptive* attack such that for some finite $t > t_0$ the state vector $\boldsymbol{\chi}(t) \notin \mathbb{X}_\mathrm{s}$ as by Definition 4.2.
- **Constraint**: The attack must remain *stealthy* as per Definition 4.3 for the entire duration of the attack or until the former objective has been accomplished.

We denote an attack as being successful if the objective is fulfilled without violating the constraint. Note that in the case of load disturbances $\dot{\boldsymbol{\omega}}(t)$ there is always a non-zero chance that the state trajectory leaves the safe region $\mathbb{X}_\mathrm{s}$, making Definition 4.2 ambiguous. As such, guarantees will be given in the absence of noise in Chapter 6 and in Chapter 5, we will merely provide illustrative examples of successful attacks (although these can be extended to probabilistic guarantees). Finally, to avoid detection stemming from secondary sources (e.g. physical inspection, system resets), the attack should preferably be carried out in the shortest time possible. This could be an additional performance metric for constructing efficient attacks, but such extensions are left to future work (see §7-2).

Finally, we summarize our contributions here and categorize them relative to existing countermeasures in the literature (see §5-1-1 and §6-2-2). We extend the framework for secure control from [124] with the added definition of *isolation*, as proposed in [41]. These results can be seen in Table 4-1. Note that attack mitigation, which is supposed to ensure graceful degradation of the systems until the attack has subsided, has received little to no attention in the literature.

Table 4-1: Categorization of attacks, adapted from [124]

	Detection	Isolation	Mitigation	Prevention
Replay attacks	§5, [90, 47]	[41]		
(Switched) ZDAs	[95, 123]			§6, [69]

Chapter 5

A self-triggered control watermarking scheme

In this section, we investigate the threat posed by replay attacks to control systems, and we propose a novel self-triggered control (STC) watermarking scheme. At the end of this chapter, we provide illustrative examples to support our findings.

Replay attacks are of particular interest since they are simple to implement by an adversary without advanced knowledge of the system or skills to decrypt messages [131]. Furthermore, according to [105] the infamous STUXNET worm remained hidden for long periods of time in part because it utilized a replay attack. As such, these types of attacks have received considerable attention in the literature.

Control systems are especially vulnerable to replay attacks during steady-state operation, as past and future outputs are (statistically) indiscernible, and while tracking a periodic reference. The latter is a common task in industrial control systems (ICSs) (see §3-6) and as such will be the main focus. Replay attacks whilst the system is in steady state are further discussed in §5-6.

5-1 Replay attacks

A replay attack is an attack where the adversary replays past recorded outputs to the controller, whilst disregarding the true current outputs of the plant. If feedback is employed (as is the case here), then these false sensor measurements will lead to corrupted input signals to the actuator [67], which can push the state trajectory outside the safe region $\mathbb{X}_s$. Sometimes, adversaries are considered who have disruption capabilities of the C2A channel (see e.g. [41]), but this is in general not necessary as even stable systems are susceptible to *disruptive* replay attacks (see Figure 4-2). Below, we give the definition of a replay attack considered here, adapted to an STC system as outlined Figure 3-2.

Definition 5.1 (Replay attack)**.** *During a replay attack*[a] *initiated at time* $T_\mathrm{a} > t_0$*, the measurement outputs as received by the controller are given by*

$$\boldsymbol{y}'_i = \boldsymbol{y}_{i-\Delta i}, \qquad \forall t_i \geqslant T_\mathrm{a}, \tag{5-1}$$

where Δi *is the loop (or delay) length.*

[a] More accurate would be to call this a *delay attack* (see e.g. [7]), but the extension to a more sophisticated scheme where past data is replayed on a loop (see e.g. [41, 9]) is straightforward. Considering that the objective of the attacker is to leave the safe set $\mathbb{X}_\mathrm{s}$ in finite time, choosing a large enough Δi is sufficient for demonstrative purposes.

Different from time-triggered control (TTC), where the adversary can replay the outputs at a predetermined time, we here consider a more *compliant* replay attack where the adversary replays a measurement at the time $\llbracket t_{i+1} \rrbracket$ as requested by the STC policy. Note that our proposed detection method does not exploit the consistency of these inter-event times (IETs), and thus an adversary not complying with the next sampling instance will (in principle) not be detected based on that discrepancy. This is, however, desirable from a robust and networked control perspective, as even in the absence of an adversary the sensors might not be able to comply (exactly) with the next IET as communication delays and package drops are bound to occur [127].

In this chapter, an adversary is considered who has taken control of the S2C channel (see Figure 3-2) and can both eavesdrop as well as manipulate the packages sent over the channel. Furthermore, the adversary is a weak Byzantine adversary, as captured in the following assumption.

Assumption 7. *The information available to the adversary* $\underset{\sim}{A}$ *at time* $t \in [t_i, t_{i+1})$ *is given by* $\mathbb{I}_\mathrm{a}(t) \supseteq \{t_0, \ldots, t_i \wedge \boldsymbol{y}_0, \ldots, \boldsymbol{y}_i\}$. ◊

The two parameters T_a and Δi need to be (correctly) chosen by the adversary to avoid detection (see §5-5). Specifically, given a periodic reference to be tracked with period T_r, we have that $\Delta T = \sum_{i=1}^{\ell \cdot \Delta i} \tau_i \approx T_\mathrm{r}$, $\ell \in \mathbb{N}$, i.e. the loop length must be approximately equal to an integer multiple of the period of the reference signal. Note that by Assumption 7 an adversary can (approximately) recover T_a from the observer outputs $\boldsymbol{y}_0, \ldots, \boldsymbol{y}_i$ and τ_i from the past event times t_i. The effect of Δi on the success of the attack is further elaborated on in Appendix A-4.

Note that here we assume the communication channels are unencrypted, as is the *status quo* in networked cyber-physical system (NCPS) [39, 118] (as most messages in control systems are not confidential [118]). If the S2C channel is encrypted, replay attacks are still possible[1]. In that case, we assume that $\mathbb{I}_\mathrm{a}(t) = \{\hat{T}_\mathrm{r} \wedge t_0, \ldots, t_i \wedge \mathfrak{y}_0, \ldots, \mathfrak{y}_i\}$, where $\hat{T}_\mathrm{r}$ denotes an estimate of the reference period (which an attacker might obtain from knowledge of the process) and $\mathfrak{y}_i$ denotes the i-th encrypted sampled output. Note that t_i would still be available to the adversary, not from the contents of the packages, but based solely on the time $\mathfrak{y}_i$ is transmitted. Basically, whilst encryption provides confidentiality of the messages, it does not provide secrecy, which the adversary can exploit. In the following, we show that Assumption 7 (or the relaxed one here) is sufficient to construct a replay attack.

5-1-1 Existing countermeasures

There exist several proposed countermeasures for detecting replay attacks in the literature. For instance, [34, 76] propose a symmetric linear encryption and decryption scheme for the detection of what they call generalized replay attacks (more similar to false data injection (FDI) attack, see Figure 4-2). In [105], an inversion-based watermarking scheme for detecting replay attacks during reference tracking is proposed. However, they do not consider load disturbance or measurement noise. An event-triggered watermarking strategy for constant references is proposed in [9], but they restrict themselves to a triggering condition of the form $\|\boldsymbol{x}[k] - \boldsymbol{r}\| \geqslant \epsilon$ and again only consider the deterministic case. Here, we will consider two prominent forms of watermarking, namely additive and multiplicative watermarking, which we introduce next.

Additive watermarking as first proposed in [90] was one of the first deterrents against replay attacks. In (dynamic) additive watermarking, a watermarker $\mathcal{W}$ (see Figure 5-1a) is added after the controller, which is given by

$$\mathcal{W} \quad : \qquad \llbracket \boldsymbol{u}_i \rrbracket = \boldsymbol{u}_i + \Delta \boldsymbol{u}_i, \tag{5-2}$$

[1] Note, however, that if encryption is present then time-varying session keys such that identical outputs lead to different ciphertexts are a better alternative, provided the computation resources are available [118].

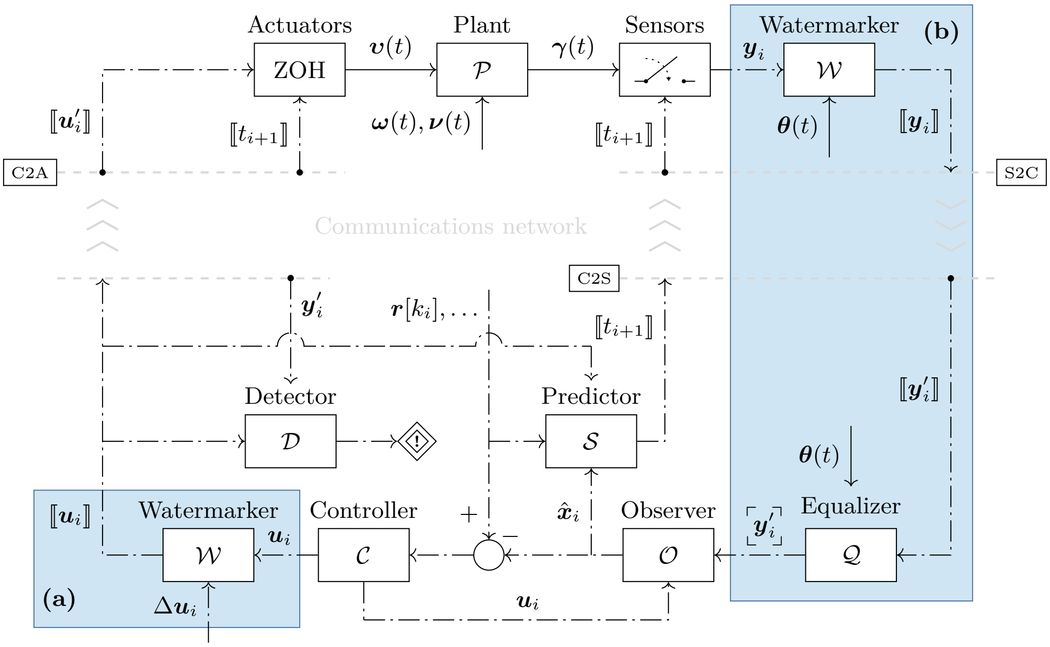


Figure 5-1: Overview of **(a)** additive watermarking and **(b)** multiplicative watermarking

where $\Delta \boldsymbol{u}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_u)$ is additive zero-mean Gaussian noise. Extensions for watermakers with data-driven [81] and non-Gaussian distributions [59] are possible. With a standard χ^2 detector, replay attacks can be detected, induced by a different realization of $\Delta \boldsymbol{u}_i$ when the outputs are replayed. The main disadvantage of the proposed scheme is that the added input noise $\Delta \boldsymbol{u}(k)$ leads to a performance deficit [134, 90]. As noted in [37], such a watermarker $\mathcal{W}$ may result in the waste of control cost especially when the attack is absent.

Proposed as an alternative to mitigate the performance loss caused by additive watermarking, (dynamic) multiplicative watermarking [41] is a scheme where outputs are filtered at the sensors by a watermarker $\mathcal{W}$, after which they are transmitted over the network and then equalized at the controller side by an equalizer $\mathcal{Q}$ (see Figure 5-1b). The method proposed is applicable for single-input and single-output (SISO) systems but can readily be applied to multiple-input and multiple-output (MIMO) systems by considering n_{y} pairs of watermarkers and equalizers. The watermarked measurement $[\![\boldsymbol{y}_i]\!]$ is obtained by passing $\boldsymbol{y}_i$ through the watermarking filter $\mathcal{W}$ given by

$$\mathcal{W} \quad : \qquad W(z\,; \boldsymbol{\theta}(t_i)) = \left(\begin{array}{c|c} \boldsymbol{A}_{\mathrm{w}} & \boldsymbol{B}_{\mathrm{w}} \\ \hline \boldsymbol{C}_{\mathrm{w}} & \boldsymbol{D}_{\mathrm{w}} \end{array} \right), \tag{5-3}$$

where the explicit dependence on $\boldsymbol{\theta}(t_i)$ for $\boldsymbol{A}_{\mathrm{w}}(\boldsymbol{\theta}(t_i))$, $\boldsymbol{B}_{\mathrm{w}}(\boldsymbol{\theta}(t_i))$, $\boldsymbol{C}_{\mathrm{w}}(\boldsymbol{\theta}(t_i))$ and $\boldsymbol{D}_{\mathrm{w}}(\boldsymbol{\theta}(t_i))$ has been omitted for brevity. The design of the matrices is given by

$$\boldsymbol{A}_{\mathrm{w}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ & \boldsymbol{0}^{\mathrm{T}} \end{bmatrix}, \qquad \boldsymbol{B}_{\mathrm{w}} = \begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix}, \qquad \boldsymbol{C}_{\mathrm{w}} = \boldsymbol{\theta}^{\mathrm{T}}(t_i), \qquad \boldsymbol{D}_{\mathrm{w}} = \theta_1(t_i), \tag{5-4}$$

with the identity matrix $\boldsymbol{I}$ and zero matrix $\boldsymbol{0}$ of appropriate sizes (see [41]). The equalizer $\mathcal{Q}$ is given by

$$\mathcal{Q} \quad : \qquad Q(z\,; \boldsymbol{\theta}(t_i)) = \left(\begin{array}{c|c} \boldsymbol{A}_{\mathrm{w}} - \boldsymbol{B}_{\mathrm{w}} \boldsymbol{D}_{\mathrm{w}}^{-1} \boldsymbol{C}_{\mathrm{w}} & -\boldsymbol{B}_{\mathrm{w}} \boldsymbol{D}_{\mathrm{w}}^{-1} \\ \hline \boldsymbol{D}_{\mathrm{w}}^{-1} \boldsymbol{C}_{\mathrm{w}} & \boldsymbol{D}_{\mathrm{w}}^{-1} \end{array} \right), \tag{5-5}$$

which are designed such that $Q(z\,;\boldsymbol{\theta}) \cdot W(z\,;\boldsymbol{\theta}) = 1, \forall z$ and $\left|Q(z\,;\boldsymbol{\theta}') \cdot W(z\,;\boldsymbol{\theta})\right| \gg 1$ when $\boldsymbol{\theta}' \neq \boldsymbol{\theta}$ for most z. Thus, the finite impulse response (FIR) watermarking filter $\mathcal{W}$ is minimum-phase, and $\mathcal{Q}$ is its stable inverse when the parameters $\boldsymbol{\theta}$ are matched. Here, the time-varying parameterization vector $\boldsymbol{\theta}(t_i)$ is the shared secret between the controller and the sensors. To ensure synchronization at switching times the internal states of $\mathcal{W}$ and $\mathcal{Q}$ are both reset to $\boldsymbol{0}$. As such, when $\boldsymbol{a}_{y,i} = \boldsymbol{0}$, for all i, we have that the equalized output $\ulcorner\!\!\llcorner \boldsymbol{y}'_i \lrcorner\!\!\urcorner = \boldsymbol{y}_i$.

5-2 Event-triggered χ^2 detector

To detect replay attacks the most common type of detector used is a χ^2 detector [131, 37]. This type of detector is a residual-based detector, which means it exploits model knowledge and distribution properties of the load disturbance and measurement noise. An alarm is raised through a binary hypothesis test, which is used almost exclusively in cyber-physical systems (CPSs) [140, 37], given by [108]

$$g_i \underset{\mathcal{H}^0}{\overset{\mathcal{H}^1}{\gtrless}} \eta_i, \qquad \begin{aligned} &\mathcal{H}^0 : \text{Nominal system operation.} && \text{(5-6a)}\\ &\mathcal{H}^1 : \text{System under attack.} && \text{(5-6b)}\end{aligned}$$

Here, g_i is a scalar detection signal computed by the detector $\mathcal{D}$, which is then compared to a (possibly constant) threshold value η_i. The main idea behind a detector is the use of an estimator to forecast the evolution of the system [92]. As we are dealing with a switched linear (SL) system (see (3-7)) we model the residual-based detector $\mathcal{D}$ as a time-varying Kalman filter given by [125]

$$\mathcal{D} \quad : \quad \begin{aligned} \boldsymbol{d}_{i\,|\,i-1} &= \boldsymbol{A}_{\kappa_i}\boldsymbol{d}_{i-1} + \boldsymbol{B}_{\kappa_i}\boldsymbol{u}_{i-1}, && \text{(5-7a)}\\ \boldsymbol{\Sigma}_{\mathrm{x},i\,|\,i-1} &= \boldsymbol{A}_{\kappa_i}\boldsymbol{\Sigma}_{\mathrm{x},i-1}\boldsymbol{A}_{\kappa_i}^{\mathrm{T}} + \boldsymbol{\Sigma}_{\mathrm{w},\kappa_i}, && \text{(5-7b)}\\ \boldsymbol{H}_i &= \boldsymbol{\Sigma}_{\mathrm{x},i\,|\,i-1}\boldsymbol{A}_{\kappa_i}^{\mathrm{T}}(\boldsymbol{C}\boldsymbol{\Sigma}_{\mathrm{x},i\,|\,i-1}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{\Sigma}_{\mathrm{v}})^{\text{-}1} && \text{(5-7c)}\\ \boldsymbol{d}_i &= (\boldsymbol{I} - \boldsymbol{H}_i\boldsymbol{C})\boldsymbol{d}_{i\,|\,i-1} + \boldsymbol{H}_i\boldsymbol{y}_i && \text{(5-7d)}\\ \boldsymbol{\Sigma}_{\mathrm{x},i} &= (\boldsymbol{I} - \boldsymbol{H}_i\boldsymbol{A}_{\kappa_i})\boldsymbol{\Sigma}_{\mathrm{x},i\,|\,i-1} && \text{(5-7e)}\end{aligned}$$

where (5-7a)-(5-7b) and (5-7c)-(5-7e) denote the prediction step and denote the update step, respectively. Note that by Assumption 1 and Assumption 3 observability holds for all IETs. Furthermore, as a detection mechanism we use a standard (static) χ^2 detector $\mathcal{D}$ (see Figure 3-2) as

$$\mathcal{D} \quad : \quad \begin{aligned} \boldsymbol{z}_i &= \boldsymbol{y}_i - \boldsymbol{C}\boldsymbol{d}_i, && \text{(5-8a)}\\ g_i &= \boldsymbol{z}_i^{\mathrm{T}}(\boldsymbol{C}\boldsymbol{\Sigma}_{\mathrm{x},i}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{\Sigma}_v)^{\text{-}1}\boldsymbol{z}_i, && \text{(5-8b)}\end{aligned}$$

where g_i is the detection signal which is fed through a binary hypothesis test as in (5-6). The above detection scheme is similar to the one proposed in [91] only here modified for an STC policy. Extensions to more sophisticated (dynamic) detectors (see e.g. [92, 81]) are possible but left to future work. The detector proposed above suffices for demonstrative purposes. Note that (5-7c) requires inverting a matrix online, which might be an expensive procedure. As such, we propose an alternative method for constructing a (sub-optimal) switched observer gain $\boldsymbol{H}_\kappa$ offline, which can be found in Appendix A-2.

For any fixed i, it is easy to verify that the detection signal g_i as given in (5-8b) follows a χ^2 distribution with n_{y} degrees of freedom, which stems from the fact that the detector residual $\boldsymbol{z}_i \sim (\boldsymbol{0}, \boldsymbol{C}\vec{\boldsymbol{\Sigma}}_{\mathrm{x},i}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{\Sigma}_{\mathrm{v}})$ [89, 131]. As such, calculating the static threshold η based on a false alarm rate $p_{\mathrm{fp}} \in (0, 1)$ is straightforward, and given by [59]

$$\eta = 2 \cdot P^{\text{-}1}\Big(\frac{n_y}{2}, 1 - p_{\mathrm{fp}}\Big), \tag{5-9}$$

where $P^{\text{-}1}$ denotes the inverse regularized lower incomplete gamma function. Note that whilst in general it is desirable to choose the false alarm rate p_{fp} small (enough), one can not make it arbitrarily small without loosing detection probability in our proposed method, or incurring a large control cost in the case of additive watermarking (see §5-1-1).

It must be noted that the reason for considering a static observer gain for $\mathcal{O}$ as discussed in §3-7 and a different, time-varying observer gain for our proposed detector $\mathcal{D}$, is that for the former we were only interested in guaranteeing stability of the STC implementation in the absence of noise, which can be achieved through Theorem 3.3. However, here we are instead interested in creating a close match in the distribution of the residual and the state vector of the plant, which can be achieved through the detector $\mathcal{D}$ as proposed in (5-7). Since the detector $\mathcal{D}$ is modular and does not affect stability in any way, it can be neglected in the stability analysis as performed in §3-7.

Finally, since we are dealing with a non-zero false alarm rate p_{fp} we make the following assumption [41, Assumption 2], which constitutes the worst-case scenario when considering replay attacks. In practice, this is rather common as a sufficiently low p_{fp} is desirable from an economic and labor perspective.

Assumption 8. *No false alarms are triggered during the interval* $[T_\text{a} - \Delta T, T_\text{a})$*.* ◊

In the absence of watermarking, under Assumption 8, a replay attack becomes ϵ-stealthy with respect to a χ^2 detector [90]. The loop length Δi must, however, be appropriately chosen to not trigger an alarm as the first replayed measurement is received (see Appendix A-4). As the plant is running in an open loop during a replay attack, and the control inputs $\boldsymbol{u}_i$ are not based on the genuine outputs of the plant, the attack can become disruptive as well given a sufficiently large t.

5-3 Augmented triggering design

In order to detect a replay attack, any output $\boldsymbol{y}_i$ replayed at a later time must not invoke the exact same control action and as such, produce a (statistically) identical proceeding output. Thus, a time-varying component must be added to check the consistency of the received output. In additive watermarking, this time-varying component comes from the realization $\Delta \boldsymbol{u}_i$ which differs from the one at $\Delta \boldsymbol{u}_{i-\Delta i}$. In multiplicative watermarking, this is induced by the time-varying parameter vector $\boldsymbol{\theta}$ and the internal states of $\mathcal{W}$ and $\mathcal{Q}$ being different from that at time $T_\text{a} - \Delta T$. In an STC policy there is one extra degree of freedom, namely the next sampling period t_{i+1} [4]. Our proposed countermeasure to replay attacks is based on exploiting this additional degree of freedom in order to detect replay attacks.

Theorem 5.1 (Early triggering [50, Collary 3])**.** *Consider a* STC *policy* $\mathcal{S}$ *and suppose the closed-loop system as in Figure 3-2 is* GES*. Then, any alternate* STC *policy* $\mathcal{S}'$ *for which* $\tau'_{i+1} \leqslant \tau_{i+1}$*, for all pairs* $(\boldsymbol{x}_i, \boldsymbol{u}_i)$*, guarantees closed-loop stability and ensures equal or better control performance.*

The intuition behind Theorem 5.1, at least for quadratic triggering conditions, is that quite often a triggering condition is a surrogate for ensuring the decrease of (a bound on) an underlying Lyapunov function. Then, triggering no later will thus always ensure this decrease is maintained and thus preserve stability. Whilst [49] propose early triggering to achieve a near-maximal average IET, in this work, we show that early triggering can also be exploited for the detection of replay attacks. In a similar fashion to [49], we will regard the next inter-event index computed by Γ in (3-13) as a *deadline*, which we will denote by $\bar{\kappa}_{i+1}$. We propose a non-deterministic STC watermarking scheme as

$$\mathcal{S} \quad : \qquad [\![t_{i+1}]\!] = t_i + h \cdot [\![\kappa_{i+1}]\!], \qquad [\![\kappa_{i+1}]\!] \sim p(\kappa \,|\, \bar{\kappa}_{i+1}\,; \hat{\boldsymbol{x}}_i, \boldsymbol{u}_i), \tag{5-10}$$

where $[\![\kappa_{i+1}]\!]$ denotes the $(i+1)$-th watermarked inter-event index, and p denotes a probability mass function (PMF) with support $\{1, \dots, \bar{\kappa}_{i+1}\}$ given by

$$p(\kappa \,|\, \bar{\kappa}_{i+1}) = \begin{cases} p_\kappa, & 1 \leqslant \kappa \leqslant \bar{\kappa}_{i+1} \\ 0, & \text{otherwise.} \end{cases} \tag{5-11}$$

Here, the entries $p_\kappa \in [0, 1]$ are to be designed (see §5-3-1) in order to aid reliant and swift detection of replay attacks. Note that p_κ and thus the distribution of $[\![\kappa_{i+1}]\!]$ in general depends on the current state (estimate) and input, which have been omitted in (5-11).

On a side note, one way to guarantee Assumption 5 holds is to use the quadratic triggering condition $\|\boldsymbol{x}_i - \boldsymbol{\chi}(t)\| > \epsilon$ as in [26], which can be shown to always trigger no later than the modified Lebesgue sampling they propose (see Appendix A-3-1). As such, this preserves stability as by Theorem 5.1. By implementing this periodic event-triggered control (PETC) policy at the sensor side, where now the new event time t_i is given by

$$\mathcal{E} \quad : \qquad t_i = h \cdot \min_k \{k \in \mathbb{N}, k > k_{i-1} \,|\, \boldsymbol{p}^{\mathrm{T}}[k] \boldsymbol{Q} \boldsymbol{p}[k] > \epsilon \vee k = k_{i-1} + [\![\bar{\kappa}_i]\!]\}, \tag{5-12}$$

we can guarantee mean square stable (MSS). In this way, the true deadlines are determined by the PETC policy, whilst $[\![\kappa_{i+1}]\!]$ sets an upper bound to the next inter-event index, creating a hierarchical structure. As our STC policy constitutes early triggering, it is reasonable to assume that the majority of the time the next sampling time t_{i+1} as demanded by the STC policy is chosen, meaning the alterations would not be too intrusive to the framework proposed here. Still, at this time this solution is not implemented, and whether Assumption 5 holds is checked *a posteriori*.

To the best of the author's knowledge [9] is the first to consider an event-triggered control (ETC) watermarking scheme for replay attacks. However, our framework proposed deviates significantly from the one they propose. For one, they consider replay attacks in the absence of load disturbances and measurement noise (i.e. deterministic systems), and they state that steps need to be taken to characterize the detectability trade-offs when random disturbances naturally enter the system model. Furthermore, their triggering condition is restricted to a *send-on-delta* condition $\|\boldsymbol{r} - \boldsymbol{x}[k]\|$ with constant reference $\boldsymbol{r}[k] = \boldsymbol{r}$, whilst we consider time-varying periodic references.

Finally, note that for the proposed early triggering mechanism to succeed, the STC policy needs to be *non-trivial* (see Definition 3.2). This might not always be straightforward to guarantee. For example, the STC policy as proposed in [50], which takes into account worst-case bounded perturbations, tends to yield periodic control as the state approaches the origin. Obviously, our proposed scheme would thus not be applicable to such an STC policy.

5-3-1 Optimal watermarking scheme

Given the early triggering approach, we would like to design p_κ such that the missed detection rate p_{fn} is as low as possible. In the absence of an attack, the false alarm rate follows

$$\mathbb{P}[g_i > \eta \,|\, \mathcal{H}_0] = p_{\text{fp}}, \qquad \forall i. \tag{5-13}$$

If we assume the system has been running for a prolonged period of time (i.e. $t = -\infty$), under a replay attack the replayed measurements are drawn from the same early triggering distribution. As such, the missed detection probability at event i can be written as

$$\mathbb{P}[g_i < \eta \,|\, \mathcal{H}_1\,; \boldsymbol{x}_i, \boldsymbol{u}_i] = p_{\text{fn}} = \sum_{\kappa,\kappa'}^{\bar{\kappa}_i} p_\kappa \cdot p_{\kappa'} \cdot \begin{cases} 1 - p_{\text{fp}}, & \kappa = \kappa', \\ I(\boldsymbol{\mu}_\kappa, \boldsymbol{\mu}_{\kappa'}, \boldsymbol{\Sigma}_\kappa, \boldsymbol{\Sigma}_{\kappa'}\,; \eta), & \text{otherwise.} \end{cases} \tag{5-14}$$

Note that we are only considering the probability of detection between two events instead of detection over multiple events. The rationale behind this is that we are interested in timely detection before the attack can inflict any physical damage to the control system. Here, the integrand $I : \mathbb{R}^{n_\text{y}} \times \mathbb{R}^{n_\text{y}} \times \mathbb{R}^{n_\text{y} \times \text{y}} \times \mathbb{R}^{n_\text{y} \times \text{y}} \to [0, 1]$ is given by

$$I(\boldsymbol{\mu}_\kappa, \boldsymbol{\mu}_{\kappa'}, \boldsymbol{\Sigma}_\kappa, \boldsymbol{\Sigma}_{\kappa'}\,; \eta) = \underset{E(\boldsymbol{\mu}_\kappa, \boldsymbol{\Sigma}_{\text{x},\kappa}\,;\eta)}{\int \cdots \int} \mathcal{N}(\boldsymbol{\mu}_{\kappa'}, \boldsymbol{\Sigma}_{\text{x},\kappa'})\, \mathrm{d}V, \tag{5-15}$$

where the hyper-ellipsoid $E(\boldsymbol{\mu}, \boldsymbol{\Sigma}\,; \eta) = \{\boldsymbol{x}_i \in \mathbb{R}^{n_\text{x}} \,|\, (\boldsymbol{x}_i - \boldsymbol{\mu})^{\text{T}} \boldsymbol{\Sigma} (\boldsymbol{x}_i - \boldsymbol{\mu}) \leqslant \eta\}$. Two observations can be made from (5-14). First, the missed detection rate p_{fn} is, as expected, a function of the false alarm rate p_{fp}. Similarly, note that η is also a function of the false alarm rate as outlined in (5-9). If $p_{\bar{\kappa}_i} = 1$ and $p_\kappa = 0$ for all $\kappa \neq \bar{\kappa}_i$, which is the case when no watermarking is present, we have $p_{\text{fn}} = 1 - p_{\text{fp}}$ which is undesirable as this implies a high missed detection rate. Second, expression (5-15) represents the volume integral over a hyper-ellipsoidal region, which can be difficult to obtain in practice.

Our goal is to construct the entries p_κ of (5-11) to aid detection of replay attacks. Given that $\bar{\kappa}_i \leqslant \bar{\kappa} < \infty$, there are finitely many probabilities to determine. For convenience we introduce $\boldsymbol{p} = \text{col}(p_1, \ldots, p_{\bar{\kappa}_i}) \in \mathbb{R}^{n_\text{p}}$, with $n_\text{p} = \bar{\kappa}_i$, as our decision variable. Ideally, we would like to choose the weights to minimize the missed detection rate p_{fn} during an attack, which we can write as

$$\min_{\boldsymbol{p}} p_{\text{fn}}(\boldsymbol{p}) \qquad \text{s.t.} \tag{5-16a}$$

$$\|\boldsymbol{p}\|_0 = 1, \tag{5-16b}$$

$$\mathbf{0} \leqslant \boldsymbol{p} \leqslant \mathbf{1}, \tag{5-16c}$$

Due to the impossibility of computing the detection probability in closed-form [91], only a relaxed version of the original optimization problem (5-16) is solved. Recognizing that there is an inherent trade-off between obtaining the largest IETs and reliant attack detection, we propose an alternative problem formulation given by

$$\min_{\boldsymbol{p}} \boldsymbol{p}^{\mathrm{T}}\boldsymbol{W}_{\mathrm{p}}\boldsymbol{p} + \boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{p} \qquad \text{s.t.} \tag{5-17a}$$

$$\|\boldsymbol{p}\|_0 = 1, \tag{5-17b}$$

$$\boldsymbol{0} \leqslant \boldsymbol{p} \leqslant \boldsymbol{1}. \tag{5-17c}$$

where $\boldsymbol{W}_{\mathrm{p}} \succ 0$ and $\boldsymbol{\gamma} \in \mathbb{N}^{n_{\mathrm{p}}}$ are a matrix and vector, respectively, to be designed. Note that the parametric dependence on $\hat{\boldsymbol{x}}_i$, $\boldsymbol{u}_i$ has been omitted for brevity. Here, the entries of $\boldsymbol{W}_p$ should reflect how different values of p_κ effect the missed detection rate whilst $\boldsymbol{\gamma}$ is a penalty factor for inducing early triggering. In general, constructing $\boldsymbol{W}_{\mathrm{p}}$ is hard and compromises have to be made. Since (5-15) is cumbersome to evaluate numerically, as an alternative we propose to incorporate the Kullback-Leibler (KL) divergence of the residuals under different inter-event indices κ into the matrix $\boldsymbol{W}_{\mathrm{p}}$. The rationale behind this is that residuals coming from distributions with large divergence have a high chance of triggering an alarm (and thus lower the missed detection rate). Given the current state estimate and input $\hat{\boldsymbol{x}}_i$ and $\boldsymbol{u}_i$, respectively, the residuals of the detector as in (5-8a) follow a normal distribution. We can define this distribution $\mathcal{N}_\kappa$ by its mean $\boldsymbol{\mu}_\kappa$ and covariance $\boldsymbol{\Sigma}_\kappa$, which are given by

$$\boldsymbol{\mu}_\kappa = \boldsymbol{\Phi}(\kappa\,; \hat{\boldsymbol{x}}_i, \boldsymbol{u}_i), \tag{5-18a}$$

$$\boldsymbol{\Sigma}_\kappa = \boldsymbol{C}\boldsymbol{\Sigma}_{\mathrm{w},\kappa}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{\Sigma}_{\mathrm{v}}, \tag{5-18b}$$

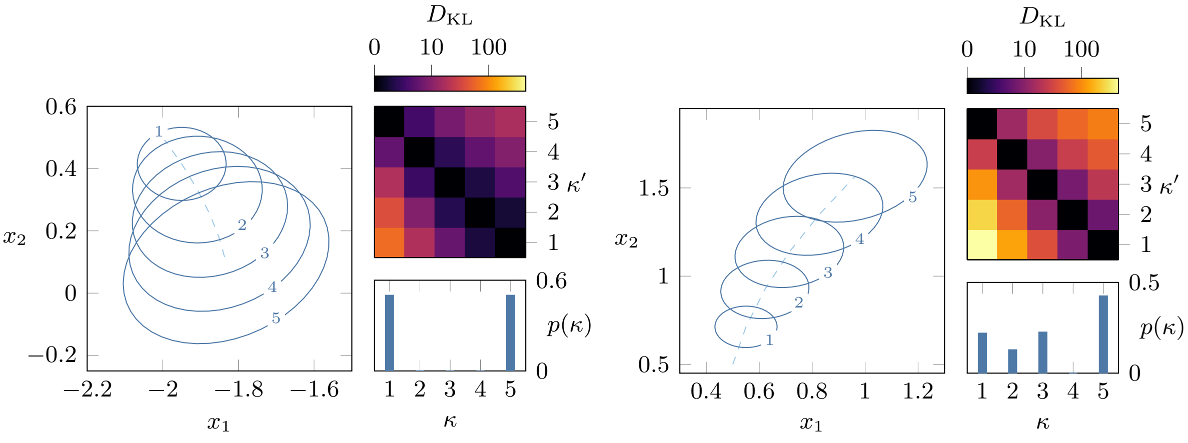
where $\boldsymbol{\Phi}$ and $\boldsymbol{\Sigma}_{\mathrm{w},\kappa}$, $\boldsymbol{\Sigma}_{\mathrm{v}}$ are defined as in (3-14) and (3-9), (3-3), respectively. For two normal distributions $\mathcal{N}_\kappa$ and $\mathcal{N}_{\kappa'}$, both of dimension n_{y}, the KL divergence is known in closed form and given by [11]

$$D_{\mathrm{KL}}(\mathcal{N}_\kappa\|\mathcal{N}_{\kappa'}) = \frac{1}{2}\cdot\left((\boldsymbol{\mu}_{\kappa'}-\boldsymbol{\mu}_\kappa)^{\mathrm{T}}\boldsymbol{\Sigma}_{\kappa'}^{-1}(\boldsymbol{\mu}_{\kappa'}-\boldsymbol{\mu}_\kappa) + \mathrm{tr}(\boldsymbol{\Sigma}_{\kappa'}^{-1}\boldsymbol{\Sigma}_{\kappa'}) - \log\frac{|\boldsymbol{\Sigma}_\kappa|}{|\boldsymbol{\Sigma}_{\kappa'}|} - n_{\mathrm{y}}\right). \tag{5-19}$$

Note that the KL divergence is in general not symmetric, and therefore we opt for the use of the (symmetric) Jeffreys divergence given by $D_{\mathrm{J}}(\mathcal{N}_\kappa\|\mathcal{N}_{\kappa'}) = D_{\mathrm{KL}}(\mathcal{N}_\kappa\|\mathcal{N}_{\kappa'}) + D_{\mathrm{KL}}(\mathcal{N}_{\kappa'}\|\mathcal{N}_\kappa)$. The rationale behind using KL divergence is inspired by [91] and motivated by the fact that minimizing the missed detention probability is related to maximizing the KL divergence between the distribution of the residuals for different values of κ. Combining these we propose the following heuristic as

$$\hat{\boldsymbol{W}}_{\mathrm{p}} = \begin{bmatrix} w(1,1) & w(1,2) & \cdots & w(1,\bar{\kappa}_i) \\ w(2,1) & w(2,2) & \cdots & w(2,\bar{\kappa}_i) \\ \vdots & \vdots & \ddots & \vdots \\ w(\bar{\kappa}_i,1) & w(\bar{\kappa}_i,2) & \cdots & w(\bar{\kappa}_i,\bar{\kappa}_i) \end{bmatrix}, \qquad \boldsymbol{\gamma} = \gamma\cdot\begin{bmatrix} n_{\mathrm{p}} \\ n_{\mathrm{p}}-1 \\ \vdots \\ 1 \end{bmatrix}, \tag{5-20}$$

where $w(\kappa,\kappa') = e^{-\sqrt{p_{\mathrm{fp}}}\cdot D_{\mathrm{J}}(\mathcal{N}_\kappa\|\mathcal{N}_{\kappa'})}$. Here, the negative exponential is introduced to convert maximizing the divergence into minimizing the missed detection rate, as well as a heuristic scaling measure. Similarly, the heuristic $\sqrt{p_{\mathrm{fp}}}$ skews the weighting such that, when the false positive rate is low, small differences in divergence are penalized more. Finally, $\gamma \in \mathbb{R}_{\geqslant 0}$ proportionally penalizes early triggering (as late triggering is desirable from a resource utilization



(a) Optimal distribution for considerable overlap favoring the largest difference

(b) Optimal distribution with penalization weight $\gamma = 0.1$ favoring $\kappa = 5$

Figure 5-2: Illustrative examples of the optimal triggering policy under the proposed heuristic with $\bar{\kappa}_i = 5$. Note the asymmetry of the KL divergence.

perspective). Note that as $\gamma \to \infty$, the optimal solution becomes trigger as late as possible which is given by the deadline $\bar{\kappa}_i$.

Considerable care must be taken as the matrix $\hat{\boldsymbol{W}}_\mathrm{p}$ might not be positive-definite (PD), rendering (5-17) no longer a quadratic program (QP). As such, in accordance with common practice in covariance regularization [18], define $\boldsymbol{W}_\mathrm{p} = \hat{\boldsymbol{W}}_\mathrm{p} - (\min\{\min\{\lambda(\hat{\boldsymbol{W}}_\mathrm{p})\}, 0\} + \epsilon_0) \cdot \boldsymbol{I}$ for some small positive $\epsilon_0 \approx 0$ to avoid zero eigenvalues. This enforces the matrix $\boldsymbol{W}_\mathrm{p}$ to be diagonally dominant which implies $\boldsymbol{W}_\mathrm{p} \succ 0$.

The heuristic as proposed in (5-20) works well in practice and as an added benefit, is simple due to the sole tunable parameter γ. Two illustrative examples are provided in (see Figure 5-2). Here, the blue ellipsoidal regions in Figure 5-3a denote the sets $\{\boldsymbol{x}_i \in \mathbb{R}^{n_\mathrm{x}} \mid (\boldsymbol{x}_i - \boldsymbol{\mu}_\kappa)^\mathrm{T} \boldsymbol{\Sigma}_\kappa (\boldsymbol{x}_i - \boldsymbol{\mu}_\kappa) \leqslant \eta\}$, where η denotes the detector threshold based on the false alarm rate as in (5-9). The dashed line denotes the mean $\boldsymbol{\mu}_\kappa$ as in (5-18b).

Since the optimization problem (5-17) needs to be solved online at each event instance for the pair $(\boldsymbol{x}_i, \boldsymbol{u}_i)$, this approach can become computationally expensive. A benefit is that the computation is performed at the controller side where usually more computational resources are available [4]. Still, it might be preferable to approximate the optimal solution *a priori*, and as such, we will discuss an offline procedure next.

5-3-2 Offline procedure

As an alternative to solving (5-17) online for each new pair $(\boldsymbol{x}_i, \boldsymbol{u}_i)$, an offline procedure can be employed instead, based on several assumptions which can be checked *a posteriori*. Such a procedure might make more economic sense, and for practical purposes, this might suffice. Note that we assume that the system is initialized at $t = -\infty$, meaning there is sufficient time to check the assumption holds, and that, unlike stability, there is no safety risk if we can guarantee no attacks are imminent.

Inspired by [53], holding the input longer creates a very distinct distribution for the residuals (with accompanying high divergence), if the magnitude of the input is large compared to the magnitude of the disturbance covariance matrix. That is, given that the input 'overpowers' the additive load disturbances (which usually occurs when the state is far enough away from the origin), the following simplification can be made.

Proposition 5.2. *Consider the optimization problem (5-17) with $\gamma = 0$ and suppose that $\|\boldsymbol{u}_i\| \geqslant \bar{u}$ such that $\|\boldsymbol{CBu}_i\| \gg \|\boldsymbol{C}(\boldsymbol{Ax}_i + \boldsymbol{E\Sigma}_{\mathrm{w}}) + \boldsymbol{\Sigma}_{\mathrm{v}}\|$ for all pairs $(\boldsymbol{x}_i, \boldsymbol{u}_i)$. Then, the optimal watermarking distribution will (approximately) be a discrete uniform one, i.e. $[\![\kappa_i]\!] \sim \mathcal{U}(1, \bar{\kappa}_i)$.*

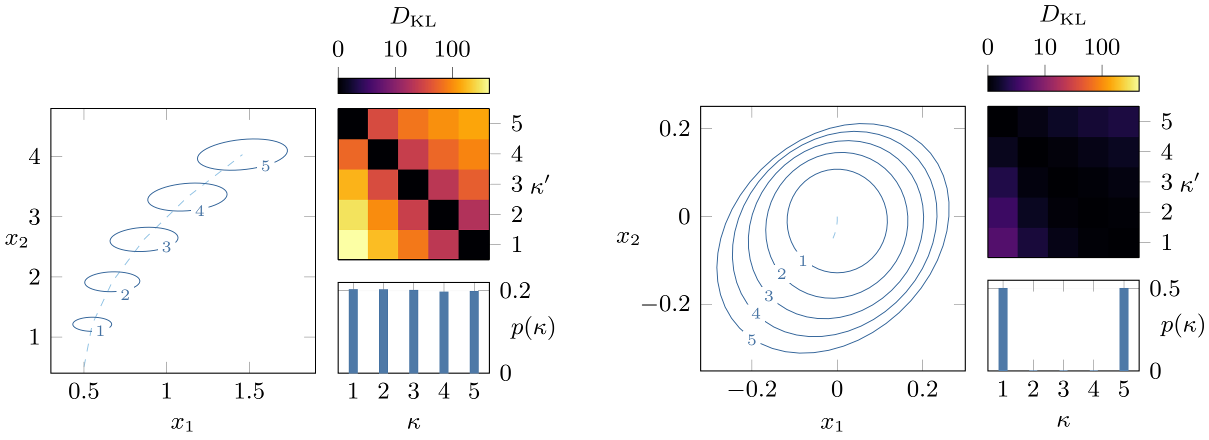
Proof: Recall that $\boldsymbol{z}_i = \boldsymbol{y}_i - \boldsymbol{Cd}_i$ given in (5-8a), where $\boldsymbol{y}_i = \boldsymbol{C}(\boldsymbol{Ax}_{i-1} + \boldsymbol{Bu}_{i-1} + \boldsymbol{E\Sigma}_{\mathrm{w}}) + \boldsymbol{\Sigma}_{\mathrm{v}}$. If $\|\boldsymbol{CBu}_i\| \gg \|\boldsymbol{C}(\boldsymbol{Ax}_i + \boldsymbol{E\Sigma}_{\mathrm{w}}) + \boldsymbol{\Sigma}_{\mathrm{v}}\|$ then, for sufficiently large $\|\boldsymbol{CBu}_i\|$, the difference $\boldsymbol{\mu}_\kappa - \boldsymbol{\mu}_{\kappa'}$ between the means of the distribution as in (5-18a) will start do dominate the expression (5-19) whenever $\kappa \neq \kappa'$. Note that this term is positive as $\boldsymbol{\Sigma}_{\kappa'}^{-1} \succ 0$, and furthermore that it is the only term containing $\boldsymbol{\mu}_\kappa$ and $\boldsymbol{\mu}_{\kappa'}$, which are dependent on $\boldsymbol{u}_i$ by (3-14). Thus, $D_{\mathrm{KL}}(\mathcal{N}_\kappa \| \mathcal{N}_{\kappa'}) \gg 0$ whenever $\kappa \neq \kappa'$ (in fact, $D_{\mathrm{KL}}(\mathcal{N}_\kappa \| \mathcal{N}_{\kappa'}) \to \infty$ as $\boldsymbol{u}_i \to \infty$ for fixed $\boldsymbol{\Sigma}_{\mathrm{w}}$ and $\boldsymbol{\Sigma}_{\mathrm{v}}$) and $D_{\mathrm{KL}}(\mathcal{N}_\kappa \| \mathcal{N}_\kappa) = 0$. Therefore, the weighting matrix as in (5-20) will approximately be $\boldsymbol{W}_{\mathrm{p}} \approx \boldsymbol{I}$. We can find an analytic solution to (5-17) by considering the Lagrangian $\mathcal{L}(\boldsymbol{p}, \boldsymbol{\mu}, \lambda) = 1/2 \cdot \boldsymbol{p}^{\mathrm{T}} \boldsymbol{p} + \boldsymbol{\mu}^{\mathrm{T}} \mathrm{col}(-\boldsymbol{p}, \boldsymbol{1} - \boldsymbol{p}) + \lambda \cdot (\boldsymbol{1}^{\mathrm{T}} \boldsymbol{x} - 1)$. First, consider the relaxed optimization problem where we neglect the inequality constraint. The Karush–Kuhn–Tucker (KKT) conditions state that for stationary $\boldsymbol{\nabla}_{\boldsymbol{p}} \mathcal{L}(\boldsymbol{p}, \lambda) = \boldsymbol{p} - \lambda \cdot \boldsymbol{1} = \boldsymbol{0}$, which implies $\boldsymbol{p} = \lambda \cdot \boldsymbol{1}$. Next, for primal feasibility of the equality constraint, we have $\boldsymbol{\nabla}_\lambda \mathcal{L}(\boldsymbol{p}, \lambda) = \boldsymbol{1}^{\mathrm{T}} \boldsymbol{p} - 1 = 0$, where substituting in the previous result gives $\boldsymbol{\nabla}_\lambda \mathcal{L}(\boldsymbol{p}, \lambda) = \boldsymbol{1}^{\mathrm{T}} \boldsymbol{p} - 1 = \lambda \cdot \boldsymbol{1}^{\mathrm{T}} \boldsymbol{1} - 1 = \lambda \cdot n_{\mathrm{p}} - 1 = 0$. From this it follows that $\lambda = 1/n_{\mathrm{p}}$ and thus $\boldsymbol{p} = 1/n_{\mathrm{p}} \cdot \boldsymbol{1}$. Note that this holds for all $n_{\mathrm{p}} \geqslant 1$ and thus for all values of $\bar{\kappa}_i$. Finally, we re-introduce the inequality constraint, and since $\boldsymbol{0} \leqslant 1/n_{\mathrm{p}} \cdot \boldsymbol{1} \leqslant \boldsymbol{1}$ holds we conclude that $\boldsymbol{p}^\star = 1/n_{\mathrm{p}} \cdot \boldsymbol{1}$. ■

A visualization of optimal trigger under sufficiently large inputs can be seen in Figure 5-3a. Solving the optimization problem (5-17), we find that the optimal distribution is indeed a discrete uniform one as stated in Proposition 5.2. Note that the results in Proposition 5.2 are independent of p_{fp} (and therefore, η) provided $\|\boldsymbol{CBu}_i\|$ is sufficiently large. Figure 5-3b denotes the situation whenever the input norm in not sufficiently large compared to the additive load disturbance, the consequences of which will be discussed in §5-6.

Finally, it must be noted that the early sampling strategy as discussed above constitutes a 'greedy' strategy, where we aim to minimize the chance of a missed detection between consecutive sampling instances. More sophisticated schemes where we take into account multiple consecutive sampling instances, in a similar vein to [49], could potentially lead to better results, but these are left to future work.

5-4 Control system design guidelines

To summarize the results from the previous section, we here provide the following design guidelines to guarantee both stability and security. These design guidelines are demonstrated in the numerical results in §5-5.



(a) Optimal distribution when separation between the means in large

(b) Ellipsoidal bounds becoming concentric near the origin

Figure 5-3: Illustrative examples of the optimal triggering policy under the proposed heuristic with $\bar{\kappa}_i = 5$ in two specific scenarios, the latter being problematic for attack detection.

1. Given a continuous-time plant $\mathcal{P}$ check that Assumption 1 holds, and design a continuous-time controller $\underset{\sim}{\mathcal{C}}$, either static full-state feedback (see §6-5) or dynamic output feedback (see §6-5), which guarantees global exponential stability (GES) in continuous-time. Alternatively, go to step **2**. first and design a digital controller $\mathcal{C}$ afterwards.

2. Select a sampling period $h \in \mathbb{R}_{>0}$ which satisfies Assumption 3.1 based on either *ad hoc* rules [86] or a formal method ensuring good performance in discrete-time[2].

3. Select a quadratic triggering matrix $\boldsymbol{Q}(\sigma)$ and (preferably large) upper bound $\bar{\kappa} \geqslant 1$ such that Assumption 4 is satisfied for $\mathcal{S}$, and a value of $\sigma \in \mathbb{R}_{\geqslant 0}$ such that the LMIs in Theorem 3.2 are feasible. The parameter $\epsilon \in \mathbb{R}_{\geqslant 0}$ can freely be chosen according to [21, Theorem 2].

4. If $\boldsymbol{C} \neq \boldsymbol{I}$, design an observer $\mathcal{O}$ according to Theorem 3.3 which implies $\lim_{i \to \infty} \tilde{\boldsymbol{x}}_i = \boldsymbol{0}$. According to Assumption 6 or [5, Lemma 5] (static controller) the cascade is ISS to observation errors $\tilde{\boldsymbol{x}}_i$, such that GES of the closed-loop system is preserved.

5. Design an early triggering mechanism for $\mathcal{S}$ either online using (5-17) for a given $\gamma \in \mathbb{R}_{\geqslant 0}$ or *a priori* under Proposition 5.2. Such an approach is inherently stable [50, Collary 3].

6. For a specified false alarm rate $p_{\mathrm{fp}} \in (0, 1)$ select η according to (5-9) for the event-triggered χ^2 detector $\mathcal{D}$.

[2] Such a procedure is called *exact emulation* [65], and it can be shown that if the continuous-time controller is designed such that it yields input-to-state stability (ISS) to sampling errors, then the same controller will achieve a semi-global practical ISS property when implemented in a sampled-data control loop [65].

5-5 Illustrative examples

In this section, we support our design method with some numerical results. Consider the unstable continuous-time plant $\underset{\sim}{\mathcal{P}}$ given by

$$\underset{\sim}{\boldsymbol{A}} = \begin{bmatrix} -0.02 & 1.01 \\ 0.35 & -0.12 \end{bmatrix}, \qquad \underset{\sim}{\boldsymbol{B}} = \begin{bmatrix} -0.05 \\ 10.06 \end{bmatrix}, \tag{5-21}$$

and full-state feedback such that $\boldsymbol{C} = \boldsymbol{I}$, such that Assumption 1 holds. Furthermore, consider a sampling period $h = 0.1$ such that the discretized dynamics $\mathcal{P}$ given by

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0.1 \\ 0.035 & 0.99 \end{bmatrix}, \qquad \boldsymbol{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{5-22}$$

are identical to the ones considered in [41], and Assumption 3 holds. From that same paper, consider the digital controller $\mathcal{C}$ given by

$$\boldsymbol{A}_{\mathrm{c}} = \boldsymbol{I}, \quad \boldsymbol{B}_{\mathrm{c}} = 0.1 \cdot \boldsymbol{I}, \quad \boldsymbol{C}_{\mathrm{c}} = \begin{bmatrix} 0.01 & 0.022 \end{bmatrix}, \quad \boldsymbol{D}_{\mathrm{c}} = \begin{bmatrix} 0.0875 & 0.198 \end{bmatrix}. \tag{5-23}$$

Consider the objective of tracking an arbitrary sinusoidal reference signal, here given by $\boldsymbol{r}(t) = \mathrm{col}(A_{\mathrm{r}} \cdot \sin(2\pi \cdot t/T_{\mathrm{r}}), A_{\mathrm{r}} \cdot \cos(2\pi \cdot t/T_{\mathrm{r}}))$ with amplitude $A_{\mathrm{r}} = 2$ and period $T_{\mathrm{r}} = 6$ such that $\boldsymbol{r}(t + T_{\mathrm{r}}) = \boldsymbol{r}(t)$. The system is considered to be initialized at $t = -\infty$ and therefore, the transient effect of the initial conditions ($\boldsymbol{x}_0 = \hat{\boldsymbol{x}}_0 = \boldsymbol{d}_0 = \boldsymbol{0}$, $\boldsymbol{\Sigma}_{\mathrm{x},0} = 10^{-3} \cdot \boldsymbol{I}$) can be neglected. This is achieved in NCSim by starting the simulation at $t = -10^4$. The matrices $\underset{\sim}{\boldsymbol{E}} = \boldsymbol{I}$ and $\underset{\sim}{\boldsymbol{\Sigma}}_{\mathrm{v}} = 10^{-2} \cdot \boldsymbol{I}$, which imply $\boldsymbol{\Sigma}_{\mathrm{w}} \approx \boldsymbol{\Sigma}_{\mathrm{v}} \approx 10^{-3} \cdot \boldsymbol{I}$ using (3-3). The seed value was set to `45538370` in NCSim.

Next, the STC policy is designed with $\boldsymbol{Q}$ as in (3-23), $\sigma = 0.32$ and $\bar{\kappa} = 10$. From Theorem 3.2 we find the system is GES with decay rate $\rho = 5.5 \cdot 10^{-1}$, which was found by means of a bisection method on the interval $[10^{-3}, 1]$. The margin parameter is initially set to zero, i.e. $\epsilon = 0$. Since $\boldsymbol{C} = \boldsymbol{I}$, no observer $\mathcal{O}$ needs to be designed and $\hat{\boldsymbol{x}}_i = \boldsymbol{y}_i$ (note that $n_{\mathrm{y}} = n_{\mathrm{x}}$). Finally, the event-triggered χ^2 detector $\mathcal{D}$ is designed with a false alarm rate $p_{\mathrm{fp}} = 0.1\%$, and as $n_y = 2$ this implies $\eta = 13.82$ as from (5-9). The safe region was chosen as $\mathbb{X}_{\mathrm{s}} = \{\boldsymbol{x}_0 \in \mathbb{R}^{n_{\mathrm{x}}} \mid \|\boldsymbol{x}_0\|_\infty \leqslant 4\}$.

An adversary $\underset{\sim}{\mathcal{A}}$ appears at $t_0 = 0$ and starts recording measurements, with $\mathbb{I}_{\mathrm{a}}(t)$ as in Assumption 7, from which $T_{\mathrm{r}} \approx 6$ can be deduced. At $T_{\mathrm{a}} = 20$, the adversary initiates a replay attack with a delay of two cycles meaning $\Delta T \approx 12$, such that $2 \cdot T_{\mathrm{r}} \approx \sum_{\ell=0}^{\Delta i - 1} |t_{i-\ell} - t_{i-(\ell+1)}|$. Four scenarios are considered, namely an STC policy without the additional watermarking as proposed in Chapter 5 (from here on referred to as *baseline*), the proposed early triggering STC, and an STC policy with additive watermarking and one with multiplicative watermarking (from here on referred to as *benchmarks*). Note that Δi differs for each scenario discussed here.

The resulting state trajectory $\boldsymbol{\chi}(t)$, received outputs $\boldsymbol{y}_i$ and hypothesis $\mathcal{H}_i \in \{0, 1\}$ as in (5-6) can be seen in Figure 5-4. For $t \in [t_0, T_{\mathrm{a}}]$ no alarms are triggered and as such Assumption 8 is satisfied. The first replayed measurement value is received at $k = 202$, indicated by the

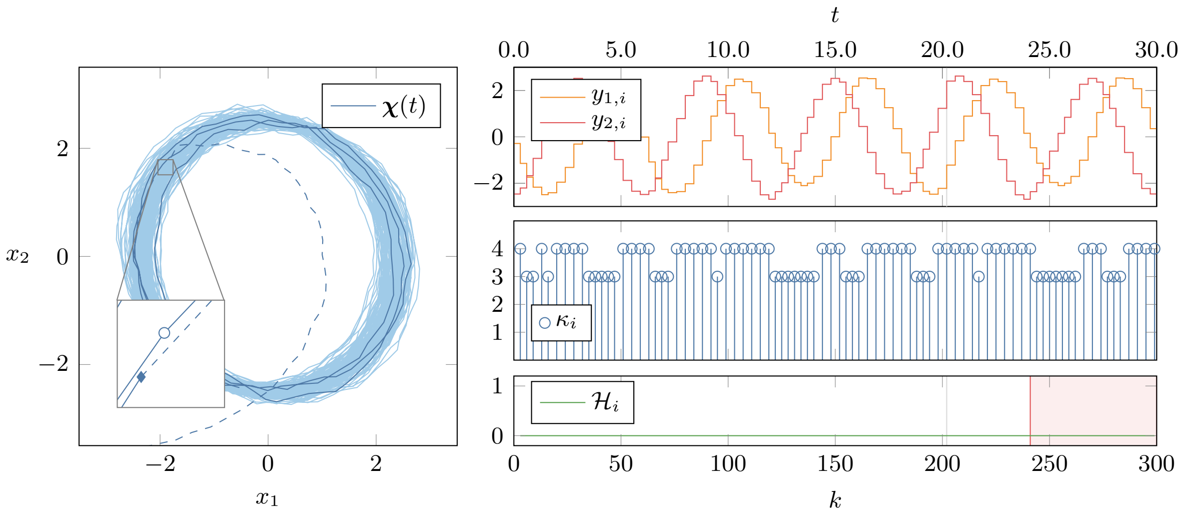


Figure 5-4: Effect of a replay attack when no watermarking is present (*baseline*)

gray line, with $\Delta i = 34$. By inspection of the top right plot, the replayed cycle becomes apparent. However, as seen from the bottom right plot, no alarm is raised even when the state trajectory leaves the safe region at $t = 24.1$, which is indicated by the red region. From the middle plot, we can see an exact repetition of the IETs once a replay attack is in progress.

For the early triggering mechanism, a uniform distribution is chosen for all pairs $(\boldsymbol{x}_i, \boldsymbol{u}_i)$, and at execution time it is checked if the assumption $\|\boldsymbol{u}_i\| \geqslant \bar{u}$ such that $\|\boldsymbol{CBu}_i\| \gg \|\boldsymbol{C}(\boldsymbol{Ax}_i + \boldsymbol{E\Sigma}_\mathrm{w}) + \boldsymbol{\Sigma}_\mathrm{v}\|$ holds. From the simulation, we find $\min_i \|\boldsymbol{CBu}_i\| = 2.57 \gg 2.7 \cdot 10^{-3} = \max_i \|\boldsymbol{C}(\boldsymbol{Ax}_i + \boldsymbol{E\Sigma}_\mathrm{w}) + \boldsymbol{\Sigma}_\mathrm{v}\|$ (disregarding the first dozen event instances) and as such, the optimal early triggering distribution is a uniform one according to Proposition 5.2. This is further confirmed by running the online optimization as in §5-3-1, which indeed verifies the findings.

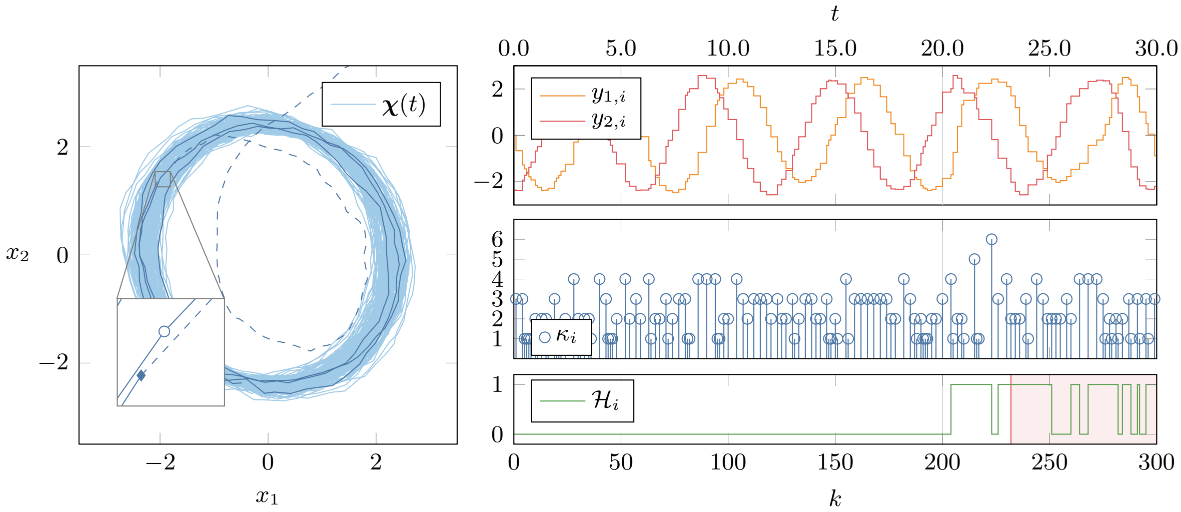


Figure 5-5: Effect of a replay attack with the proposed STC watermarking policy

The results can be seen in Figure 5-5, where this time the first replayed measurement value is received at $k = 200$ and $\Delta i = 52$. Different from the baseline, the attack is detected at $t = 20.2$ before the state trajectory leaves the safe region at $t = 23.2$. Interestingly, at the first replayed instance at $k = 200$ the attack is not yet detected (as the same IET is requested), and the first alarm is only raised at $k = 202$. Such behavior is inevitable for a non-deterministic approach. In the middle plot it can be seen that during the replay attack, the IETs are not identical to their delayed one (i.e. those at $\Delta i = 52$ sampling instances ago). This time-varying behavior is what makes the detection of replay attacks possible.

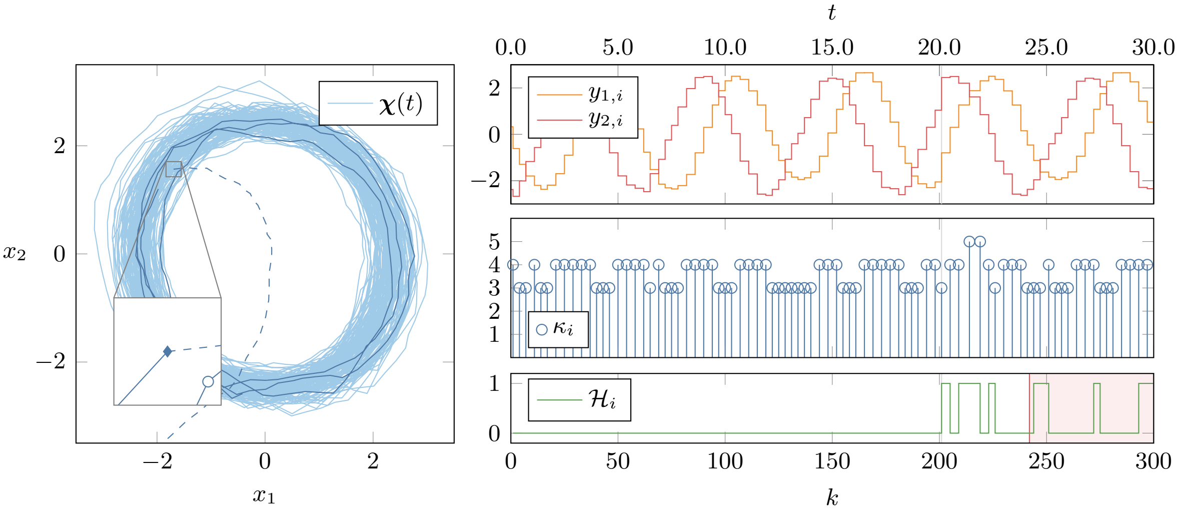


Figure 5-6: Effect of a replay attack with additive watermarking (*benchmark*)

Additive watermarking comes with one free parameter $\mathbf{\Sigma}_\mathrm{u}$ which must be chosen to balance a low missed detection rate and an acceptable loss in control performance. Whilst [91] propose an optimal watermarking strategy, they consider a static linear–quadratic–Gaussian (LQG) controller. As such, their results are not directly applicable to the dynamic controller considered here. Therefore, we have to resort to trial and error to set the covariance matrix. Starting at $\mathbf{\Sigma}_\mathrm{u} = \mathbf{0}$, the covariance was increased with steps of $10^{-3} \cdot \boldsymbol{I}$ up to the smallest value such that the replay attack was detected before the state trajectory left the safe region $\mathbb{X}_\mathrm{s}$. This way, a fair comparison can be made (at least for the particular scenario considered here).

For the first benchmark, we consider additive watermarking with $\mathbf{\Sigma}_\mathrm{u} = 3.9 \cdot 10^{-2}$ (note that $n_\mathrm{u} = 1$), the results of which can be seen in Figure 5-6. The first replayed measurement is received at $k = 201$, with $\Delta i = 34$, after which an alarm is immediately raised. Similar to the proposed early triggering method, additive watermarking is able to detect the attack. As depicted in the middle plot, the IETs corresponding to the replayed values are largely identical to the ones from the actual outputs, with notable exceptions at $k = 226, 251$.

Finally, multiplicative watermarking is considered, with a filter order $N_\mathrm{w} = 4$ and switch instances at $t = T_\mathrm{w} \cdot \mathbb{Z}$ with $T_\mathrm{w} = 1.5$. Similar to [41], the parameter vector $\boldsymbol{\theta}(t) = \mathrm{col}(1, 0, 0, 0) + \Delta\boldsymbol{\theta}$ with $\Delta\boldsymbol{\theta} \sim \mathcal{U}(-0.1 \cdot \mathbf{1}, 0.1 \cdot \mathbf{1})$, where a new realization is drawn at each switch instance. The results can seen in Figure 5-7, where the first replayed measurement is received at $k = 202$ with $\Delta i = 34$. Note that up until T_a the trajectory and outputs

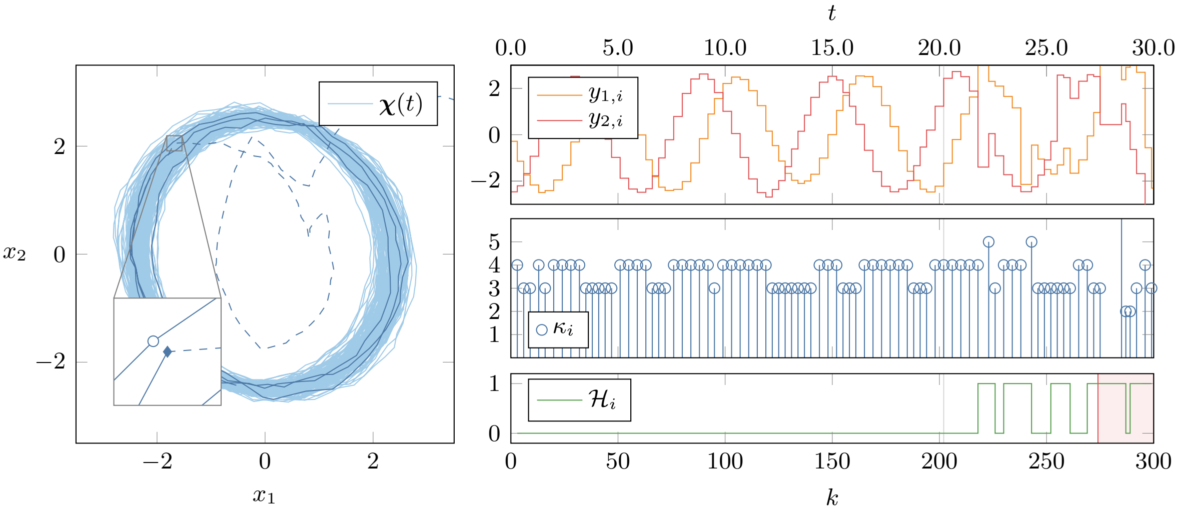
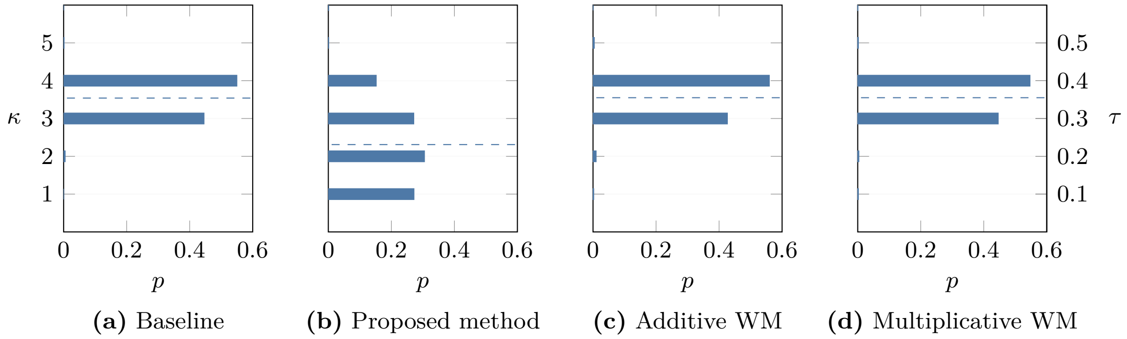


Figure 5-7: Effect of a replay attack with multiplicative watermarking (*benchmark*)

are identical to the baseline, as expected. From the middle plot, we see that from $k \geqslant 218$ onwards, when the parameter vector $\boldsymbol{\theta}(t)$ has been updated, different IETs are requested due to the erratic behavior of $\boldsymbol{y}_i$.

A switch in the parameter $\boldsymbol{\theta}(t)$ happens at $t = T_\mathrm{a}$ and the replay attack remains undetected until the next switch happens at $t = 21.5$, and the attack is detected. Although the outputs are replayed, we can see that for $t \geqslant 21.5$ they start to behave erratically. This is due to the difference in the parametrization of the replayed outputs and the equalizer which are no longer in sync.



(a) Baseline **(b)** Proposed method **(c)** Additive WM **(d)** Multiplicative WM

Figure 5-8: Comparison of the IETs for the four scenarios considered

Next, the IETs for all four scenarios are investigated. Each scenario was simulated in NCSIM for $T = 10^4$ time units without any replay attack, from which the statistics about the IETs were obtained. The average inter-event index $\vec{\tau}_\mathrm{avg} = 1/N_\mathrm{i} \cdot \sum_{i=1}^{N_\mathrm{i}} \tau_i$, where N_i denotes the total number of events during the simulation time T. The average inter-event index $\vec{\kappa}_\mathrm{avg}$ is defined similarly. For the baseline seen in Figure 5-8a, we find that $\vec{\tau}_\mathrm{avg} \approx 0.35$, indicated by the dashed line in the right plot, and $2 \leqslant \kappa \leqslant 5$ (i.e. $\kappa_\mathrm{min} = 2$, $\kappa_\mathrm{max} = 5$).

In Figure 5-8b the IETs for the proposed watermarking method can be seen. Here, $\vec{\tau}_\mathrm{avg} \approx 0.23$ and $1 \leqslant \kappa \leqslant 7$, meaning the average IET is significantly lower then that of the baseline. This is the price one pays for the ability to detect replay attacks, as early triggering obviously decreases IETs. The plot also indicates a more uniform distribution of κ_i, as expected.

Finally, the two benchmarks can be seen in Figure 5-8c and Figure 5-8d, where for additive watermarking $\vec{\tau}_{\text{avg}} \approx 0.36$ and $1 \leqslant \kappa \leqslant 5$, which is remarkably close to the baseline and with a similar distribution. As expected, the multiplicative watermarking strategy has an identical average, minimal, and maximal IET compared to the baseline as well as the same distribution.

5-5-1 Quantitative comparison

In this section, the baseline scenario is compared to both the proposed STC watermarking strategy and additive watermarking. Multiplicative watermarking is excluded given its identical characteristics under nominal system operation.

Apart from the average IET $\vec{\tau}_{\text{avg}}$ as seen in the previous section, a natural metric of choice for (periodic) reference tracking is the average squared tracking error $E_{\text{r,avg}} = 1/(T_{\text{sim}}/h) \cdot \sum_{k=0}^{T_{\text{sim}}/h} \|\boldsymbol{r}[k] - \boldsymbol{x}[k]\|^2$ [12], as well as $E_{\text{r,max}} = \max_k \|\boldsymbol{r}[k] - \boldsymbol{x}[k]\|^2$. Furthermore, the average actuation $U_{\text{avg}} = 1/N \cdot 1/(T_{\text{sim}}/h) \cdot \sum_{k=0}^{T_{\text{sim}}/h} |u[k]|^2$ will also be compared for all three method. Finally, as argued in §3-2, the average squared change in actuation input $\Delta U_{\text{avg}} = 1/(T_{\text{sim}}/h) \cdot \sum_{k=0}^{T_{\text{sim}}/h} |u[k+1] - u[k]|^2$ is used as a comparative metric [129], as one of the benefits of STC is fewer actuation changes.

To summarize, the performance of the five metrics can be seen in Figure 5-1. Here, the trade-off for our proposed method as well as that of additive watermarking becomes apparent. For our proposed method, the earlier triggering leads in a decrease of $\bar{\tau}_{\text{avg}}$, which is undesirable and constitutes a higher degree of utilization of the network. However, the average squared change in actuation input ΔU_{avg} is even lower than the baseline, which is desirable. On the contrary, additive watermarking significantly increases ΔU_{avg}, leading to higher deterioration of the actuators. Furthermore, due to the random nature of the additive random noise on the input, the maximum tracking error is also slightly higher. Interestingly, the average squared tracking error remains almost the same for the additive watermarking method, which is somewhat unexpected. This could be explained by the fact that the random noise sometimes nudges the input in a more optimal direction (offsetting the times when the opposite happens), as the dynamical controller does not take into account the aperiodic sampling policy. More sophisticated (tracking) controller designs that take both future references and aperiodic sampling into account are left for future work.

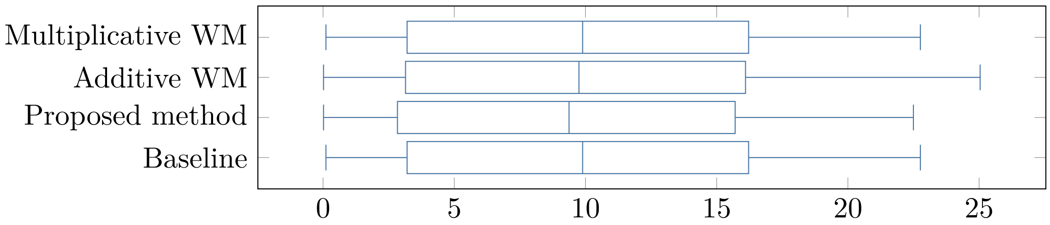


Figure 5-9: Boxplot of the squared tracking error $\|\boldsymbol{r}[k] - \boldsymbol{y}[k]\|^2$

Table 5-1: Summary of the trade-offs between the different watermarking schemes

Method	$\vec{\tau}_{\text{avg}}$	$E_{\text{r,avg}}$	$E_{\text{r,max}}$	U_{avg}	ΔU_{avg}
Baseline	0.35	16.2	22.8	2.21	2.04
Proposed method	0.23 **−34.3%**	15.7 −3.1%	22.5 −1.2%	2.14 −3.4%	1.63 **−20.2%**
Additive WM	0.36 +2.9%	16.1 −0.7%	25.0 **+10.0%**	2.24 +1.1%	3.45 **+69.6%**
Multiplicative WM	0.35 –	16.2 –	22.8 –	2.21 –	2.04 –

5-5-2 Qualitative comparison

Apart from quantitative differences, several qualitative differences need to be considered as well. For each each of the two benchmarks we compare them with our proposed method.

What both benchmarks and the proposed method have in common is that they can be applied to legacy systems [43, 110], which is in part due to them being modular. The watermarking scheme can be designed after the closed-loop control system has been designed (this is not the case for e.g. [9]). This is opposed to encryption, which often comes with prohibitive computational costs on the microprocessors at the sensors and furthermore requires additional bandwidth [118], violating the real-time constraints in NCPS [140]. Secondly, we have shown that all three methods are capable of detecting replay attacks, whilst the baseline is not.

In practice, injecting random noise into the actuation signal leads to jittering, which can be harmful to the actuators and lead to increased wear and tear [41]. Therefore, apart from the performance loss, additive watermarking is often not desirable due to actuator deterioration. Our proposed early triggering scheme does not further burden the actuators, similar to multiplicative watermarking. Another advantage is that, whilst additive watermarking increases the control cost, early triggering guarantees the same (upper bound to) the control cost (see Theorem 5.1) and in practice, often leads to increased performance. Finally, whilst [91] propose an optimization problem to select an optimal $\mathbf{\Sigma}_{\text{u}}$ (but only for a static controller $\boldsymbol{K}$), the tunable parameter in their optimization problem is the loss in LQG performance, which might not be a very intuitive metric in practice.

The biggest difference between our proposed method and multiplicative watermarking is that our watermarking policy is completely contained from the controller side. In multiplicative watermarking, the watermarker $\mathcal{W}$ is located at the sensors. This has the drawback that synchronization between $\mathcal{W}$ and $\mathcal{Q}$ must be guaranteed at all times, which can be a challenging task in practice [7]. Furthermore, whenever the watermarker $\mathcal{W}$ breaks down or the signal $[\![\boldsymbol{y}_i]\!]$ is requested elsewhere, multiplicative watermarking results in a loss of *availability*, which is the absolute priority in ICSs [39, 118, 107]. Robust control during parameter desynchronization has been discussed in [43], whilst a protocol ensuring synchronization has been proposed in [42]. Still, desynchronization might be lost when a package is lost at switching time, which remains to be further investigated. As explored in [137], if the adversary is aware of the multiplicative watermarking scheme he might try to identify the parameter vector $\boldsymbol{\theta}$, to remove the effects of watermarking. A higher filter order N_w and faster switching are therefore desirable, but this again increases computational load at the sensors[3], which are often severely computationally limited [96]. As a final note, concrete design guidelines on how to appropriately choose the filter order N_w as well as the frequency at which the parameter vector $\boldsymbol{\theta}$ needs to be updated are not provided.

As the actual detection is not performed based on the consistency of the timing, but rather on the contents of the packages themselves, this naturally provides additional robustness as opposed to a scheme that exploits consistency in IETs[4]. Finally, we believe the proposed scheme can be extended upon, and more sophisticated co-design methods where both the control input and next sampling instance are designed in tandem to achieve both good performance, as well as reliable attack detection, are possible. This highlights another benefit of STC (as opposed to PETC), as it gives additional freedom on the controller side.

False alarm rate

To demonstrate how well the true false alarm rate is approximated in simulation by the threshold $\eta = 13.82$, the system is initialized at $\boldsymbol{x}_0 = \mathbf{0}$ and the simulation run for $N = 10^5$ event-times in the attack-free case. In compliance with standard Monte Carlo techniques, the first dozen samples are discarded to improve stationarity. We can compute the estimated false alarm rate $\hat{p}_\mathrm{fp}$ as being the parameter from an independent and identically distributed (i.i.d.) Bernoulli random variable, given that the distribution of the detection signal is supposedly a χ^2 distribution with $n_\mathrm{y} = 2$ degrees of freedom. The success parameter $\hat{p}_\mathrm{fp}$ and s.e. can be calculated according to [25]

$$\hat{p}_\mathrm{fp} = \frac{1}{N} \cdot \sum_{i=0}^{N-1} \mathbb{1}_{\{g_i \geqslant \eta\}}, \qquad \text{s.e.}(\hat{p}_\mathrm{fp}) = \sqrt{\frac{\hat{p}_\mathrm{fp} \cdot (1 - \hat{p}_\mathrm{fp})}{N}}. \tag{5-24}$$

The results can be seen in Table 5-2, where we recall that $p_\mathrm{fp} = 0.1\%$. From these results, we see that the estimated false alarm rate is within three s.e.s of the true value, indicating that threshold η is indeed in line with the false alarm rate $p_\mathrm{fp} = 0.1\%$.

5-6 Shortcomings

On the contrary of Proposition 5.2, whenever $\|\boldsymbol{C}\boldsymbol{B}\boldsymbol{u}_i\| \ll \|\boldsymbol{C}(\boldsymbol{A}\boldsymbol{x}_i + \boldsymbol{E}\boldsymbol{\Sigma}_\mathrm{w}) + \boldsymbol{\Sigma}_\mathrm{v}\|$ the distributions of the residuals for all $1 \leqslant \kappa \leqslant \bar{\kappa}$ become very similar (i.e. their means are close

[4] Such a scheme designed for PETC has been explored as well, and its implementation is available in NCSIM but not discussed here any further.

Table 5-2: Estimated false alarm rate $\hat{p}_\mathrm{fp}$ and reported deviations in standard error (s.e.)

	Baseline	Proposed method	Additive watermarking
$\hat{p}_\mathrm{fp}$ (± s.e.)	0.139 (±0.022)%	0.148 (±0.019)%	0.121 (±0.021)%
# of SDs	1.74	2.61	1.01

together, see Figure 5-3b). In that case, the noise dominates and early triggering does not aid in the detection of replay attacks. The former usually occurs whenever the state trajectory is close to the origin.

As the trajectory is close to the origin, the discrete uniform watermarking is no longer optimal. Therefore, we switch to an online watermarking procedure with $\gamma = 0$ as described by 5-17. Whilst the optimization problem needs to be solved online this does not appear to be prohibitive, considering 5-17 is a QP with a relatively low number of decision variables and few constraints.

Using the same parameters as in §5-5, a replay attack is initiated when the system is in steady state. This attack is not detected before the trajectory leaves the safe region at $t = 25.8$ as can be seen in Figure 5-10. The adversary picks $\Delta i = 50$ for all following scenarios as the reference period no longer needs to be taken into account. The first replayed measurement is received at $k = 201$. Note that an alarm is triggered at $k = 297$, but at this time the attack has already become disruptive.

To remedy this shortcoming and provide security even in steady state, we propose one of the following solutions whenever $\boldsymbol{r}[k]$ is constant (for an extended period of time):

- **Switch watermarking method**: The system switches to an alternative watermarking strategy such as additive watermarking[5]capable of detecting replay attacks in steady state.
- **Increase $\kappa_{\max}$**: By increasing both $\bar{\kappa}$ and ϵ the state trajectories wander further away from the stationary reference, which allows for the detection of replay attacks.
- **Dynamic late-triggering through $\eta(s)$**: By the introduction of a (carefully constructed) *buffer variable* $\eta(s)$, late-triggering of the STC policy could be accomplished [49]. Such dynamic triggering conditions have even been shown to perform better in the case of external disturbances [21].

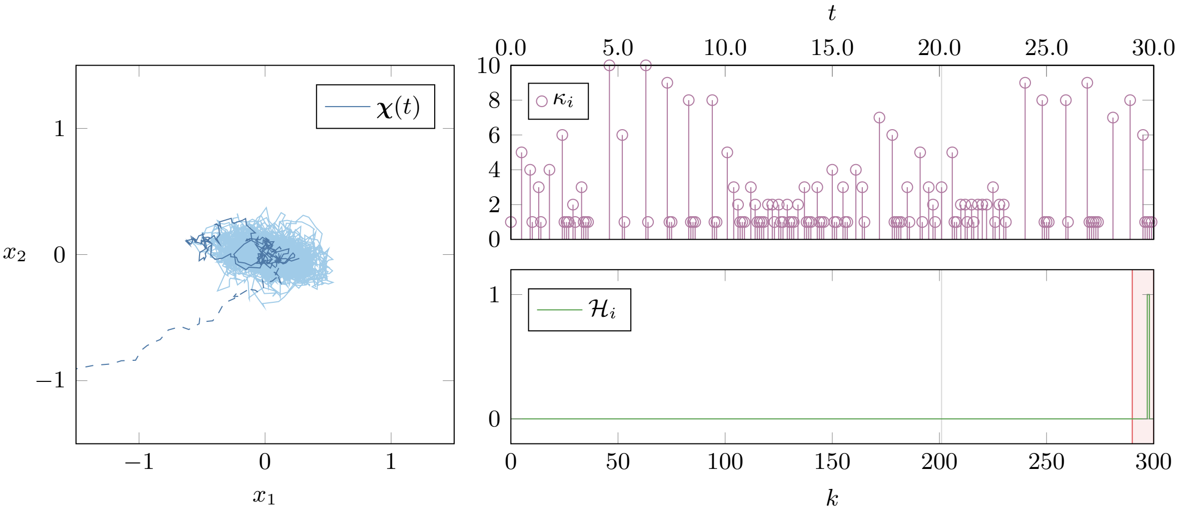


Figure 5-10: Effect of a replay attack during steady state regulation with the proposed method with $\sigma = 0.32$, $\epsilon = 0$, $\bar{\kappa} = 10$ and online watermarking

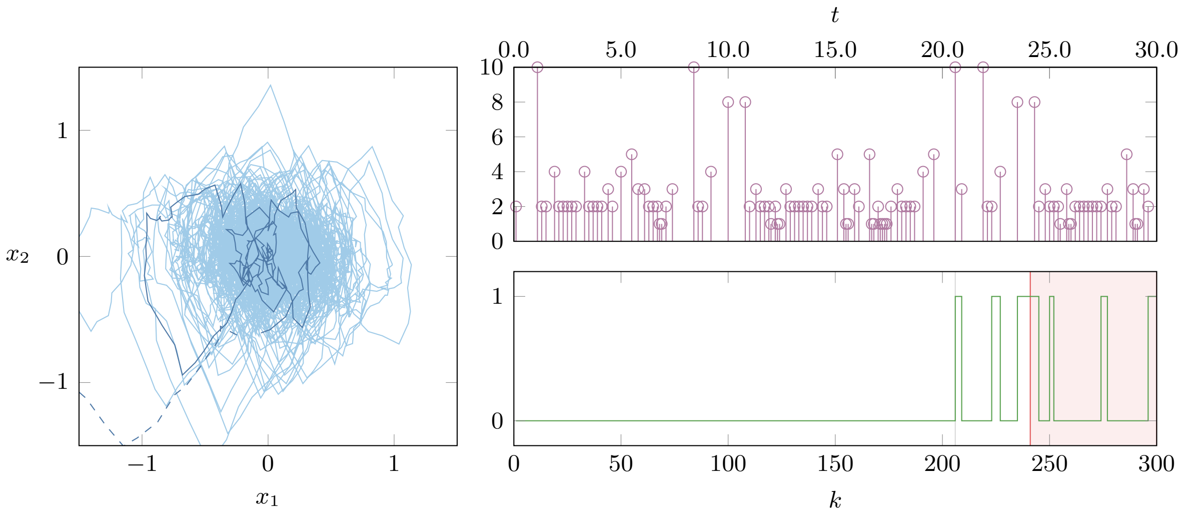


Figure 5-11: Effect of a replay attack during steady state with additive watermarking

Whilst the latter of the three proposed solutions is interesting, this option is not further explored here as its construction requires considerable care to preserve stability. Therefore, this is left to future work. In Figure 5-11 the effect of additive watermarking can be seen. Here, the input covariance matrix $\mathbf{\Sigma}_\mathrm{u} = 5.1 \cdot 10^{-2}$ is found using the same procedure as described in 5-5. As evident from the right plot the loss of control performance is significant. The replay attack is, however, detected before the trajectory leaves the safe region at $t = 24.1$.

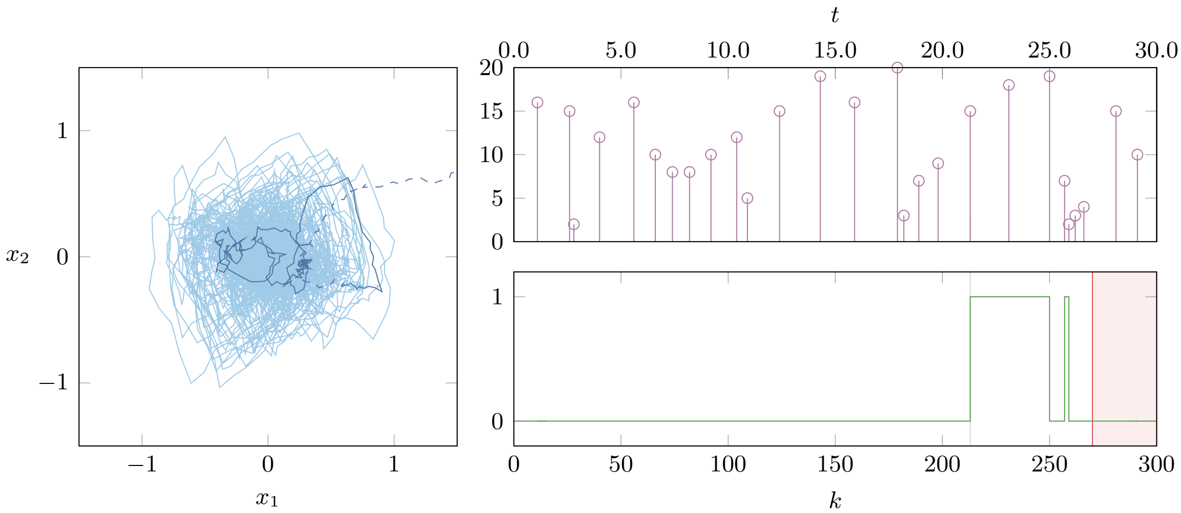


Figure 5-12: Effect of a replay attack during steady state regulation with the proposed method with $\sigma = 0.32$, $\epsilon = 1$ and $\bar{\kappa} = 20$

Finally, the proposed STC watermarked policy can be modified such that replay attacks are detected in steady state (but again, at a significant loss in control performance). To

[5] The reason that we consider only additive watermarking here instead of also multiplicative watermarking is that multiplicative watermarking needs additional configurations at the sensors to be setup.

this extent, consider switching to an early triggering STC policy with $\bar{\kappa} = 20$ and $\epsilon = 1$, for which the resulting trajectory can be seen in Figure 5-12. Note that increasing ϵ is guaranteed to preserve stability by [21, Theorem 2], as the system Figure 3-2 can be shown to be homogeneous (see [21, Definition 4]) as shown in Appendix A-1. The replay attack is detected at $k = 213$ before the state trajectory leaves the safe region at $t = 27.0$. However, similar to additive watermarking, a significant loss of control performance is evident when compared to Figure A-3.

To summarize, we have illustrated that the proposed early triggering STC policy can detect replay attacks when considering the task of tracking a periodic, non-constant reference. In the case of a stationary reference, additional modifications need to be made to still guarantee attacks will be detected. From the comparison between different watermarking methods, each had different trade-offs, making direct superiority ambiguous.

Chapter 6

Switched zero dynamics attacks

In Chapter 5, it was demonstrated how a replay attack can have a devastating effect on the safety of a control system. These attacks are hard to detect because the distribution of replayed outputs (and thus of the residuals) is indistinguishable from the distribution of genuine outputs. Consequently, detection of an attack that results in no output at all is thus virtually impossible to detect (without active countermeasures). These correspond to 0-stealthy attacks in light of Definition 4.3 [123, Lemma 1], and are aptly named zero dynamic attacks (ZDAs). In this chapter, first the relevant notions concerning ZDAs are reiterated. Then, we show that control systems with an self-triggered control (STC) policy are vulnerable to what we call *switched* ZDAs, and finally we show possible countermeasures for control systems to provide resilience to such attacks.

6-1 Zero dynamics attacks

A ZDA is an open loop stealthy attack which, under certain circumstances, can be disruptive as well. In time-triggered control (TTC), with perfect model knowledge (see Assumption 9), an adversary can compute such an attack offline, i.e. *a priori*, meaning it is independent of the inputs and outputs of the plant [114]. We show that for STC, a successfully switched ZDA does need to leverage disclosure resources of the inter-event times (IETs) (see Figure 4-2).

In the following sections, we first present our result for the noise-free case with $\dot{\boldsymbol{\omega}}(t), \boldsymbol{\nu}(t) = \mathbf{0}$ for all t, and initial condition $\boldsymbol{x}_0 = \mathbf{0}$. ZDAs are usually initiated during steady state so that the value of $\boldsymbol{x}_0$ is easily guessed (at least approximately) and the error in the dynamics is small [98] (see Proposition 6.2). The influence of noise and a non-zero initial condition will be demonstrated in §6-5-1. Below, the defining characteristic of a ZDA is described, where the definition has been adapted to control systems with an STC policy.

Definition 6.1 (Zero dynamic attack)**.** *A* zero dynamic attack *is an actuator attack which satisfies*

$$\boldsymbol{a}_{\mathrm{u},i} \neq \mathbf{0} \qquad \text{s.t.} \qquad \boldsymbol{y}_i = \mathbf{0}, \tag{6-1}$$

for all $i \geqslant 0$*. Without loss of generality, assume the attack starts at* $t_0 = 0$*.*

Contrary to Chapter 5, where the adversary did not need to have any knowledge of the control system, here a strong Byzantine adversary is considered. Furthermore, the adversary has the capability to manipulate the packages sent over the C2A channel (see Figure 3-2).

Assumption 9. *The information available to the adversary* $\underset{\sim}{\mathcal{A}}$ *at time* $t \in [t_i, t_{i+1})$ *is given by* $\mathbb{I}_{\mathrm{a}}(t) \supseteq \{t_{i+1} \wedge \underset{\sim}{\mathcal{P}}, \mathcal{C}\}$ *with* $\underset{\sim}{\mathcal{P}}$ *and* $\mathcal{C}$ *given by* (2-1) *and* (2-2)*, respectively.* ◊

For simplicity, we restrict ourselves to single-input and single-output (SISO) systems for the remainder of this chapter. Extensions to multiple-input and multiple-output (MIMO) systems are possible and for the most part straightforward (see e.g. Definition B.1), see e.g. [66]. It is important to realize that MIMO systems are susceptible to the same vulnerabilities as discussed later in the chapter. Secondly, we further restrict ourselves to stable systems $\underset{\sim}{\mathcal{P}}$ in this chapter for brevity, but again, the extension to unstable systems (with at least one eigenvalue for which $\mathfrak{Re}\{\underset{\sim}{\boldsymbol{A}}\} < 0$) is possible under further constraints. This can be done by taking into account only the stable eigenspace of $\underset{\sim}{\boldsymbol{A}}$ (see e.g. [123]).

The attacks described in Definition 6.1 correspond to the solutions of the *output-zeroing problem*, a related notion from geometric control (see §6-2-1). These solutions are determined by the zeros of $\underset{\sim}{\mathcal{P}}$, which can be explained by taking the Laplace transform of (2-1a), (2-1b) from which we can obtain [124]

$$\underbrace{\begin{bmatrix} s \cdot \boldsymbol{I} - \underset{\sim}{\boldsymbol{A}} & -\underset{\sim}{\boldsymbol{B}} \\ \boldsymbol{C} & 0 \end{bmatrix}}_{\boldsymbol{R}(s)} \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{U} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{6-2}$$

Here, $\boldsymbol{R}(s)$ is called the Rosenbrock system matrix. For SISO systems, the values $s \in \mathbb{C}$ for which $\boldsymbol{R}(s)$ is no longer invertible are called the *zeros* of $\underset{\sim}{\mathcal{P}}$. The related definition of a *transmission zero* in the case of MIMO systems can be found in Appendix B.

The zeros of a system explain how a ZDA can be *stealthy* (see Definition §4.3). Yet, these attacks in and of themselves are not necessarily dangerous, as they might be of finite energy and thus inflict no damage to the control system [114] (i.e. such attacks do not succeed in pushing $\boldsymbol{\chi}(t)$ outside $\mathbb{X}_{\mathrm{s}}$). However, whenever their exist zeros with $\mathfrak{Re}\{s\} > 0$, implying $\underset{\sim}{\mathcal{P}}$ is non-minimum phase, then ZDAs become *disruptive* as well, making them dangerous.

6-1-1 Sampling zeros

One might reasonably expect that if $\underset{\sim}{\mathcal{P}}$ is minimum-phase, then the system is safe from ZDAs. However, this is often not the case due to the introduction of (unstable) *sampling zeros* when considering a sampled-data system as in Figure 3-2. For SISO systems, the relative degree $n_\nu \in \mathbb{N}_0$ is scalar and equal to the number of poles minus the number of zeros of $\underset{\sim}{\mathcal{P}}$. For MIMO systems, the (vector-valued) notion of relative degree for systems with $n_\mathrm{u} = n_\mathrm{y}$ can be found in Appendix B-2.

Regardless of the relative degree n_ν of $\underset{\sim}{\mathcal{P}}$, the discretization $\mathcal{P}$ will have a relative degree of 1 for almost all sampling periods h [114]. This means that zero-order hold (ZOH) discretization introduces $n_\nu - 1$ additional zeros, aptly named *sampling zeros*. Problematically, these sampling zeros might be unstable even though the original continuous-time system possesses no unstable zeros. Such unstable zeros are guaranteed to occur in some systems if sampling happens sufficiently fast [72].

Theorem 6.1 (Unstable sampling zeros, adapted from [17, Lemma 3.3.3])**.** *Consider a continuous-time plant $\underset{\sim}{\mathcal{P}}$ as in (2-1) and suppose $\underset{\sim}{\mathcal{P}}$ is* SISO *and has relative degree $n_\nu \geqslant 3$. Then, as $h \to 0$ the discretized dynamics $\mathcal{P}$ as in (3-1) will contain at least one unstable sampling zero.*

Note that in many engineering applications, as well as industrial control systems (ICSs), relatively fast sample rates and a relative degree n_ν greater than two are common [135]. Thus, control systems need to be actively protected to prevent ZDAs from being disruptive.

6-2 Construction of a ZDA

For the construction of ZDAs there exist three main methods in the literature. The first method is by solving the discrete-time equivalent of the system (6-2) for a (real) unstable zero $z_\mathrm{u} \in \mathbb{C}$, from which the input-zero direction $\boldsymbol{U}$ can be obtained and the ZDA can be constructed as $\boldsymbol{u}[k] = \boldsymbol{U} \cdot z_\mathrm{u}^k$ [124, 70]. Secondly, the notion of a *control invariant subspace* can be employed, and the adversary can evolve a special linear time-invariant (LTI) system alongside the evolution of the dynamics as in (3-1). The output of this system constructs the attack vector $\boldsymbol{a}_{\mathrm{u},i}$, through the design of a feedback matrix $\boldsymbol{F}$ [123, 99] (see §6-2-1). Finally, the system $\mathcal{P}$ can be converted to the so-called *Byrnes-Isisdori normal form* given by

$$\mathcal{P} \quad : \quad \begin{aligned} \boldsymbol{x}_\mathrm{z}[k+1] &= \boldsymbol{S}\boldsymbol{x}_\mathrm{z}[k] + \boldsymbol{P}\bar{\boldsymbol{C}}\boldsymbol{x}_\mathrm{o}[k], && \text{(6-3a)}\\ \boldsymbol{x}_\mathrm{o}[k+1] &= \bar{\boldsymbol{A}}\boldsymbol{x}_\mathrm{o}[k] + \bar{\boldsymbol{B}}(\boldsymbol{b}_\mathrm{z}^\mathrm{T}\boldsymbol{x}_\mathrm{z}[k] + \boldsymbol{b}_\mathrm{o}^\mathrm{T}\boldsymbol{x}_\mathrm{o}[k] + \boldsymbol{b}_\mathrm{u}^\mathrm{T}\boldsymbol{u}[k]), && \text{(6-3b)}\\ \boldsymbol{y}[k] &= \bar{\boldsymbol{C}}\boldsymbol{x}_\mathrm{o}[k], && \text{(6-3c)} \end{aligned}$$

where $\boldsymbol{x}_\mathrm{o} \in \mathbb{R}^{n_\mathrm{z}}$, $\boldsymbol{x}_\mathrm{z} \in \mathbb{R}^{n_\mathrm{x}-n_\mathrm{z}}$ and the structure of $\boldsymbol{S}$, $\boldsymbol{P}$, $\bar{\boldsymbol{A}}$, $\bar{\boldsymbol{B}}$, $\bar{\boldsymbol{C}}$, $\boldsymbol{b}_\mathrm{z}^\mathrm{T}$, $\boldsymbol{b}_\mathrm{o}^\mathrm{T}$ and $\boldsymbol{b}_\mathrm{u}^\mathrm{T}$ can be found in [69]. Here, $n_\mathrm{z} \in \mathbb{N}_0$ denotes the relative degree of the discretization $\mathcal{P}$ (which might be different from n_ν) and $\boldsymbol{x}_\mathrm{z}[k]$ denote the zero dynamics. A PYTHON implementation[1] for obtaining (6-3) given the dynamics as in (3-1) can be found in Appendix B-4.

Since the first method proposed above is not directly extendable to switched linear (SL) systems, only the latter two methods have been incorporated in NCSIM. For brevity, we

[1] There seems to be no readily available algorithm to construct (6-3), and as such one is provided here.

here only discuss how our proposed switched ZDAs in the framework of a controlled invariant subspace.

6-2-1 Controller invariant subspace

As mentioned in the previous section, the adversary can exploit knowledge in geometric control, together with Assumption 9, to construct a ZDA based on the notion of a *controlled invariant subspace*. The definition of a controlled invariant subspace is given below.

Definition 6.2 ((Controlled) invariant subspace [10])**.** *A subspace $V \subseteq \mathbb{R}^{n_\mathrm{x}}$ is an* $\boldsymbol{A}$-invariant subspace *if and only if $\boldsymbol{A}V \subseteq V$, where $\boldsymbol{A}V = \mathrm{span}\{\boldsymbol{A}\boldsymbol{v}_1, \dots, \boldsymbol{A}\boldsymbol{v}_p\}$ and the collection $\boldsymbol{v}_1, \dots, \boldsymbol{v}_p$ is a basis for V with $\dim(V) = p \leqslant n_\mathrm{x}$. Furthermore, V is an* $(\boldsymbol{A}, \boldsymbol{B})$-controlled invariant subspace *with $\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_1 & \cdots & \boldsymbol{b}_{n_\mathrm{u}} \end{bmatrix}$ if and only if $\boldsymbol{A}V \subseteq V + \mathrm{span}\{\boldsymbol{b}_1, \dots, \boldsymbol{b}_{n_\mathrm{u}}\}$.*

Equivalently, V is a $(\boldsymbol{A}, \boldsymbol{B})$-controlled invariant subspace if and only if there exists a feedback matrix $\boldsymbol{F}$ such that V is a $(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{F})$-invariant subspace.

Intuitively, a controlled invariant subspace in a subspace for which there exists a control input for each state vector inside the subspace, such that under the evolution of the system dynamics, this control input pushes the trajectory back inside the subspace. Due to the dynamics $\mathcal{P}$ being LTI, this control input can be constructed by considering static full-state feedback.

Since we are interested in constructing an attack vector such that the output of the system remains zero (see Definition 6.1), we aim to find the maximal controlled invariant subspace $V^\star$ which is contained in the nullspace of $\boldsymbol{C}$. More formally, $V^\star \subseteq \ker(\boldsymbol{C})$ is the subspace such that for all $V' \supset V^\star$, this implies $V' \nsubseteq \ker(\boldsymbol{C})$. Below, we propose an algorithm to find both the maximal controlled invariant subspace $V^\star$ as well as the accompanying feedback matrix $\boldsymbol{F}$.

It must be noted that whilst line number 14 in Algorithm 1 involves solving an underdetermined linear least squares (LLS) problem, the degrees of freedom are exactly equal to $n_\mathrm{x} - n_\mathrm{z}$, which is almost always one. Therefore, n_z of the eigenvalues of $\boldsymbol{A} + \boldsymbol{F}\boldsymbol{B}$ correspond with the zeros of $\mathcal{P}$ [99] (with corresponding eigenspace $V^\star$), and as such if $\mathcal{P}$ contains an unstable zero then this implies $\boldsymbol{A} + \boldsymbol{F}\boldsymbol{B}$ is unstable [123].

6-2-2 Existing countermeasures

The existing countermeasures for ZDAs in the literature are focused on TTC. However, many of these can be extended to be compatible with aperiodically sampled systems (or require no modification at all). Below, a (non-comprehensive) overview of several common strategies are given.

One solution proposed in [123] (among others proposed in the same paper) is to modify the $\boldsymbol{C}$ by deploying additional measurements. This solution might be a viable approach in practice, as sensors are not only often one of the most inexpensive parts of the control system, but redundant observability can also aid in the detection of false data injection (FDI) attacks (see e.g. [15, 74]).

Algorithm 1 Controlled invariant subspace, adapted from [10]

```
Requires: $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$
Returns: $\boldsymbol{F}$ s.t. $(\boldsymbol{A}+\boldsymbol{F}\boldsymbol{B})V^\star \subseteq V^\star$ where $V^\star \subseteq \ker(\boldsymbol{C})$
$\boldsymbol{V} \leftarrow [\ \boldsymbol{v}_1 \ \cdots \ \boldsymbol{v}_{n_\mathrm{k}}\ ]$ s.t. $\mathrm{span}\{\boldsymbol{v}_1,\dots,\boldsymbol{v}_{n_\mathrm{k}}\} = \ker(\boldsymbol{C})$
$\boldsymbol{V}^+ \leftarrow \boldsymbol{0}$
while $\mathrm{range}(\boldsymbol{V}^+) \neq \mathrm{range}(\boldsymbol{V})$ do
    $\boldsymbol{W} \leftarrow \boldsymbol{A}^{\text{-}1}[\ \boldsymbol{V} \ \ \boldsymbol{B}\ ]$
    $\boldsymbol{\Pi}_{WV} \leftarrow \boldsymbol{V}(\boldsymbol{V}^\mathrm{T}\boldsymbol{V})^{\text{-}1}\boldsymbol{V}^\mathrm{T}\boldsymbol{W}(\boldsymbol{W}^\mathrm{T}\boldsymbol{W})^{\text{-}1}\boldsymbol{W}^\mathrm{T}$
    $\boldsymbol{\lambda}, \boldsymbol{E} \leftarrow$ Eigenvalues and eigenvectors of $\boldsymbol{\Pi}_{WV}$
    $\boldsymbol{V}^+ \leftarrow [\ \boldsymbol{e}_1 \ \cdots \ \boldsymbol{e}_{n_\mathrm{v}}\ ]$ s.t. $\boldsymbol{e}_i \in \mathrm{range}(\boldsymbol{E})$ and $\lambda_i = 1$
    if $\mathrm{range}(\boldsymbol{V}^+) = \mathrm{range}(\boldsymbol{V})$ then
        break                                  ▷ If $\boldsymbol{V}^+ = \boldsymbol{0}$, then $V^\star = \varnothing$, return error
    else
        $\boldsymbol{V} \leftarrow \boldsymbol{V}^+$
$\boldsymbol{\Phi}^\mathrm{T} \leftarrow$ Solution of $[\ \boldsymbol{V} \ \ \boldsymbol{B}\ ]\boldsymbol{\Phi} = \boldsymbol{A}\boldsymbol{V}$
$\boldsymbol{U} \leftarrow [\ \boldsymbol{\phi}_{n_\mathrm{x}+1} \ \cdots \ \boldsymbol{\phi}_{n_\mathrm{x}+n_\mathrm{u}}\ ]$
$\boldsymbol{F} \leftarrow \arg\min \|\boldsymbol{V}^\mathrm{T}\boldsymbol{F} + \boldsymbol{U}^\mathrm{T}\|$       ▷ LLS on underdetermined system
```

Considering a set of available additional measurements $\mathbb{S} = \{\boldsymbol{c}_1^\mathrm{T}, \dots, \boldsymbol{c}_{n_\mathrm{s}}^\mathrm{T}\}$, an algorithm for computing the least amount of sensors needed such that $\ker(\boldsymbol{C})$ no longer contains a controlled invariant subspace can be found in Appendix B-5.

As opposed to changing the sensors, [69] propose modifying the actuators to incorporate a generalized hold function $h_\mathrm{g} : [0, h) \to \mathbb{R}$. Proper construction of such a generalized hold moves the unstable zeros inside the unit circle. By extending these results for aperiodic sampling, their optimal hold function $h_\mathrm{g} : [0, \kappa \cdot h) \to \mathbb{R}$ such that $\boldsymbol{v}(t) = h_\mathrm{g}(s) \cdot \boldsymbol{u}_i$ for $t \in [t_i, t_{i+1})$ is given by

$$h_\mathrm{g}(s\,;\kappa) = \underset{\sim}{\boldsymbol{B}}^\mathrm{T} e^{\underset{\sim}{\boldsymbol{A}}^\mathrm{T}(h\cdot\kappa - s)} \int_0^{h\cdot\kappa} e^{\underset{\sim}{\boldsymbol{A}}\cdot t} \underset{\sim}{\boldsymbol{B}}\underset{\sim}{\boldsymbol{B}}^\mathrm{T} e^{\underset{\sim}{\boldsymbol{A}}^\mathrm{T}\cdot t}\, \mathrm{d}t \boldsymbol{O}^{\text{-}1}\boldsymbol{O}'\boldsymbol{B}', \tag{6-4}$$

where $\boldsymbol{O}$ is the observability matrix of the pair $(\boldsymbol{A}, \boldsymbol{B})$ and $\boldsymbol{O}'$ is the observability matrix of the pair $(\boldsymbol{A}', \boldsymbol{B}')$. Here, $\boldsymbol{A}'$, $\boldsymbol{B}'$ are a minimal realization of a transfer function with identical poles and gain as in [69, Remark 1], but zeros which can be chosen such that they are inside the unit circle. The explicit dependence of the inter-event index κ for all of the aforementioned matrices has been omitted. Note that the hold function h_g is parameterized by κ and thus needs to change based on the next IET.

The two existing countermeasures, as well as several others, are compared to the proposed solution in §6-5-3. Note that only a qualitative comparison is performed since no common metric can compare the effectiveness of the three methods. Resilience to ZDA is a binary property, but advantages and disadvantages can still be discussed.

Partial system knowledge

Assumption 9 can be relaxed for adversaries having only partial system knowledge. For example, whenever the sampling period is very small, i.e. $h \to 0$, the sampling zeros approach the (publicly available) roots of the Euler-Frobenius polynomial, independent of the system parameters. As such, an adversary might succeed in constructing a ZDA in such a case with only the information on the relative degree of the system [114].

The effects of quantization were investigated in [70], potentially increasing the chance of detection through the output of the system. However, they show that the adversary may reduce such errors by avoiding direct quantization of the attack vector, thereby succeeding in constructing a ZDA. In [98], a robust ZDA is described where the adversary has imperfect knowledge of the plant dynamics $\underset{\sim}{\mathcal{P}}$. They demonstrate that this alternative attack does not require the exact model knowledge anymore, yet manages to be both stealthy and disruptive. The price the adversary has to pay, however, is that disclosure resources of the output measurement are needed. Therefore, the robust ZDA needs to be constructed online.

The robust ZDA as described above shares characteristics with the switched ZDA proposed here. Both need additional disclosure resources (see Figure 4-2) and can not be constructed *a priori*[2]. Whilst this might be a slight complication, it would not be reasonable to assume that this would prevent a strong Byzantine adversary from constructing such a ZDA. Therefore, in the next section switched ZDAs are defined, and afterward, possible countermeasures are discussed.

6-3 Switched zero dynamic attack

We demonstrated how the closed-loop system under an STC policy can be modeled as a SL system as given by (3-7). Here, we leverage this fact and describe how an adversary can use a similar structure to construct a ZDA. An important observation is that for the discretized dynamics $\mathcal{P}$ the measurement matrix $\boldsymbol{C}$ remains unchanged, for any sampling period $h \in \mathbb{R}_{>0}$. Thus, the maximal controlled invariant subspace $V^{\star}$ is independent of h. This given is utilized in the following proposition.

Proposition 6.2 (Switched zero dynamic attack (ZDA))**.** *Consider the* SL *system as in* (3-7) *with* $\boldsymbol{x}_0 = \boldsymbol{0}$ *and suppose the switching indices* $(\kappa_1, \kappa_2, \ldots)$ *are known. Then, the attacker system is given by*

$$\boldsymbol{f}_{i+1} = (\boldsymbol{A}_{\kappa_{i+1}} + \boldsymbol{F}_{\kappa_{i+1}} \boldsymbol{B}_{\kappa_{i+1}}) \boldsymbol{f}_i \tag{6-5}$$

with non-zero initial condition $\boldsymbol{f}_0$ *and output attack vector* $\boldsymbol{a}_{\mathrm{u},i} = \boldsymbol{F}_{\kappa_{i+1}} \boldsymbol{f}_i$*, with* $\boldsymbol{F}_{\kappa}$ *constructed using* Algorithm 1 *for* $1 \leqslant \kappa \leqslant \bar{\kappa}$*, creates a* ϵ*-stealthy* ZDA *as by* Definition 4.3*.*

[2] In the noise-free case, the attack can potentially be constructed offline by predicting the next m IETs. However, the adversary would need exact knowledge of the initial condition $\boldsymbol{x}_0$, which seems impractical.

Proof: The proof is an extension of the result in [10] and follows from an induction argument. Suppose that at index i we have $\boldsymbol{x}_i = \boldsymbol{f}_i \in V^\star \subseteq \ker(\boldsymbol{C})$. Using (3-7) the dynamics will evolve according to $\boldsymbol{x}_{i+1} = \boldsymbol{A}_{\kappa_{i+1}}\boldsymbol{x}_i + \boldsymbol{F}_{\kappa_{i+1}}\boldsymbol{B}_{\kappa_{i+1}}\boldsymbol{f}_i = (\boldsymbol{A}_{\kappa_{i+1}} + \boldsymbol{F}_{\kappa_{i+1}}\boldsymbol{B}_{\kappa_{i+1}})\boldsymbol{x}_i$, which is equal to $\boldsymbol{f}_{i+1}$ as by (6-5). Note that $\boldsymbol{u}_i = \boldsymbol{0}$ since $\boldsymbol{y}_i = \boldsymbol{0}$. Furthermore, this implies $\boldsymbol{x}_{i+1} \in V^\star$ as $V^\star$ is $(\boldsymbol{A}_{\kappa_{i+1}} + \boldsymbol{F}_{\kappa_{i+1}}\boldsymbol{B}_{\kappa_{i+1}})$-invariant by construction of $\boldsymbol{F}_\kappa$ using Algorithm 1. For the non-zero initial condition $\boldsymbol{f}_0 \neq \boldsymbol{0}$, the error dynamics $\tilde{\boldsymbol{f}}_i = \boldsymbol{x}_i - \boldsymbol{f}_i$ (under full-state feedback) given in [123, Theorem 2] can be written as $\tilde{\boldsymbol{f}}_{i+1} = \boldsymbol{A}_{\kappa_{i+1}}\tilde{\boldsymbol{f}}_i$ under an STC policy, with output $\boldsymbol{y}_i = \boldsymbol{C}\tilde{\boldsymbol{f}}_i$ and initial condition $\tilde{\boldsymbol{f}}_0 = -\boldsymbol{f}_0$. Thus, $\|\boldsymbol{y}_i\| = \|\boldsymbol{C}(\boldsymbol{A}_{\kappa_1} \cdots \boldsymbol{A}_{\kappa_i})\boldsymbol{f}_0\| \leqslant \|\boldsymbol{C}\| \cdot \|\boldsymbol{f}_0\|$ for all i, where was have made use of the fact that $\boldsymbol{A}_\kappa$ is Schur stable for all $1 \leqslant \kappa \leqslant \bar{\kappa}$ (see (3-8)) and the triangle inequality. Thus, for all $\epsilon > 0$ there exists an appropriate initial condition $\boldsymbol{f}_0 \in V^\star$ with $\|\boldsymbol{f}_0\| \leqslant \eta/\|\boldsymbol{C}\|$ such that a switched ZDA is ϵ-stealthy, where η is known by the adversary due to Assumption 9. ■

Note that for $\boldsymbol{x}_0 = \boldsymbol{f}_0$ it directly follows that $\boldsymbol{y}_0 = \boldsymbol{0}$ for all $k \geqslant 0$ [123]. Furthermore, the initial condition $\boldsymbol{f}_0$ must preferably be chosen as lying in the unstable eigenspace (if it exists) of the matrix $(\boldsymbol{A}_{\kappa_1} + \boldsymbol{F}_{\kappa_1}\boldsymbol{B}_{\kappa_1})$ such that the attack becomes disruptive (see Theorem 6.5). To summarize, the strategy for the adversary is as follows:

1. With Assumption 9, using $\boldsymbol{A}$, $\boldsymbol{B}$ and (3-8), compute the matrices $\boldsymbol{F}_\kappa$ using Algorithm 1 offline for all $\kappa \in \{1, \ldots, \bar{\kappa}\}$.
2. Choose an sufficiently small initial condition $\boldsymbol{f}_0 \in V^\star \subseteq \ker(\boldsymbol{C})$ such that the attack is ϵ-stealthy with respect to the detector $\mathcal{D}$.
3. At each event time t_i, retrieve κ_{i+1} and input the attack vector $\boldsymbol{a}_{u,i} = \boldsymbol{F}_{\kappa_{i+1}}\boldsymbol{f}_i$ and update $\boldsymbol{f}_{i+1}$ as in (6-5).

The attack strategy proposed above relies largely on two assumptions. For one, the attack vector $\boldsymbol{a}_{\mathrm{u},i}$ needs to be computed at the instant of the i-th event, relying strongly on Assumption 2. Furthermore, we consider that the adversary adheres to the IETs as transmitted by the STC policy. For this reason, the next event time $[\![t_{i+1}]\!]$ is also sent to the actuators (see Figure 3-2). Such a model might be reasonable as a surrogate for the commonly used *to-zero* strategy, here adapted to aperiodic sampling. In many situations, this strategy performs better as opposed to the *to-hold* strategy [112]. Furthermore, as a further incentive for the adversary to adhere to the IETs to avoid detection, the unusually high amount of traffic could possibly be detected on the controller side. Note that whenever an adversary does not adhere to the IETs, then a ZDA might be possible regardless of the countermeasures as proposed in §6-4[3].

6-4 Extending inter-event times

As discussed in §6-1-1, the discretized system will contain unstable sampling zeros when h is small enough (see Theorem 6.1). On the contrary, if h is large enough then the discretized dynamics $\mathcal{P}$ will not contain any unstable zeros. This observation stems from the following theorem.

[3] In fact, a similar reasoning still holds for TTC, as updating the actuators at h/m with $m \in \mathbb{N}$ sufficiently large would still result in unstable sampling zeros, which is stated in Theorem 6.1.

Theorem 6.3 (Stable zeros, adapted from [142, Theorem 2])**.** *Consider a continuous-time plant* $\underset{\sim}{\mathcal{P}}$ *as in* (2-1) *and suppose* $\underset{\sim}{\mathcal{P}}$ *is* SISO*, stable, and has relative degree* $n_\nu \geqslant 1$*. Furthermore, suppose* $\underset{\sim}{\mathcal{P}}$ *has no zeros at the origin. Then, as* $h \to \infty$ *the discretized dynamics* $\mathcal{P}$ *as in* (3-1) *will contain only stable zeros.*

Furthermore, a lower bound $\underline{h}$ on the sampling period such that the above holds, here stated only for simple poles, is given below.

Lemma 6.4 (Lower bound to stable zeros, [93, Theorem 1])**.** *A lower bound* $\underline{h}$ *such that* Theorem 6.3 *does not hold is given by*

$$\underline{h} = \frac{\log\left(2 \cdot \alpha \cdot (n_{\mathrm{x}} + 1)\right)}{|\min \mathfrak{Re}\{\lambda(\underset{\sim}{\boldsymbol{A}})\}|}, \tag{6-6}$$

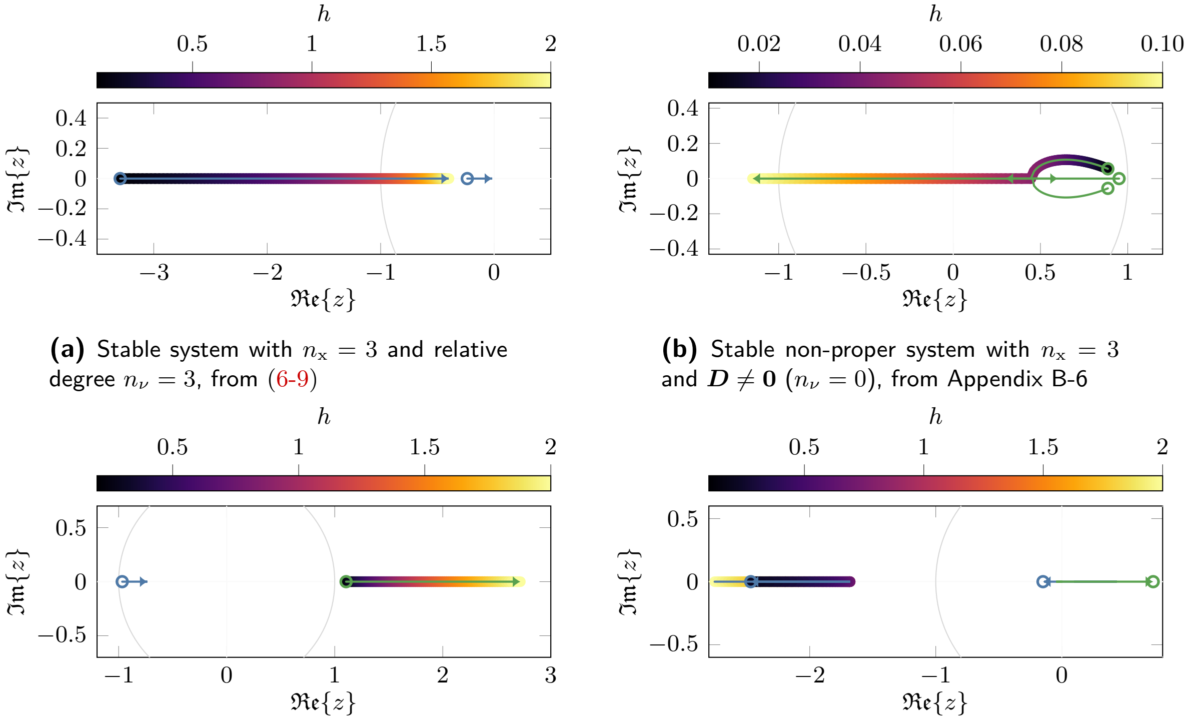
where $\alpha = \max_i |a_i/P(0)|$ *and* $\underset{\sim}{P}(s) = \boldsymbol{C}(s \cdot \boldsymbol{I} - \underset{\sim}{\boldsymbol{A}})^{-1}\underset{\sim}{\boldsymbol{B}}$ *is the transfer function which can be written (by means of partial fraction decomposition) as*

$$P(s) = \sum_{i=0}^{n_{\mathrm{x}}} \frac{a_i}{s - b_i}. \tag{6-7}$$

To the best of the author's knowledge, no upper bound $\bar{h}$ for which any sampling period $h > \bar{h}$ guarantees stable zeros is known. Deriving such a bound would be of particular interest for providing a sufficient condition.

Note that both the assumption on $\underset{\sim}{P}$ being stable and having relative degree $n_\nu \geqslant 1$ as in Theorem 6.3 are vital to ensuring the (sampling) zeros will eventually become stable. If these conditions are not met, then in general very little can be said about the behavior of the zeros of the sampled system [97]. This is demonstrated for several different systems in 6-1. For example, one can see in Figure 6-1b that for a non-proper system increasing the sampling period can make stable zeros eventually become unstable. In 6-1c we see that for unstable intrinsic zeros, which follow the same mapping $z \hookleftarrow e^{s \cdot h}$ as the poles [97], the discrete-time zeros as unstable regardless of $h \in \mathbb{R}_{>0}$. Finally, the unstable system in 6-1d demonstrates some exotic behavior where larger sampling times initially seem to move the sampling zero closer to the origin, but for some $h \approx 0.5$ this direction reverses. Conclusively, studying the behavior for sampling zeros of arbitrary dynamics can be unpredictable. However, under certain conditions, we can conclude the following.

Theorem 6.5 (Vulnerability of STC to switched ZDAs)**.** *Consider the closed-loop as in* Figure 3-2 *and suppose* $\underset{\sim}{\mathcal{P}}$ *is* SISO*, has relative degree* $n_\nu \geqslant 3$ *and furthermore* $\kappa_{\max} < \lfloor \underline{h}/h \rfloor$*. Then, for all* $\epsilon > 0$ *their exists a disruptive* ϵ*-stealthy attack such that for some* $t > t_0$ *we have* $\boldsymbol{\chi}(t) \notin \mathbb{X}_{\mathrm{s}}$*.*



(a) Stable system with $n_\mathrm{x} = 3$ and relative degree $n_\nu = 3$, from (6-9)

(b) Stable non-proper system with $n_\mathrm{x} = 3$ and $\boldsymbol{D} \neq \boldsymbol{0}$ ($n_\nu = 0$), from Appendix B-6

(c) Unstable system with $n_\mathrm{x} = 3$ and a unstable intrinsic zero, from [130, §6]

(d) Unstable system with $n_\mathrm{x} = 4$ and a stable intrinsic zero, from [21, §5]

Figure 6-1: Root locus plot of both intrinsic zeros (green) and sampling zeros (blue) for different sampling periods $h \in \mathbb{R}_{>0}$

Proof: Since $\kappa_{\max} < \lfloor \underline{h}/h \rfloor$, according to Lemma 6.4 the largest IET is strictly smaller than the one needed for stable sampling zeros. Since $n_\nu \geqslant 3$ according to Theorem 6.1 the SL system (3-7) has at least one unstable sampling zero for all $\kappa \leqslant \kappa_{\max}$. We can therefore construct a switched ZDA as by Proposition 6.2, where we can choose $\boldsymbol{f}_0$ such that the attack is ϵ-stealthy. Finally, since for all $\kappa \leqslant \kappa_{\max}$ the matrix $\boldsymbol{A}_\kappa + \boldsymbol{F}_\kappa \boldsymbol{B}_\kappa$ has at least one eigenvalue strictly larger then one, we have that $\lim_{i\to\infty} \|\boldsymbol{f}_i\| = \infty$. Thus, their exists some i such that $\|\boldsymbol{f}_i\| \geqslant \rho$. Recalling that $\boldsymbol{x}_i = \boldsymbol{f}_i$, apart from the transient behavior due to the non-zero initial condition $\boldsymbol{f}_0$ (which is negligible even for small i), ρ can be chosen such that $\|\boldsymbol{\chi}(t)\| \geqslant \rho$ implies $\boldsymbol{\chi}(t) \notin \mathbb{X}_\mathrm{s}$ with $t \geqslant t_i$. ■

Inspired by Lemma 6.4, we postulate that extending IETs using an STC policy might be able to prevent ZDAs, as demonstrated in Figure 6-1a. Several illustrative examples are given in the next section. In §5-6 several methods to extend the IETs were proposed. To aid the prevention of ZDAs, here we suggest related although slightly different methods:

1. **Increase** $\kappa_{\min}$: By increasing either ϵ or σ, or both, we can expect $\kappa_{\min}$ to be larger which moves the sampling zeros closer to the origin.
2. **Dynamic late-triggering through** $\eta(s)$: Similar to §5-6, dynamic STC might be an interesting future direction to prevent switched ZDAs, particularly if sporadic but large IETs are able let attack vectors decay to zero (see 6-5-2).

3. **Time-regularized STC**: By introduction of a (dynamic) strict lower bound $\underline{\kappa} \in \mathbb{N}$ (as used in CETC to prevent Zeno behavior), one could potentially also increase the minimum IET. This solution is not further investigated here and left to future work.

To the best of the author's knowledge, this is the first time a countermeasure based on modifying the IETs by means of an STC strategy has been proposed. In retrospect, we recognize that the suggestion was made in [94] to hold the input signal for multiple sampling periods $m{\cdot}h$, which is in a similar vein as the method proposed here. The main deviation with our proposed method is that they propose a constant $m \in \mathbb{N}$, whilst using STC we propose a time-varying κ_i. Interestingly, their comment on a potential benefit being the lower cost of actuation is indeed the case for an STC policy (see §3-2).

6-4-1 Resilience to switched ZDAs

Theorem 6.5 recognizes that STC are not inherently safe from ZDA, provided that the adversary has disclosure resources of the IETs. Note that the converse of Theorem 6.5 is not necessarily true (see §6-3) and a switched ZDA might still be possible even if the largest IET $\bar{\tau} > \underline{h}$, because this depends (among other things) on the frequency of which this IET occurs. Thus, we aim to find guarantees when switched ZDAs can not be disruptive. To this end, we propose to make use of traffic model abstractions. More information on traffic model abstraction can be found in e.g. [27, 23].

Traffic models

First of all, note that in the absence of load disturbances and noise (i.e. the deterministic case) our proposed STC policy has en equivalent periodic event-triggered control (PETC) policy [32], which we will consider here. For these PETC systems, abstractions to analyze the IETs in terms of finite state automata are available [22]. In particular, the tool ETCetera (see [27]) allows us to construct these traffic model abstractions and from them, extract relevant quantitative metrics.

The (infinite) sequence of inter-event indices $\kappa_1, \kappa_2, \ldots$ is the traffic generated by the PETC. In some cases, after a transient phase, this traffic might be periodic with period $m \in \mathbb{N}$. Therefore, we are interested in the limiting behavior of the IETs of the PETC policy as $i \to \infty$, starting from any initial condition $\boldsymbol{x}_0 \in \mathbb{R}^{n_\mathrm{x}}$. Suppose a cycle of inter-event indices $\vec{\kappa}_1, \ldots, \vec{\kappa}_m$ occurs for sufficiently large $i \geqslant I$ and all initial conditions $\boldsymbol{x}_0 \in \mathbb{R}^{n_\mathrm{x}}$. Let us introduce

$$\vec{\boldsymbol{A}}_\mathrm{a} = (\boldsymbol{A}_{\vec{\kappa}_1} + \boldsymbol{B}_{\vec{\kappa}_1}\boldsymbol{F}_{\vec{\kappa}_1}) \cdots (\boldsymbol{A}_{\vec{\kappa}_m} + \boldsymbol{B}_{\vec{\kappa}_m}\boldsymbol{F}_{\vec{\kappa}_m}), \tag{6-8}$$

with $\boldsymbol{F}_\kappa$ as in Algorithm 1, such that $\boldsymbol{f}_{i+m} = \vec{\boldsymbol{A}}_\mathrm{a}\boldsymbol{f}_i$ holds for all initial conditions $\boldsymbol{f}_0$ and $i \geqslant I$. Note that the sequence of inter-event indices $\vec{\kappa}_1, \ldots, \vec{\kappa}_m$ (provided they exist) can be obtained using ETCetera for a linear PETC system with full-state feedback quadratic triggering condition, as is the case here. Guarantees for when the traffic automaton contains a minimizing cycle are given in [52, Proposition 13]. Note that whilst we are dealing with output feedback and furthermore, the observation error $\tilde{\boldsymbol{x}}_i$ might be very large, an abstraction can still be constructed as the next sampling time is based solely on the estimated state vector $\hat{\boldsymbol{x}}_i$.

It must be noted that constructing such an abstraction can be computationally expensive (see §5-5). There could exist sets of measure zero which hinder convergence of the (semi)-algorithm, or the PETC might not exhibit periodic behavior at all and instead the traffic generated might be chaotic [52]. We elaborate on this in §5-6. However, assuming a limiting cycle of inter-event times has been obtained, we can state the following.

Proposition 6.6 (Resilience to switched ZDAs)**.** *Consider the closed-loop architecture as in* Figure 3-2 *and suppose* $\vec{\boldsymbol{A}}_\mathrm{a}$ *as in* (6-8) *is Schur stable. Then, the system is resilient to a disruptive switched* ZDA*, provided* $\|\boldsymbol{f}_0\|$ *is sufficiently small.*

Proof: Given that i is sufficiently large, the transient due to the non-zero initial condition $\boldsymbol{f}_0 \neq \boldsymbol{0}$ as in Proposition 6.2 can be neglected, and we have $\boldsymbol{f}_i = \boldsymbol{x}_i$ for $i \geqslant I$. Given that $\vec{\boldsymbol{A}}_\mathrm{a}$ is Schur stable we have $\lim_{i\to\infty} \boldsymbol{f}_i \to \boldsymbol{0}$ which implies $\lim_{i\to\infty} \boldsymbol{x}_i \to \boldsymbol{0}$. Thus, the magnitude of the state vector remains finite for all time meaning $\|\boldsymbol{x}_i\| \leqslant \bar{x}$ for all i, for which the size of $\bar{x}$ is dependent on the initial condition $\boldsymbol{f}_0$. If $\|\boldsymbol{f}_0\|$ is sufficiently small then $\|\boldsymbol{x}_i\| \leqslant \bar{x}$ for all i implies $\boldsymbol{\chi}(t) \in \mathbb{X}_\mathrm{s}$, for all t. ■

If the above proposition holds, then the attack vector $\boldsymbol{f}_i$ has finite energy. The results rely on $\|\boldsymbol{f}_0\|$ being approximately equal to zero (such that the attack vector cannot grow excessively during the transient $i < I$), which can be enforced through a sufficiently small threshold η (see Proposition 6.2). As we are assuming the absence of load disturbances and noise, we are free to set $\eta > 0$ as small as necessary (the effect of which is demonstrated in 6-5-2). Similarly, a sufficiently large safe region $\mathbb{X}_\mathrm{s}$ can obtain a similar effect, although it might not be reasonable to assume the safe region to be arbitrarily large in practice.

Alternatively, one can increase the sampling period h such that the discretized plant $\mathcal{P}$ no longer contains unstable sampling zeros. However, choosing such a sampling period is based fundamentally on a worst-case analysis across the state-space of the system [23]. Therefore, employing an STC policy, even with a considerably large triggering parameter σ, can often lead to better performance. This is because STC has the advantage of having 'feedback' in determining the transmission times [53], and thus both faster (and slower) triggering rates are possible when needed.

6-5 Illustrative examples

Consider the stable continuous-time plant $\underset{\sim}{\mathcal{P}}$ with output-feedback given by

$$\underset{\sim}{\boldsymbol{A}} = \begin{bmatrix} -5 & 1 & 0 \\ -8 & 0 & 1 \\ -4 & 0 & 0 \end{bmatrix}, \qquad \underset{\sim}{\boldsymbol{B}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad \boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}. \tag{6-9}$$

Note that the continuous-time system contains no zeros, which means that the relative degree $n_\nu = 3$. We design a linear–quadratic–regulator (LQR) controller $\boldsymbol{K}$ by an emulation design method such that $\underset{\sim}{\boldsymbol{A}} - \underset{\sim}{\boldsymbol{B}}\boldsymbol{K}$ is Hurwitz. By arbitrarily selecting the weighting matrices $\boldsymbol{W}_{\mathrm{x}} = 0.01 \cdot \boldsymbol{I}$ and $\boldsymbol{W}_{\mathrm{u}} = 0.5$ and construct the algebraic Riccati equation (ARE) given by [101]

$$\underset{\sim}{\boldsymbol{A}}^{\mathrm{T}}\boldsymbol{R} + \boldsymbol{R}\underset{\sim}{\boldsymbol{A}} - \boldsymbol{R}\underset{\sim}{\boldsymbol{B}}\boldsymbol{K} = \boldsymbol{W}_{\mathrm{x}}, \qquad \boldsymbol{K} = \boldsymbol{W}_{\mathrm{u}}^{-1}\underset{\sim}{\boldsymbol{B}}^{\mathrm{T}}\boldsymbol{R}, \tag{6-10}$$

which has a unique stabilizing solution $\boldsymbol{R} \succ 0$ given Assumption 1 holds. This is the case for (6-9), and solving (6-10) we obtain $\boldsymbol{K} = [\ -0.015 \quad -0.009 \quad 0.035\ ]$. A sampling period of $h = 0.2$ is chosen (for which Assumption 3 holds), such that

$$\boldsymbol{A} = \begin{bmatrix} 0.28 & 0.12 & 0.01 \\ -1.02 & 0.88 & 0.19 \\ -0.48 & -0.06 & 1.00 \end{bmatrix}, \qquad \boldsymbol{B} = \begin{bmatrix} 0.00 \\ 0.02 \\ 0.20 \end{bmatrix}. \tag{6-11}$$

The STC policy is a *trigger-on-relative-state-error* from (3-23), with $\sigma = 0.32$ and $\bar{\kappa} = 10$. We use the simplified linear matrix inequality (LMI) conditions from [61, Theorem III.4] and find the system is globally exponentially stable (GES) with decay rate $\rho = 2.25$. These are applicable because the full-state feedback controller is static. The decay rate was found by means of a bisection method on the interval $[10^{-1}, 10]$. The seed was set to `45538370` in NCSim. Given that $\boldsymbol{C} \neq \boldsymbol{I}$, we design an observer $\mathcal{O}$ as in Theorem 3.3, which yields $\boldsymbol{L} = [\ 0.37 \quad 0.80 \quad 0.36\ ]^{\mathrm{T}}$. Since the controller $\mathcal{C}$ is static, Assumption 6 is satisfied [2], and the subsystem $\underset{\sim}{\mathcal{G}}$ (see §3-7) is input-to-state stable (ISS) to observation errors $\tilde{\boldsymbol{x}}_i$.

In this example, we consider the objective of regulation and the presence of a strong Byzantine adversary (see Assumption 9). The safe region is defined as $\mathbb{X}_{\mathrm{s}} = \{\boldsymbol{x}_i \in \mathbb{R}^{n_{\mathrm{x}}} \mid \|\boldsymbol{x}_i\|_\infty \leqslant 3\}$. From (3-11) we find that $\bar{\kappa}_{\max} = 8$. Furthermore, from Lemma 6.3 we find $\underline{h} = 1.73$. As $\kappa_{\max} \leqslant \bar{\kappa}_{\max} = 8 < \lfloor \underline{h}/h \rfloor = 9$, by Theorem 6.5 the closed-loop system is susceptible to a disruptive switched ZDA attack.

First, consider a switched ZDA with initial condition $\boldsymbol{x}_0 = \hat{\boldsymbol{x}}_0 = \boldsymbol{0}$ starting at $t_0 = 0$. Using Algorithm 1 we find the maximal controlled invariant subspace $V^\star = \operatorname{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2\}$ with $\boldsymbol{v}_1 = \operatorname{col}(0, 1, 0)$, $\boldsymbol{v}_2 = \operatorname{col}(0, 0, 1)$. The initial attack vector $\boldsymbol{f}_0 = 10^{-9} \cdot \boldsymbol{v}_1 \in V^\star$ and the simulation is run for $T = 10$ time units.

The norm of the resulting state trajectory $\boldsymbol{\chi}(t)$ as well as the state estimate $\hat{\boldsymbol{x}}_i$ can be seen in Figure 6-2. The state leaves the safe region at $t = 8.6$, at which time the output $\boldsymbol{y}_i$ is still equal to zero, meaning the switched ZDA is both stealthy and disruptive.

Whilst the attack appears to be detected at the sudden jump of the output at $k = 47$, this is merely due to numerical artifacts. The simulation quickly exceeds the needed floating-point capacity and loses precision as the state norm grows geometrically. Therefore, (rounding) errors are greatly amplified. Theoretically, with infinite precision, the state norm would become arbitrarily large with the output remaining zero. In the next section, we show that even in the case of imperfect knowledge (practical) switched ZDAs are still possible.

6-5-1 Influence of imperfections

Under nominal system operations, the state vector will most likely not be precisely at the origin. Therefore, here the effects of imperfections on switched ZDAs are investigated. We

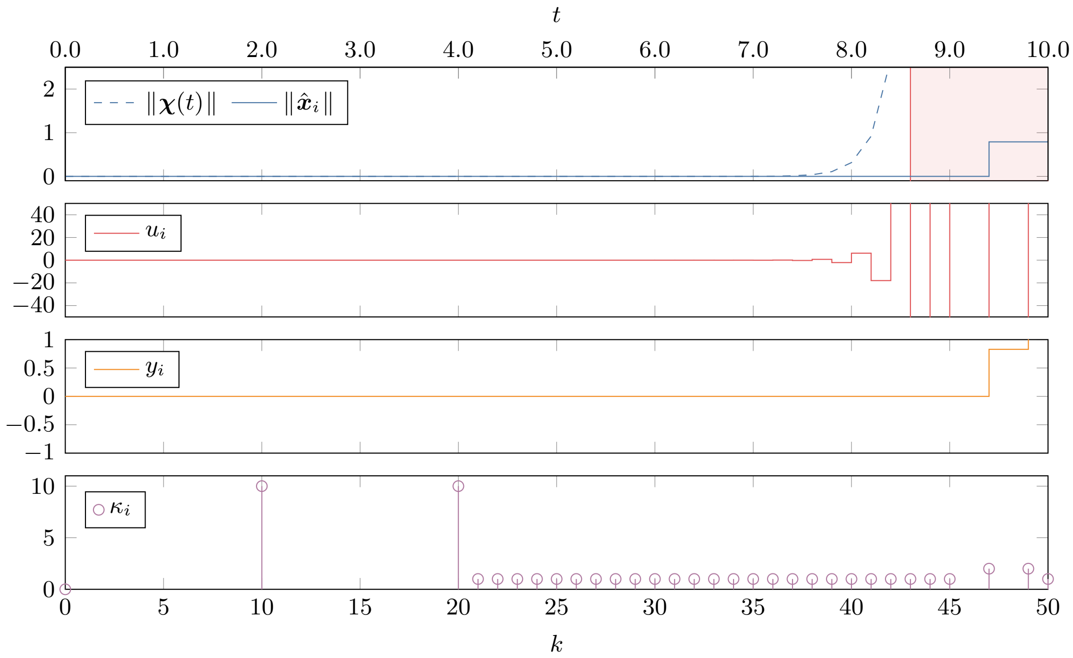


Figure 6-2: Effect of a ZDA on a noise-free system with initial condition $\boldsymbol{x}_0 = \boldsymbol{0}$

demonstrate that a non-zero initial condition as well as load disturbances $\boldsymbol{w}_i$ are not prohibitive for a successful switched ZDA.

In Figure 6-3 the system is initialized at $\boldsymbol{\chi}(0) = \boldsymbol{1}$. Due to the assumption of a prolonged operation time, we furthermore set $\hat{\boldsymbol{x}}_0 = \boldsymbol{\chi}(0)$. The initial attack vector $\boldsymbol{f}_0 = 10^{-9} \cdot \boldsymbol{v}_1 \in V^\star$ and the simulation is run for $T = 20$. The switched ZDA is once again successful as the state trajectory leaves the safe region at $t = 18.6$ whilst the output remains zero, except for the transient at the beginning of the simulation due to the non-zero initial condition $\boldsymbol{\chi}(0)$.

Next, we consider the effect of load disturbances $\boldsymbol{w}_i$ on the detectability of switched ZDA. Similar to 5-5, the matrix $\underset{\sim}{\boldsymbol{E}} = \boldsymbol{I}$, which implies $\boldsymbol{\Sigma}_{\mathrm{w}} \approx 10^{-3} \cdot \boldsymbol{I}$ using (3-3). The seed value was set to 45538370 in NCSim. The results can be seen in Figure 6-4. At $t = 7.0$ the trajectory leaves the safe region. The output $\boldsymbol{y}_i$ is not equal to zero, but it is hardly possible to distinguish the effect of the attack from that of the actual noise [98]. Again, whilst the state trajectory diverges, the state estimate $\hat{\boldsymbol{x}}_i$ remains close to the origin.

The trajectories of $\boldsymbol{\chi}(t)$ for the two scenarios described above are plotted in phase space in Figure 6-5. The nullspace $\ker(\boldsymbol{C})$ of the measurement matrix is shown as the gray vertical plane. Figure 6-5a elegantly illustrates how the adversary 'hides' his attack within this nullspace whilst steadily letting the state trajectory diverge.

6-5-2 Influence of large IETs

As alluded to in §6-4, extending the IETs can move the unstable sampling zeros into the unit circle. Illustrative examples are presented next.

To extend the IETs, the triggering parameter is increased to $\sigma = 1.2$ whilst the upper bound $\bar{\kappa}$ remains the same. With these parameters GES is preserved with a decay rate $\rho = 2.12$

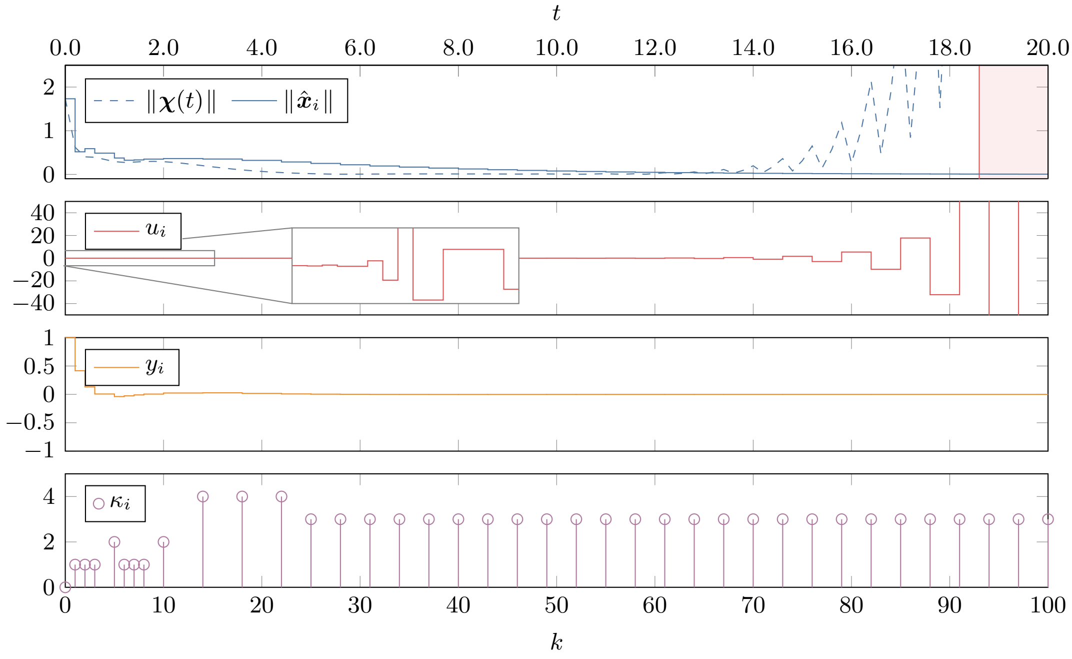


Figure 6-3: Effect of a ZDA on a noise-free system with non-zero initial condition $\boldsymbol{x}_0 = \mathbf{1}$

(note that the dynamics as in (6-11) are stable). The initial attack vector $\boldsymbol{f}_0 = 10^{-6} \cdot \boldsymbol{v}_1 \in V^\star$ and the simulation is run for $T = 20$ time units. From 6-6, it can be seen that the switched ZDA is unsuccessful in being disruptive, and the trajectory remains inside the safe region.

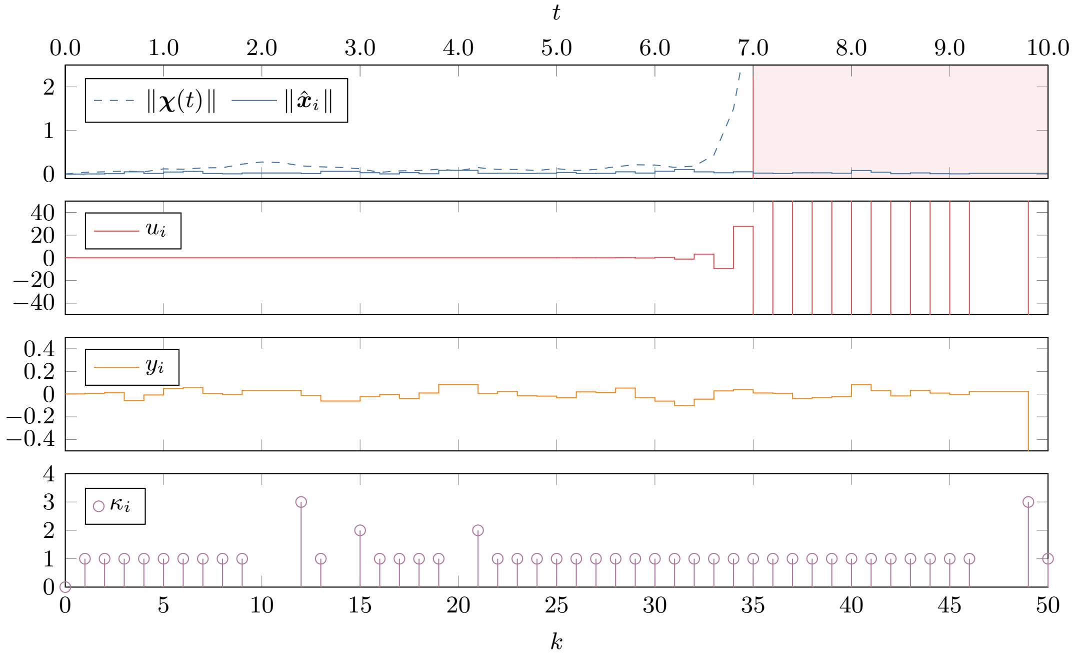
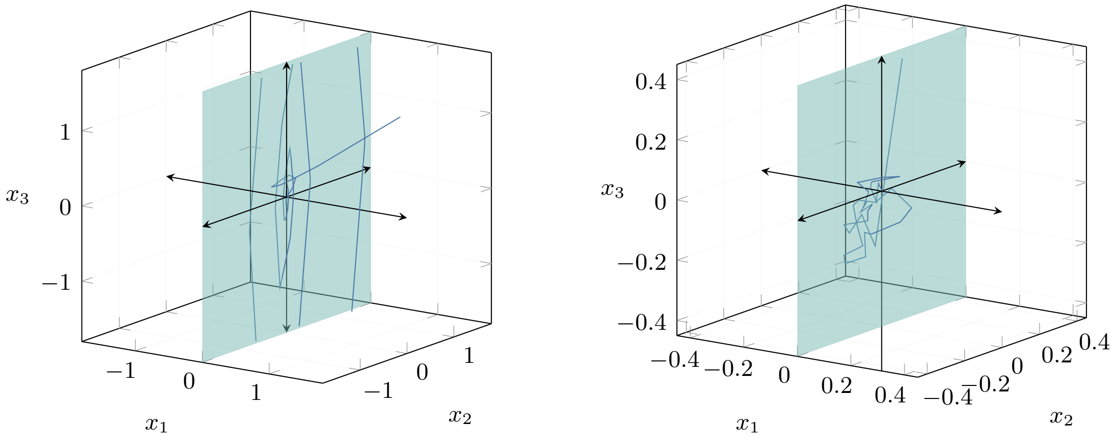


Figure 6-4: Effect of a ZDA on a system with load disturbance ($\boldsymbol{\Sigma}_\mathrm{w} \approx 10^{-3} \cdot \boldsymbol{I}$) and initial condition $\boldsymbol{x}_0 = \mathbf{0}$



(a) Illustration of the effect of a non-zero initial condition on a ZDA

(b) Illustration of the effect of load disturbances on a ZDA

Figure 6-5: Phase space with trajectories and the effect of a ZDA. Note that the trajectories are clipped outside the box $[-1.5, 1.5]^3$.

Interestingly, at the beginning of the simulation when the IETs are still small, the norm of the input initially increases, but as the IETs get larger the sampling zeros become stable and the input decays back to zero.

Using ETCetera we can check whether Proposition 6.6 holds. To this extent, an attempt was made to create a traffic model abstraction. Unfortunately, with a depth of $l = 11$, no limiting cycle had been verified, although a lower bound to the smallest average inter-event index $\vec{\kappa}_{\text{avg}} = 1.6$ was found[4]. This highlights one of the difficulties in working with these finite state abstractions, and further verification with sufficient computational resources is left for future work. Numerically simulating the trajectories of $7.2 \cdot 10^3$ initial conditions evenly spaced on half of the unit sphere, we find that all of them eventually exhibit a periodic cycle of six inter-event indices, suggesting $m = 1$ and $\vec{\kappa}_1 = 6$ (although this is not a formal guarantee). For some initial conditions, these cycles appear only after 21 iterations, meaning $l > 21$ in order to verify these cycles. Note that from Figure 6-1a we find that the SL system as in (3-7) with $\kappa = 6$ has no unstable sampling zeros.

Finally, we would like to demonstrate that in practice guaranteeing (sporadic) large IETs is not sufficient for preventing switched ZDAs, if these IETs are too infrequent. In the following example, we once again take into account load disturbances with covariance matrix $\boldsymbol{\Sigma}_{\text{w}} \approx \boldsymbol{I}$. The triggering parameter is set to $\sigma = 0.5$ whilst the margin parameter $\epsilon = 2 \cdot 10^{\text{-}4}$. The initial attack vector $\boldsymbol{f}_0 = 10^{\text{-}9} \cdot \boldsymbol{v}_1 \in V^\star$ and the simulation is run for $T = 40$.

The simulation results can be seen in Figure 6-7. From the bottom plot, we see that sporadic burst of short IETs, followed by one or two very large IETs. Initially, the norm of the control input $|v(t)|$ seems to sometimes grow, but then quickly decay back to zero. However, at $t = 37.2$ the trajectory does leave the safe region with the output remaining in between the detection bounds. Note the peaks in the state norm $\|\boldsymbol{\chi}(t)\|$, the most notable one at $t = 35.0$, which are not visible in the output.

[4] The algorithm was run on a HP ZBook Studio G3 with Intel Core i7 @ 2.60GHz and 8.00 GB of RAM. The algorithm was interrupted after 23 hours of runtime, and no MACE simulation was found.

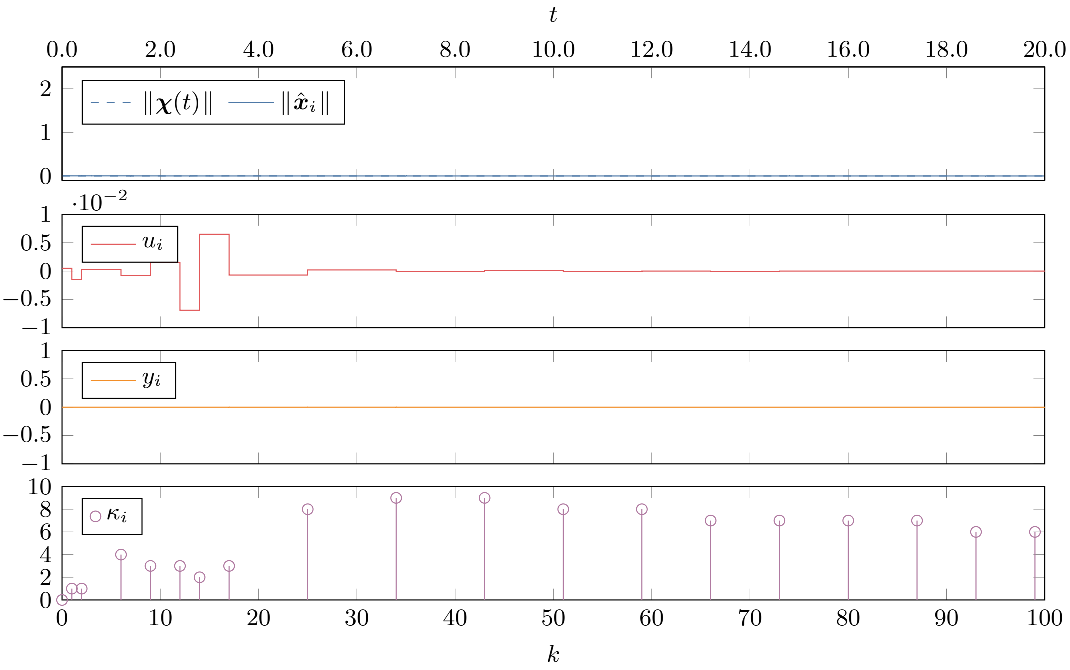


Figure 6-6: Effect of a ZDA on a noise-free system with large IETs

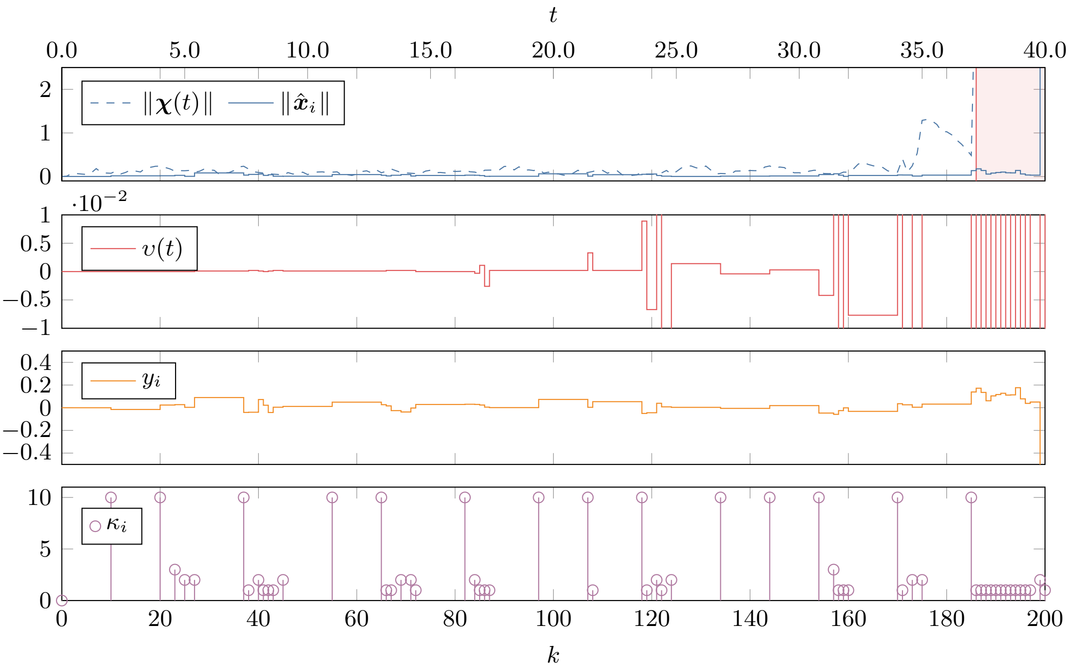


Figure 6-7: Effect of a ZDA on a system with load disturbances and large IETs

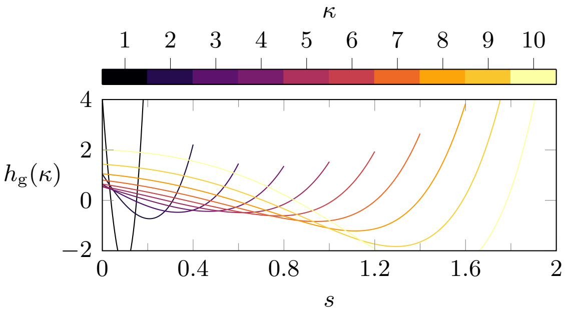


Figure 6-8: Generalized hold function

6-5-3 Qualitative comparison

The previous sections demonstrate that longer IETs are capable of moving unstable sampling zeros into the unit circle. However, more research is needed on how to translate these findings into actual design procedures. In this section, we compare our method with existing countermeasures in the literature. A comparative analysis remains challenging as noted previously. Therefore, here we only conduct a qualitative comparison.

First, let us compare the method of deploying additional measurements described in [123], which is outlined in Algorithm 2. For the system given in (6-11), it is trivial to verify that for any $\boldsymbol{c}^{\mathrm{T}} = \mathrm{col}(c_1, c_2, c_3) \in \mathbb{S}$ with either $c_1 \neq 0$ or $c_2 \neq 0$ the discretized system $\mathcal{P}$ as in (3-1) no longer contains any zeros. With this additional measurement, ZDAs are no longer possible. It remains to be seen whether practically implementing these sensors is both physically possible and economically viable. Finally, the failure of such a redundant sensor would of course imply the system is once again vulnerable to switched ZDAs.

Furthermore, let us consider the method of generalized hold as proposed in [69]. If we desire that the closed-loop system no longer has any zeros, then we can compute (a realization of) $\boldsymbol{A}'$, $\boldsymbol{B}'$ and from there construct the generalized hold function $h_g(s\,;\kappa)$ as in (6-4). These results can be seen in Figure 6-8. Note that the generalized hold mechanism is different for each value $1 \leqslant \kappa \leqslant \bar{\kappa}$. The above solution does prevent switched ZDAs, but requires a device that can generate continuous-time signals, which may not be practical or even feasible [114]. Therefore, [69] also proposes a piecewise constant formulation, at the cost of an additional performance deficit. Finally, usually the digital controller $\mathcal{C}$ has to be reconfigured according to the change to the holding device [69].

Lastly, we briefly mention yet two other existing countermeasures. First, [123] propose either perturbing the system matrix $\boldsymbol{A}$ or encoding the input matrix $\boldsymbol{B}$. However, perturbing $\boldsymbol{A}$ might be undesirable or even physically impossible, whilst encoding $\boldsymbol{B}$ relies on a shared secret between the actuators and the controller (which is even more troublesome considering the threat of a strong Byzantine adversary). Finally, [94] suggest using an additional sampler with rate h/m, $m > 1$. With sufficiently large m, the lifted system has no unstable zeros, but this does require faster sampling which might not be feasible due to hardware limitations. For the plants as in (6-11), and the fact that $(\boldsymbol{A}, \boldsymbol{C})$ is observable according to Assumption 1, it follows that for a dual-rate sampler with $m = 4$ (i.e. a sampling period of 0.05 time units) the lifted system no longer contains unstable zeros [94, Lemma 7].

Chapter 7

Conclusion

In this chapter, we summarize the results obtained and discuss our findings. We have shown that the extra degree of freedom obtained by employing an self-triggered control (STC) policy can aid in both the detection and prevention of replay attacks and switched zero dynamic attack (ZDA), respectively. These results have been provided in light of the same framework, although different assumptions on the capabilities of the attacker are considered.

We would like to stress here that, similar to [9], the augmentation of the inter-event times (IETs) of an STC policy to aid attack detection/prevention stands as a proof-of-concept. Further research and development is needed in order to make this strategy viable in practice, which is something the author aims to pursue. We do believe that this novel idea can be an alternative to other types of watermarking under the right circumstances. Furthermore, the vulnerability of STC systems (and for that matter, switched linear (SL) system or time-varying system) to (switched) ZDAs is something that, to the best of the author's knowledge, has never been shown before.

7-1 Discussion

As discussed in §5-6, the proposed watermarking method is mainly suited for periodic, non-constant reference tracking. Whilst we have shown that by modifying the triggering parameters ϵ and $\bar{\kappa}$ whenever the trajectory is close to the origin does allow replay attacks to be detected, this comes at the cost of significant drift of the trajectories away from the origin, which can be highly undesirable. As such, changing the watermarking scheme would possibly be preferable. We would like to note that our method proposed here in a sense adequately complements the work of [9], be it that they do not consider disturbances and noise which they leave to future work. Since their method applies to tracking constant references, it would be interesting to see if a blend of both strategies can provide better detection capabilities with smaller performance loss.

In addition, since we are in general considering dynamic controllers for periodic reference tracking, the controller state $\boldsymbol{c}[k]$ as in (2-2) can aid the detection of replay attacks even when watermarking is not present, as discussed in Appendix A-4. Thus, the replay attack as considered in 5-5 constitutes a *worst-case* scenario. Note that in practice, however, other phenomena such as false alarms, package drops, and quantization errors might make the start of a replay attack indistinguishable from noise and faults. Still, it could be argued that due to the former, replay attacks pose less of a threat to control systems performing periodic reference tracking, as the adversary has less chance to succeed. Note that static reference tracking controllers (e.g. proportional-only controllers, which are still widely used in industry [35]) do not pose the same difficulty for the adversary.

We provide sufficient conditions making switched ZDAs no longer disruptive in §6-4-1, which relies on increasing the triggering parameter σ such that the IETs are increased. Such an approach might destroy closed-loop stability (if $\underset{\sim}{\boldsymbol{A}}$ is not Hurwitz) before resilience is achieved. Furthermore, even if stability is preserved then the loss in performance by increasing σ might be undesirable or even intolerable. Therefore, we believe that the proposed approach is overly conservative, and more sophisticated methods are possible.

Finally, as discussed in §6-4-1 and demonstrated in §6-5, computing traffic abstraction can be computationally expensive. As such, guaranteeing resilience through to switched ZDAs for Theorem 6.6 might prove prohibitive in practice. Furthermore, as shown in 6-5-2 additive load disturbances can severely alter the behavior of IETs, and lead to vulnerability nonetheless. Stochastic traffic model abstractions as described in [28] might provide a solution and provide an interesting study of future work.

7-2 Future work

We believe STC policies can play an important role in secure control, but recognize work needs to be done. This could include (but is not limited to):

- Considering bounded perturbations $\boldsymbol{\delta}(t)$ as discussed in §2-1, similar to [41]. However, this constitutes a different (non-stochastic) framework.
- Investigate the relation between the proposed online watermarking procedure and the missed detection rate, as well as the suboptimality of the offline triggering procedure when the assumption in Proposition 5.2 does not hold.
- Incorporation of model mismatch between the adversary and the control system as in [98]. Furthermore, switched ZDAs might (not) be possible for periodic event-triggered control (PETC) systems (where the next sampling time is not known in advance). Investigating the effect of timing mismatch is also of interest.
- Investigating the effect of input saturation on both the disruptiveness as well as stealthiness of such an attack. Experiments have shown that such physical limitations uncover stealthy attacks [123]. Therefore, it would also be interesting to investigate the interplay between the effects of extending IETs and input saturation.
- Developing more sophisticated (switched) ZDAs through optimization-based methods, which can incorporate actuation constraints as well as detection constraints. The groundwork for this has been laid in NCSim but is not presented here for conciseness.

- Constructing attacks that are not only stealthy but also achieve disruptiveness quickly (meaning less time for the control system to respond) as discussed in §4-2. One possible method is by means of variable horizon model predictive control (VHMPC) [113], where we minimize the horizon after which the state trajectory leaves the safe region. The groundwork for this has been laid in NCSIM but is not presented here for conciseness.
- Inspired by [67], formulating resilience to (switched) ZDAs in a reach-while-stay framework. A game-theoretical framework might also be very well suited for such analysis.

Finally, as noted by [64] the actual deployment of event-triggered control (ETC) and STC policies in relevant applications is still rather marginal. This, in combination with the lack of security awareness in control systems engineering [102], means there is significant room and need for acceleration. Hopefully, novel applications of STC such as the watermarking scheme proposed here can provide an impulse to more frequent adoption of such techniques. However, added resilience to cyberattacks is almost always a trade-off: it is well-known that security always comes at a cost, which is not only monetary but can also be in terms of availability or loss of performance [39]. As the saying goes in economics, so too does it seem to hold in secure control: there is no such thing as a free lunch.

Appendix A

Supplementary material for Chapter 5

Here, the supplementary materials for Chapter 5 are provided, as well as some of the supplementary material for §3-5.

A-1 Linear impulsive system model

The construction of a (hybrid) linear impulsive (LI) system model is as follows. Consider the augmented state vector $\boldsymbol{\xi}(t) = \mathrm{col}(\boldsymbol{\chi}(t), \vec{\boldsymbol{c}}(t), \boldsymbol{y}_i, \boldsymbol{u}_i)$ for $t \in [t_i, t_{i+1})$ and $\vec{\boldsymbol{c}}(t) = \boldsymbol{c}[k]$ for $t \in h{\cdot}[k, k{+}1)$, which is a right-continuous[1]signal. First of all, note that $\boldsymbol{\xi} : \mathbb{R}_{\geqslant 0} \to \mathbb{R}^{n_\mathrm{x}+n_\mathrm{c}+n_\mathrm{y}+n_\mathrm{u}}$ as we have that

$$\bigcup_{i=0}^{\infty} [t_i + t_{i+1}) = \mathbb{R}_{\geqslant 0} \tag{A-1}$$

by the definition of Γ from (3-12) (i.e. guaranteed Zeno-freeness). Furthermore, since each event time t_i is an integer multiple of the sampling period $h \in \mathbb{R}_{>0}$ the derivative of $\boldsymbol{\xi}(t)$ over a sampling period is well-defined. The LI system model can then be written as

$$\begin{bmatrix} \dot{\boldsymbol{\xi}}(t) \\ \dot{s}(t) \end{bmatrix} = \begin{bmatrix} \bar{\underset{\sim}{\boldsymbol{A}}}\boldsymbol{\xi}(t) + \bar{\underset{\sim}{\boldsymbol{B}}}\boldsymbol{\delta}(t) \\ 1 \end{bmatrix}, \qquad s(t) \in [0, h), \tag{A-2a}$$

$$\begin{bmatrix} \boldsymbol{\xi}(t^+) \\ s(t^+) \end{bmatrix} = \begin{cases} \begin{bmatrix} \bar{\boldsymbol{J}}_0\boldsymbol{\xi}(t) \\ 0 \end{bmatrix} & \boldsymbol{\xi}(t)^\mathrm{T}\bar{\boldsymbol{Q}}\boldsymbol{\xi}(t) \leqslant 0 \\ \begin{bmatrix} \bar{\boldsymbol{J}}_1\boldsymbol{\xi}(t) \\ 0 \end{bmatrix} & \boldsymbol{\xi}(t)^\mathrm{T}\bar{\boldsymbol{Q}}\boldsymbol{\xi}(t) > 0 \end{cases}, \qquad s(t) = h, \tag{A-2b}$$

[1] A right-continuous signal $\vec{\boldsymbol{c}}(t)$ is a piecewise continuous signal where the limit (from the right) $\lim_{t \to a^+} \vec{\boldsymbol{c}}(t)$ exist for all discontinuities a.

where the flow matrices $\bar{\underset{\sim}{\boldsymbol{A}}}$, $\bar{\underset{\sim}{\boldsymbol{B}}}$ are given by

$$\bar{\underset{\sim}{\boldsymbol{A}}} = \begin{bmatrix} \underset{\sim}{\boldsymbol{A}} & \mathbf{0} & \mathbf{0} & \underset{\sim}{\boldsymbol{B}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \qquad \bar{\underset{\sim}{\boldsymbol{B}}} = \begin{bmatrix} \boldsymbol{E} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{A-3}$$

and jump matrices $\boldsymbol{J}_0$, $\boldsymbol{J}_1$ are given by

$$\bar{\boldsymbol{J}}_0 = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{A}_{\mathrm{c}} & -\boldsymbol{B}_{\mathrm{c}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} \end{bmatrix}, \qquad \bar{\boldsymbol{J}}_1 = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\boldsymbol{B}_{\mathrm{c}}\boldsymbol{C} & \boldsymbol{A}_{\mathrm{c}} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{C} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\boldsymbol{D}_{\mathrm{c}}\boldsymbol{C} & \boldsymbol{C}_{\mathrm{c}} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{A-4}$$

A-2 Offline static Kalman filter design

A modified design of a standard χ^2 detector $\mathcal{D}$ (see Figure 3-2) is given as

$$\mathcal{D} \quad : \quad \begin{aligned} \boldsymbol{d}_i &= \boldsymbol{A}_{\kappa_i}\boldsymbol{d}_{i-1} + \boldsymbol{B}_{\kappa_i}\boldsymbol{u}_{i-1} + \boldsymbol{H}_{\kappa_i}\boldsymbol{z}_i, && \text{(A-5a)} \\ \boldsymbol{z}_i &= \boldsymbol{y}_i - \boldsymbol{C}\boldsymbol{d}_i, && \text{(A-5b)} \\ g_i &= \boldsymbol{z}_i^{\mathrm{T}}(\boldsymbol{C}\vec{\boldsymbol{\Sigma}}_{x,\kappa_i}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{\Sigma}_v)^{\text{-}1}\boldsymbol{z}_i. && \text{(A-5c)} \end{aligned}$$

Note that the detector as (A-5) runs an internal state estimator with estimated state $\boldsymbol{d}_i$. The switched observer gain $\boldsymbol{H}_\kappa$ is given by

$$\boldsymbol{H}_\kappa = \boldsymbol{A}_\kappa\vec{\boldsymbol{\Sigma}}_{\mathrm{x},\kappa}\boldsymbol{C}^{\mathrm{T}}(\boldsymbol{\Sigma}_v + \boldsymbol{C}\vec{\boldsymbol{\Sigma}}_{\mathrm{x},\kappa}\boldsymbol{C}^{\mathrm{T}})^{\text{-}1}. \tag{A-6}$$

The 'steady-state' covariance matrix $\vec{\boldsymbol{\Sigma}}_{x,\kappa}$ of the state can be found as the solution to the discrete algebraic Riccati equation (DARE) given by [92]

$$\boldsymbol{A}_\kappa\vec{\boldsymbol{\Sigma}}_{\mathrm{x},\kappa}\boldsymbol{A}_\kappa^{\mathrm{T}} - \vec{\boldsymbol{\Sigma}}_{\mathrm{x},\kappa} + \boldsymbol{\Sigma}_w = \boldsymbol{H}_\kappa(\boldsymbol{C}\vec{\boldsymbol{\Sigma}}_{\mathrm{x},\kappa}\boldsymbol{A}_\kappa^{\mathrm{T}}), \tag{A-7}$$

which can be solved for offline for all $1 \leqslant \kappa \leqslant \bar{\kappa}$. Note that since we are dealing with a switched linear (SL) system a stationary distribution might not necessarily exist [71]. Therefore, the covariance matrix as defined above is only an approximation, but from our results on numerical simulations, it appears to be a sufficiently good one. This might be in part due to the fact that for a constant κ, convergence happens exponentially fast and usually occurs in just a few steps [131, 56].

A-3 Sketches of several proofs

In this section, some (non-rigorous) sketches are provided for proofs currently absent in the literature. We believe these sketches can be converted into formal proffs, given the needed revisions, which is left to future work.

A-3-1 MSS of PETC policies with quadratic triggering conditions

We start by generalizing the triggering conditions, where we notice $\|\boldsymbol{x}_i - \boldsymbol{\chi}(t)\| \geqslant \|\boldsymbol{x}_i - \boldsymbol{\chi}(t)\|_\infty$ and thus by [50, Collary 3] we have that an extension to the 2-norm is straightforward. As such, a periodic event-triggered control (PETC) policy will trigger no later than the one considered in [27] given the quadratic triggering condition $\|\boldsymbol{x}_i - \boldsymbol{\chi}(t)\| > \epsilon$. Note that by [21, Theorem 2], we can choose ϵ arbitrarily large given that the architecture we consider here is a heterogeneous hybrid system, which can be deduced from A-1. Thus, every other quadratic triggering condition with matrix $\boldsymbol{Q}$, such as *trigger-on-relative-error* $\|\boldsymbol{x}_i - \boldsymbol{\chi}(t)\| > \sigma\|\boldsymbol{\chi}(t)\|$, will trigger no later than the triggering condition $\|\boldsymbol{x}_i - \boldsymbol{\chi}(t)\| > \epsilon$ provided ϵ is sufficiently large. From this, it then follows that PETC systems with arbitrary quadratic triggering conditions are stable. As discussed in §5-3 this could be extended to self-triggered control (STC) by means of a hierarchical structure, employing a PETC at the sensor side.

A-3-2 ISS of STC policies with dynamic controllers

The proof follows almost similar reasoning as the one proposed in [3]. Note that we can show that the hybrid system formulation (A-2) coincides with the one proposed in [50, 21]. As such, the hybrid system satisfies the relevant assumption referenced in [21, Lemma 1] and thus [21, Theorem 2] dictates that globally exponentially stable (GES), which can be checked by the linear matrix inequality (LMI) conditions adapted from [61], implies input-to-state stable (ISS) with respect to bounded perturbations $\boldsymbol{\delta}(t)$. If we chose $\underset{\sim}{\boldsymbol{E}} = \underset{\sim}{\boldsymbol{B}}$ then the observation errors belong to the bounded perturbations, and thus is follows that a STC policy with a quadratic triggering condition and dynamic controller $\mathcal{C}$ is ISS with respect to observation error $\tilde{\boldsymbol{x}}$.

A-4 Influence of the controller state

We demonstrate how the use of a dynamic controller can aid in the detection of replay attacks even when no watermarking is present. This boils down to the fact that even though the replayed measurement $\boldsymbol{y}_i = \boldsymbol{y}_{i-\Delta i}$ and the next inter-sampling time $t_{i+1} = t_{i+1-\Delta i}$ are identical, the control input might not be exactly identical, i.e. $\boldsymbol{u}_i \neq \boldsymbol{u}_{i-\Delta i}$. This is due to the dynamic controller state $\boldsymbol{c}[k]$ differing between the time instances, and thus an alarm is raised.

To illustrate, we considered the same dynamics and parameter value as in §5-5, only now, the adversary initiates a replay attack with a delay of five cycles meaning $\Delta T \approx 30$, such that

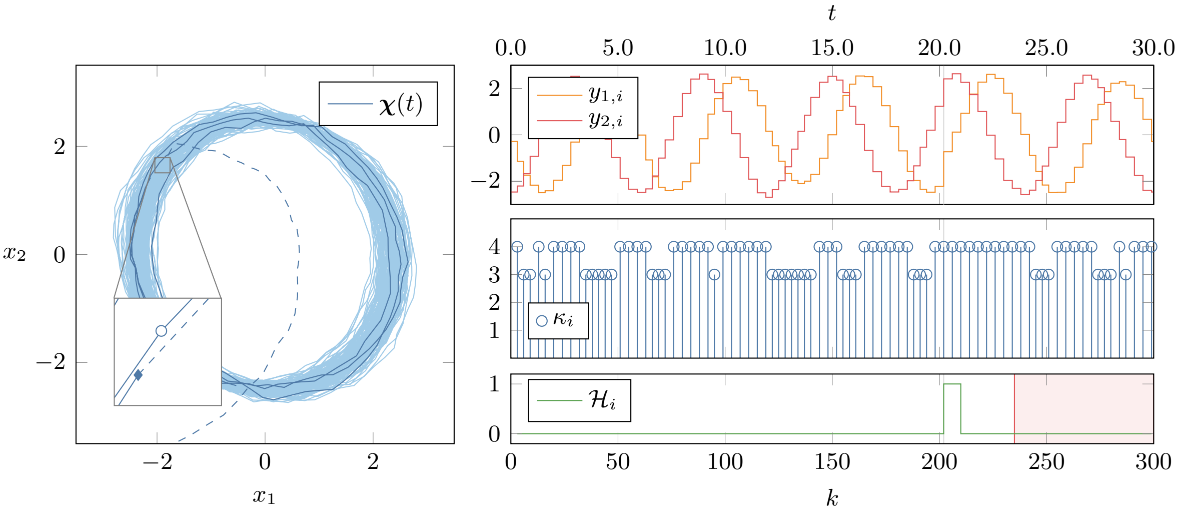


Figure A-1: Effect of the dynamic controller state on the detection of replay attacks

$5 \cdot T_\mathrm{r} \approx \sum_{\ell=0}^{\Delta i-1} |t_{i-\ell} - t_{i-(\ell+1)}|$. Without the presence of watermarking, we find $\Delta i = 84$. The resulting trajectory can be seen in Figure A-1. Since the seed is identical, the first replayed value is received similarly at $k = 202$.

As can be seen in Figure A-1, two alarms are raised at $k = 202, 206$. However, from there on no further alarms are raised, as the controller state $\boldsymbol{c}[k]$ converges to $\boldsymbol{c}[k - 300]$ such that the requested control inputs become identical once more. Thus, whilst an initial alarm is raised, no more alarms are raised thereon after, even once the trajectory leaves the safe region after $t = 23.5$. As shown in §5-5 a different choice of Δi by the adversary might have made it such that no alarms are raised at all. Therefore, without watermarking the system is still susceptible to replay attacks.

Finally, we would like to raise the point that infrequent alarms, only at attack initiation, might not be enough, as this reasonably can be mistaken for a false alarm. This is demonstrated in Figure A-2, where we also show our proposed method for comparison. Due to the sporadic false alarms, a replay attack might not be detected, as only the last alarm corresponds to an actual attack. For our proposed method, the difference between a false alarm and a replay attack is clear.

A-5 Different early triggering procedures in steady state

Here, we illustrate the effect of different early triggering procedures in a steady state. First, a discrete uniform distribution is used as in Proposition 5.2. However, since the trajectory is close to the origin $\|\boldsymbol{CBu}_i\| \ll \|\boldsymbol{C}(\boldsymbol{Ax}_i + \boldsymbol{E\Sigma}_\mathrm{w}) + \boldsymbol{\Sigma}_\mathrm{v}\|$ does not hold. From Figure A-3, we can see no alarm is raised and the replay attack is not detected.

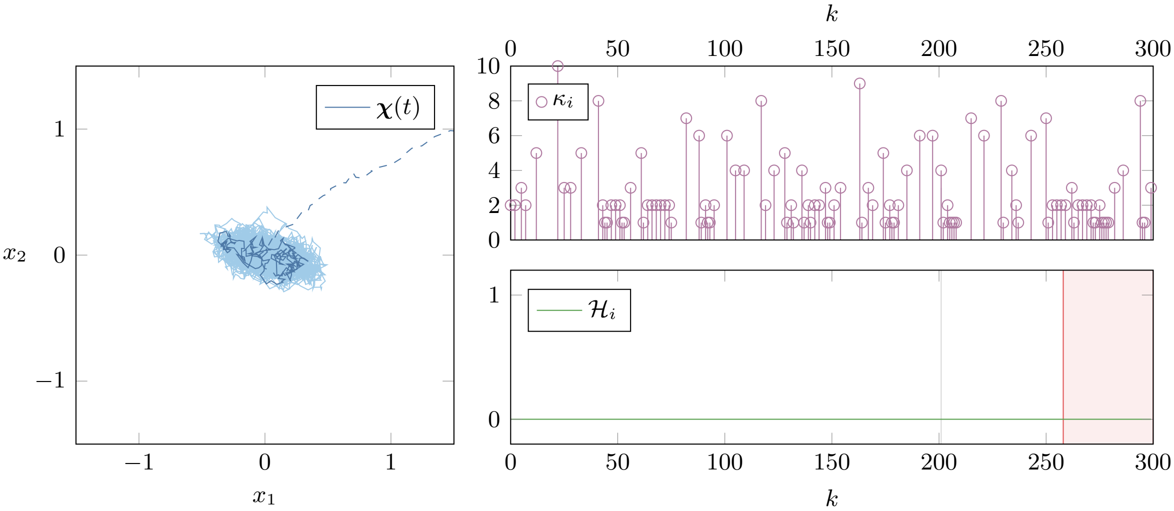


Figure A-3: Effect of a replay attack during steady state regulation with the proposed method with $\sigma = 0.32$, $\epsilon = 0$ and $\bar{\kappa} = 10$ offline procedure

Finally, as observed in §5-6 and can be seen in 5-10, the optimal online early triggering strategy appears to be similar to a distribution where $\kappa = 1$ and $\bar{\kappa}_i$ appear with equal probability. This can also be seen in 5-2a. In Figure A-4 the implementation of this strategy can be seen. The inter-event time (IET) do indeed seem to share similarities with those in 5-10. Whilst there is a single alarm raised at $k = 259$ before the trajectory leaves the safe region at $t = 27.1$, as discussed in Appendix A-4 this alarm becomes hard to distinguish from a false alarm. Thus, the performance of the watermarking strategy is still not satisfactory.

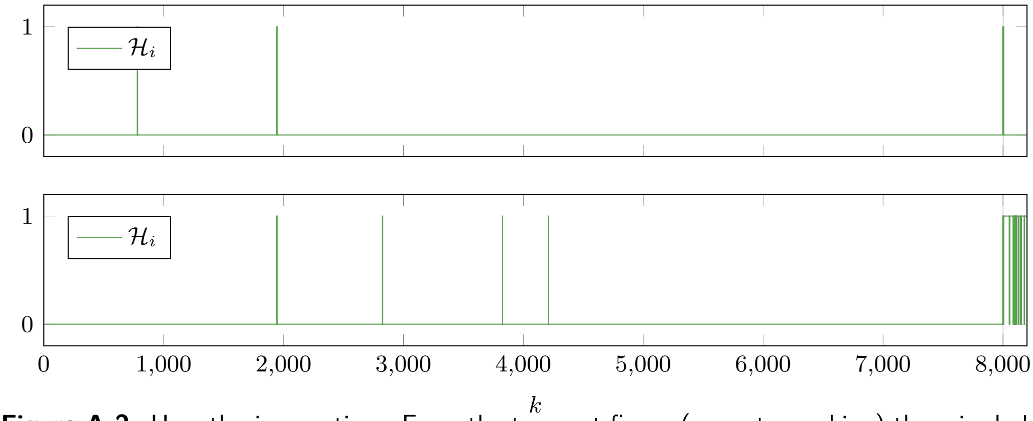


Figure A-2: Hypothesis over time. From the topmost figure (no watermarking) the raised alarm at the initialization of the attack can be mistaken for a false alarm. From the bottom figure (proposed method) the attack is distinguishable.

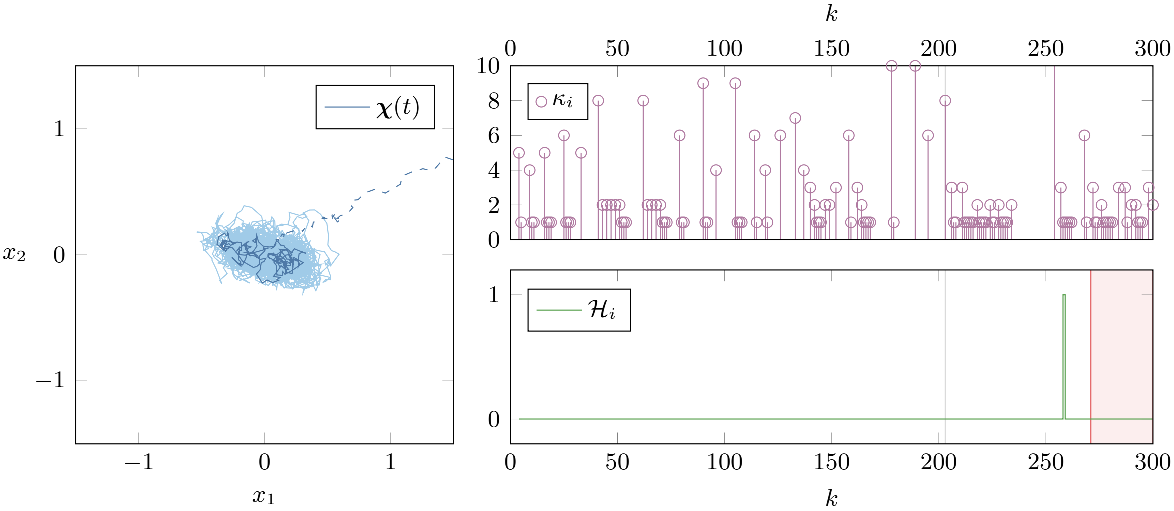


Figure A-4: Effect of a replay attack during steady state regulation with the proposed method with $\sigma = 0.32$, $\epsilon = 0$, $\bar{\kappa} = 10$ and maximal difference

Appendix B

Supplementary material for Chapter 6

B-1 Extension of ZDAa on MIMO systems

Multiple-input and multiple-outputs (MIMOs) systems are just as vulnerable, if not more, to (switched) zero dynamic attacks (ZDAs). Some relevant extensions to such systems are provided here.

For MIMO system the condition of (6-2) not being invertible is slightly altered, and one defines a *transmission zero* as a value $s \in \mathbb{C}$ for which the matrix $\boldsymbol{R}(s)$ loses rank, which is a generalization of the former.

B-2 Relative degree for MIMO systems

The *relative degree* n_ν of a continuous-time plant $\underset{\sim}{\mathcal{P}}$ with an equal number of inputs and outputs is formally defined for MIMO systems as follows.

> **Definition B.1** (Relative degree, adapted from [141, Definition 1])**.** *Consider a continuous-time plant* $\underset{\sim}{\mathcal{P}}$ *as in* (2-1) *and suppose* $n_{\mathrm{u}} = n_{\mathrm{y}}$*. Let* $1 \leqslant \ell \leqslant n_{\mathrm{y}}$ *denote the* ℓ*-th row of the measurement matrix* $\boldsymbol{C} = \mathrm{col}(\boldsymbol{c}_1^{\mathrm{T}}, \dots, \boldsymbol{c}_\ell^{\mathrm{T}}, \dots, \boldsymbol{c}_{n_{\mathrm{y}}}^{\mathrm{T}})$*. Then,* r_ℓ *is said to be the* ℓ*-th relative degree of* $\underset{\sim}{\mathcal{P}}$ *if* $\boldsymbol{c}_\ell^{\mathrm{T}} \underset{\sim}{\boldsymbol{A}}^r \underset{\sim}{\boldsymbol{B}} = \boldsymbol{0}$ *for all* $0 \leqslant r \leqslant r_\ell - 2$ *and* $\boldsymbol{c}_\ell^{\mathrm{T}} \underset{\sim}{\boldsymbol{A}}^{r_\ell - 1} \underset{\sim}{\boldsymbol{B}} \neq \boldsymbol{0}$*. We call* $\boldsymbol{r} = \mathrm{col}(r_1, \dots, r_{n_y})$ *the* relative degree *of* $\underset{\sim}{\mathcal{P}}$*.*

B-3 Masking attacks

Apart from ZDAs MIMO systems face yet another threat, namely *masking attacks*. Whenever $n_{\mathrm{u}} > n_{\mathrm{y}}$ the system $\underset{\sim}{\mathcal{P}}$ is always conducive to a stealthy yet disruptive attack, as one input can mask the effect of one input by another one [94]. Note that input redundancy, somewhat counter-intuitively, does not imply that there are more control inputs than measurement signals, as there exist systems with $n_\mathrm{u} = n_\mathrm{y}$ who are input redundant [73] (and thus susceptible to masking attacks).

B-4 Algorithm for obtaining the Byrnes-Isisdori normal form

To the author's best knowledge, no algorithmic implementation is readily available for constructing the Byrnes-Isisdori normal form in widely used scripting languages. Below, an Python implementation is provided based on [141, Algorithm 1].

```python
import numpy as np
import numpy.typing as npt
import scipy as sp
import control as ct
from itertools import chain

def get_byrnes_isidori_normal_form(sys: ct.StateSpace) -> ct.StateSpace:
    # FROM: "On the relative degree and normal forms of linear systems by output transformation with applications to tracking"  # nopep8

    def get_relative_degrees(sys: ct.StateSpace) -> npt.NDArray[np.dtype[int]]:
        A, B, C, D, n_x, n_u, n_y = sys.A, sys.B, sys.C, sys.D, sys.nstates, sys.ninputs, sys.noutputs
        if not np.all(D == 0):
            raise ValueError(f"Feedforward matrix D must be a zero matrix, recieved {D}")
        rel_degrees = np.zeros(n_y)
        for output_row in range(rel_degrees.size):
            for k in range(1, n_x + 1):
                if np.any(C[output_row, :] @ np.linalg.matrix_power(A, k - 1) @ B) != 0:
                    rel_degrees[output_row] = k
                    break
        return rel_degrees.astype(int)

    A, B, C, D, n_x, n_u, n_y = sys.A, sys.B, sys.C, sys.D, sys.nstates, sys.ninputs, sys.noutputs
    if n_u != n_y:
        raise ValueError(f"This algorithm assumes the same number of inputs as outputs, found n_u = {n_u}, n_y = {n_y}")
    rel_degs_list = get_relative_degrees(sys)
    rel_deg = np.sum(rel_degs_list)
    Gamma = np.stack([C[output_row, :] @ np.linalg.matrix_power(A, rel_degs_list[output_row] - 1) @ B for output_row in range(n_y)], axis=0)
    try:
        Gamma_inv = np.linalg.inv(Gamma)
    except np.linalg.LinAlgError:
        raise ValueError(f"The provided system (A, B, C) has irregular relative degree meaning inv(Gamma) does not exist")
    Theta = B @ Gamma_inv
    Q = np.concatenate([np.stack([np.linalg.matrix_power(A, k) @ Theta[:, output_row] for k in range(rel_degs_list[output_row])], axis=1) for output_row in range(n_y)], axis=1)
    O = np.concatenate([np.stack([C[output_row, :] @ np.linalg.matrix_power(A, k) for k in range(rel_degs_list[output_row])], axis=0) for output_row in range(n_y)], axis=0)
    O_O = sp.linalg.null_space(O)
    S_inv = np.linalg.inv(O @ Q)
    N = np.linalg.inv(O_O.T @ O_O) @ O_O.T @ (np.eye(n_x) - Q @ S_inv @ O)
    A_f = N @ A @ Q @ S_inv
    A_00 = N @ A @ O_0
    rel_deg_cum = np.insert(np.cumsum(rel_degs_list), 0, 0)
    a_0i = [[A_f[:, k] for k in range(rel_deg_cum[output_row], rel_deg_cum[output_row + 1])] for output_row in range(n_y)]
    S_0i = [row_roll(np.tile(np.stack(a_0i[output_row], axis=1),(rel_degs_list[output_row], 1)),
                         np.repeat(-np.arange(0, rel_degs_list[output_row]), A_f.shape[0]), axis=1, fill=0) for output_row in range(n_y)]
    H_0i = [np.concatenate([np.linalg.matrix_power(A_00, k) for k in range(rel_degs_list[output_row])], axis=1) @ S_0i[output_row] for output_row in range(n_y)]
    H = np.concatenate(H_0i, axis=1)
    T = np.concatenate((N - H @ O, O), axis=0)
    T_inv = np.linalg.inv(T)
    Alpha_i = [C[output_row, :] @ np.linalg.matrix_power(A, rel_degs_list[output_row]) @ np.concatenate((O_0, O_0 @ H + Q @ S_inv), axis=1) for output_row in range(n_y)]
    alpha_i0 = [Alpha_i[output_row][0:(n_x - rel_deg)] for output_row in range(n_y)]
    alpha_i = [Alpha_i[output_row][(n_x - rel_deg):] for output_row in range(n_y)]
    A_tilde, B_tilde, C_tilde = T @ A @ T_inv, T @ B, C @ T_inv
    sys_normal_form = ct.ss(A_tilde, B_tilde, C_tilde, D, dt=sys.dt)
    sys_normal_form.set_states(tuple(chain.from_iterable([[f"eta[{l}]"] for l in range(n_x - rel_deg)] + [[f"xi[{output_row},{k}]" for k in range(rel_degs_list[output_row])] for output_row in range(n_y)])))
    return sys_normal_form
```

B-5 Algorithm for deploying additional measurements

Algorithm 2 Deploy additional sensors, adapted from [123, Algorithm 1]

```
Requires: $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\mathbb{S}$, $\boldsymbol{F}$
Returns: $\boldsymbol{C}'$
$\boldsymbol{V} \leftarrow [\ \boldsymbol{v}_1 \ \cdots \ \boldsymbol{v}_{n_\mathrm{k}}\ ]$ s.t. $\mathrm{span}\{\boldsymbol{v}_1, \dots, \boldsymbol{v}_{n_\mathrm{k}}\} = \ker(\boldsymbol{C})$
$\boldsymbol{V}^+ \leftarrow \boldsymbol{0}$
while $\mathrm{range}(\boldsymbol{V}^+) \neq \mathrm{range}(\boldsymbol{V})$ do
    for $\ell, \boldsymbol{c}^\mathrm{T} \in \mathbb{S}$ do
        $\boldsymbol{W} \leftarrow [\ \boldsymbol{w}_1 \ \cdots \ \boldsymbol{w}_{1-n_\mathrm{x}}\ ]$ s.t. $\mathrm{span}\{\boldsymbol{w}_1, \dots, \boldsymbol{w}_{1-n_\mathrm{x}}\} = \ker(\boldsymbol{c}^\mathrm{T})$
        $\boldsymbol{\Pi}_{WV} \leftarrow \boldsymbol{V}(\boldsymbol{V}^\mathrm{T}\boldsymbol{V})^{\text{-}1}\boldsymbol{V}^\mathrm{T}\boldsymbol{W}(\boldsymbol{W}^\mathrm{T}\boldsymbol{W})^{\text{-}1}\boldsymbol{W}^\mathrm{T}$
        $\boldsymbol{\lambda}, \boldsymbol{E} \leftarrow$ Eigenvalues and eigenvectors of $\boldsymbol{\Pi}_{WV}$
        $\boldsymbol{V}^+ \leftarrow [\ \boldsymbol{e}_1 \ \cdots \ \boldsymbol{e}_{n_\mathrm{v}}\ ]$ s.t. $\boldsymbol{e}_i \in \mathrm{range}(\boldsymbol{E})$ and $\lambda_i = 1$
        if $\dim(\mathrm{range}(\boldsymbol{V}^+)) = \dim(\mathrm{range}(\boldsymbol{V}))$ then
            Remove $\boldsymbol{c}^\mathrm{T}$ from $\mathbb{S}$
            $\ell \leftarrow \ell - 1$
        else
            $V^\star \leftarrow (\boldsymbol{A} + \boldsymbol{BF})$ – invariant $\subseteq \mathrm{range}(\boldsymbol{V}^+)$ using Algorithm 1
            $\boldsymbol{V} \leftarrow [\ \boldsymbol{v}_1 \ \cdots \ \boldsymbol{v}_{n_\mathrm{k}}\ ]$ s.t. $\mathrm{span}\{\boldsymbol{v}_1, \dots, \boldsymbol{v}_{n_\mathrm{k}}\} = V^\star$
$\boldsymbol{C}' \leftarrow \mathrm{col}(\boldsymbol{C}, \boldsymbol{c}_1^\mathrm{T}, \dots, \boldsymbol{c}_\ell^\mathrm{T})$
```

B-6 Additional dynamics

Consider the stable continuous-time dynamics $\underset{\sim}{\mathcal{P}}$ given by

$$\underset{\sim}{\boldsymbol{A}} = \begin{bmatrix} 13.0 & -16.2 & 3.3 \\ 6.1 & -2.1 & -4.0 \\ -7.6 & 24.7 & -17.2 \end{bmatrix}, \quad \underset{\sim}{\boldsymbol{B}} = \begin{bmatrix} 3.9 \\ -2.3 \\ 5.3 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 2.7 & -2.84 & 0.74 \end{bmatrix}, \tag{B-1}$$

with $\boldsymbol{D} = 1$. The poles of the system are $-0.51, -2.23, -3.57$ and the (intrinsic) zeros are $-11.04 \pm 6.23j, -5.07$.

Bibliography

[1] Sanad Al-Areqi, Daniel Görges, and Steven Liu. Event-based networked control and scheduling codesign with guaranteed performance. *Automatica*, 57:128–134, July 2015.

[2] J. Almeida, C. Silvestre, and A. M. Pascoal. Observer based self-triggered control of linear plants with unknown disturbances. In *2012 American Control Conference (ACC)*, pages 5688–5693, Montreal, QC, June 2012. IEEE.

[3] Joao Almeida, Carlos Silvestre, and Antonio M. Pascoal. Observer based self-triggered control of an acyclic interconnection of linear plants. In *IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 7553–7558, Maui, HI, USA, December 2012. IEEE.

[4] João Almeida, Carlos Silvestre, and Antonio Pascoal. Self-triggered output feedback control of linear plants. In *Proceedings of the 2011 American Control Conference*, June 2011.

[5] João Almeida, Carlos Silvestre, and António Pascoal. Self-triggered observer based control of linear plants. *IFAC Proceedings Volumes*, 44(1):10074–10079, January 2011.

[6] João Almeida, Carlos Silvestre, and António M. Pascoal. Self-Triggered Output Feedback Control of Linear Plants in the Presence of Unknown Disturbances. *IEEE Transactions on Automatic Control*, 59(11):3040–3045, November 2014. Conference Name: IEEE Transactions on Automatic Control.

[7] Robert Annessi, Joachim Fabini, Felix Iglesias, and Tanja Zseby. Encryption is Futile: Delay Attacks on High-Precision Clock Synchronization, November 2018. arXiv:1811.08569 [cs].

[8] Duarte J. Antunes and M. H. Balaghi I. Consistent Event-Triggered Control for Discrete-Time Linear Systems With Partial State Information. *IEEE Control Systems Letters*, 4(1):181–186, January 2020. Conference Name: IEEE Control Systems Letters.

[9] Angelo Barboni, Ahmad W. Al-Dabbagh, and Thomas Parisini. An Event-Triggered Watermarking Strategy for Detection of Replay Attacks. *IFAC-PapersOnLine*, 55(6):317–322, January 2022.

[10] G. Basile and G. Marro. *Controlled and conditioned invariant subspaces in linear system theory*, volume 3. Journal of Optimization Theory and Applications, May 1969.

[11] Dmitry Belov and Ronald Armstrong. Distributions of the Kullback-Leibler divergence with applications. *The British journal of mathematical and statistical psychology*, 64:291–309, May 2011.

[12] Lennart Blanken. *Learning and repetitive control for complex systems: with application to large-format printers.* Phd Thesis, Technische Universiteit Eindhoven, Eindhoven, May 2019. ISBN: 9789038647579.

[13] D. P. Borgers, V. S. Dolk, G. E. Dullerud, A. R. Teel, and W. P. M. H. Heemels. Time-Regularized and Periodic Event-Triggered Control for Linear Systems. In Sophie Tarbouriech, Antoine Girard, and Laurentiu Hetel, editors, *Control Subject to Computational and Communication Constraints: Current Challenges*, Lecture Notes in Control and Information Sciences, pages 121–149. Springer International Publishing, Cham, 2018.

[14] D. P. Borgers and W. P. M. H. Heemels. Event-separation properties of event-triggered control systems. *IEEE Transactions on Automatic Control*, 59(10):2644–2656, October 2014. Conference Name: IEEE Transactions on Automatic Control.

[15] Alvaro A. Cardenas, Saurabh Amin, and Shankar Sastry. Secure Control: Towards Survivable Cyber-Physical Systems. In *2008 The 28th International Conference on Distributed Computing Systems Workshops*, pages 495–500, June 2008. ISSN: 2332-5666.

[16] Ahmet Cetinkaya, Hideaki Ishii, and Tomohisa Hayakawa. Event-triggered control over unreliable networks subject to jamming attacks. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 4818–4823, December 2015.

[17] Tongwen Chen and Bruce Allen Francis. *Optimal Sampled-Data Control Systems.* Springer, London, 1995.

[18] Young-Geun Choi, Johan Lim, Anindya Roy, and Junyong Park. Fixed support positive-definite modification of covariance matrix estimators via linear shrinkage. *Journal of Multivariate Analysis*, 171:234–249, May 2019.

[19] Angelo Corallo, Mariangela Lazoi, Marianna Lezzi, and Angela Luperto. Cybersecurity awareness in the context of the Industrial Internet of Things: A systematic literature review. *Computers in Industry*, 137:103614, May 2022.

[20] M. Schulze Darup, A. B. Alexandru, D. E. Quevedo, and G. J. Pappas. Encrypted control for networked systems – An illustrative introduction and current challenges, October 2020. arXiv:2010.00268 [cs, eess, math].

[21] Gabriel de Albuquerque Gleizer and Manuel Mazo. Self-triggered output-feedback control of LTI systems subject to disturbances and noise. *Automatica*, 120:109129, October 2020.

[22] Gabriel de Albuquerque Gleizer and Manuel Mazo. Computing the average inter-sample time of event-triggered control using quantitative automata. *Nonlinear Analysis: Hybrid Systems*, 47:101290, February 2023.

[23] Gabriel de Albuquerque Gleizer and Manuel Mazo, Jr. Computing the sampling performance of event-triggered control. In *Hybrid Systems: Computation and Control*, Tennessee, March 2021. Delft University of Technology. Publication Title: arXiv e-prints ADS Bibcode: 2021arXiv210300919D Type: article.

[24] C. De Persis and P. Tesi. Networked control of nonlinear systems under Denial-of-Service. *Systems & Control Letters*, 96:124–131, October 2016.

[25] Frederik Michel Dekking, Cornelis Kraaikamp, Hendrik Paul Lopuhaä, and Ludolf Erwin Meester. *A Modern Introduction to Probability and Statistics.* Springer Texts in Statistics. Springer, London, 2005.

[26] Giannis Delimpaltadakis. *Grasping the Sampling Behaviour of Event-Triggered Control: Self-Triggered Control, Abstractions and Formal Analysis.* Phd Thesis, Delft University of Technology, Delft, 2022.

[27] Giannis Delimpaltadakis, Gabriel de Albuquerque Gleizer, Ivo van Straalen, and Manuel Mazo Jr. ETCetera: beyond Event-Triggered Control. In *25th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '22, pages 1–11, New York, NY, USA, May 2022. Association for Computing Machinery.

[28] Giannis Delimpaltadakis, Luca Laurenti, and Manuel Mazo Jr. Formal Analysis of the Sampling Behaviour of Stochastic Event-Triggered Control, February 2022. arXiv:2202.10178 [cs, eess, math].

[29] Yingjie Deng, Dingxuan Zhao, and Tao Liu. Self-triggered tracking control of underactuated surface vessels with stochastic noise. In *2021 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, pages 266–273, June 2021.

[30] V. S. Dolk, P. Tesi, C. De Persis, and W. P. M. H. Heemels. Event-Triggered Control Systems Under Denial-of-Service Attacks. *IEEE Transactions on Control of Network Systems*, 4(1):93–105, March 2017. Conference Name: IEEE Transactions on Control of Network Systems.

[31] M. C. F. Donkers and W. P. M. H. Heemels. Output-Based Event-Triggered Control With Guaranteed $\mathcal{L}_{\infty}$-Gain and Improved and Decentralized Event-Triggering. *IEEE Transactions on Automatic Control*, 57(6):1362–1376, June 2012. Conference Name: IEEE Transactions on Automatic Control.

[32] M.C.F. Donkers. *Networked and event-triggered control systems.* Phd Thesis 1 (Research TU/e / Graduation TU/e), Technische Universiteit Eindhoven, Eindhoven, 2011. ISBN: 9789038627496.

[33] M.C.F. Donkers, P. Tabuada, and W.P.M.H. Heemels. On the minimum attention control problem for linear systems: A linear programming approach. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 4717–4722, December 2011. ISSN: 0743-1546.

[34] Dajun Du, Changda Zhang, Xue Li, Minrui Fei, and Huiyu Zhou. Attack Detection for Networked Control Systems Using Event-Triggered Dynamic Watermarking. *IEEE Transactions on Industrial Informatics*, 19(1):351–361, January 2023. Conference Name: IEEE Transactions on Industrial Informatics.

[35] D. Ender. Process Control Performance : Not as Good as You Think. *Control Engineering*, page 6, September 1993.

[36] Alina Eqtami, Dimos V. Dimarogonas, and Kostas J. Kyriakopoulos. Event-triggered control for discrete-time systems. In *Proceedings of the 2010 American Control Conference*, pages 4719–4724, June 2010. ISSN: 2378-5861.

[37] Chongrong Fang, Yifei Qi, Peng Cheng, and Wei Xing Zheng. Cost-effective watermark based detector for replay attacks on cyber-physical systems. In *2017 11th Asian Control Conference (ASCC)*, pages 940–945, December 2017.

[38] Xing Fang and Wen-Hua Chen. Model Predictive Control with Preview: Recursive Feasibility and Stability, February 2022. arXiv:2202.12585 [cs, eess].

[39] Davide Fauri, Bart de Wijs, Jerry den Hartog, Elisa Costante, Emmanuele Zambon, and Sandro Etalle. Encryption in ICS networks: A blessing or a curse? In *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 289–294, October 2017.

[40] Federal Bureau of Investigation (FBI), Cybersecurity and Infrastructure Security Agency (CISA), and Department of Energy (DOE). Tactics, Techniques, and Procedures of Indicted State-Sponsored Russian Cyber Actors Targeting the Energy Sector. White paper 1, Joint Cybersecurity Advisory (CSA), Washington, March 2022.

[41] Riccardo M. G. Ferrari and André M. H. Teixeira. Detection and Isolation of Replay Attacks through Sensor Watermarking. *IFAC-PapersOnLine*, 50(1):7363–7368, July 2017.

[42] Riccardo M. G. Ferrari and André M. H. Teixeira. A Switching Multiplicative Watermarking Scheme for Detection of Stealthy Cyber-Attacks. *IEEE Transactions on Automatic Control*, 66(6):2558–2573, June 2021. Conference Name: IEEE Transactions on Automatic Control.

[43] Riccardo M.G. Ferrari and André M.H. Teixeira. Detection of Cyber-Attacks: A Multiplicative Watermarking Scheme. In Riccardo M.G. Ferrari and André M.H. Teixeira, editors, *Safety, Security and Privacy for Cyber-Physical Systems*, Lecture Notes in Control and Information Sciences, pages 173–201. Springer, 2021.

[44] B.A. Francis and T.T. Georgiou. Stability theory for linear time-invariant plants with periodic digital controllers. *IEEE Transactions on Automatic Control*, 33(9):820–832, September 1988. Conference Name: IEEE Transactions on Automatic Control.

[45] Gene F. Franklin, J. David Powell, and Michael L. Workman. *Digital control of dynamic systems*. Ellis-Kagle Press, Half Moon Bay, CA, 3rd ed. ; reprinted in 2006 with corrections edition, 2006.

[46] Anqi Fu, Ivana Tomic, and Julie A. McCann. Asynchronous Sampling for Decentralized Periodic Event-Triggered Control. In *2019 American Control Conference (ACC)*, pages 145–150, July 2019. ISSN: 2378-5861.

[47] Alexander J. Gallo, Sribalaji C. Anand, André M. H. Teixeira, and Riccardo M. G. Ferrari. Design of multiplicative watermarking against covert attacks. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 4176–4181, December 2021. ISSN: 2576-2370.

[48] Xiaohua Ge, Qing-Long Han, Xian-Ming Zhang, and Derui Ding. Dynamic Event-triggered Control and Estimation: A Survey. *International Journal of Automation and Computing*, 18(6):857–886, December 2021.

[49] Gabriel de A. Gleizer, Khushraj Madnani, and Manuel Mazo. Self-Triggered Control for Near-Maximal Average Inter-Sample Time. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 1308–1313, December 2021. ISSN: 2576-2370.

[50] Gabriel de A. Gleizer and Manuel Mazo. Self-Triggered Output Feedback Control for Perturbed Linear Systems. *IFAC-PapersOnLine*, 51(23):248–253, January 2018.

[51] Gabriel de A. Gleizer and Manuel Mazo. Scalable Traffic Models for Scheduling of Linear Periodic Event-Triggered Controllers. *IFAC-PapersOnLine*, 53(2):2726–2732, January 2020.

[52] Gabriel de Albuquerque Gleizer and Manuel Mazo Jr. Chaos and order in event-triggered control, February 2022. arXiv:2201.04462 [cs, eess].

[53] Tom Gommans, Duarte Antunes, Tijs Donkers, Paulo Tabuada, and Maurice Heemels. Self-triggered linear quadratic control. *Automatica*, 50(4):1279–1287, April 2014.

[54] Antonio González, Angel Cuenca, Julián Salt, and Jelle Jacobs. Robust stability analysis of an energy-efficient control in a Networked Control System with application to unmanned ground vehicles. *Information Sciences*, 578:64–84, November 2021.

[55] Andy Greenberg. Sandworm Hackers Caused Another Blackout in Ukraine—During a Missile Strike. *WIRED*, November 2023. Section: tags.

[56] Ziyang Guo, Dawei Shi, Karl Henrik Johansson, and Ling Shi. Worst-case stealthy innovation-based linear attack on remote state estimation. *Automatica*, 89:117–124, March 2018.

[57] Romulo Meira Góes, Eunsuk Kang, Raymond Kwong, and Stéphane Lafortune. Stealthy deception attacks for cyber-physical systems. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 4224–4230, December 2017.

[58] W. P. Maurice H. Heemels, Romain Postoyan, M. C. F. (Tijs) Donkers, Andrew R. Teel, Adolfo Anta, Paulo Tabuada, and Dragan Nešić. Periodic Event-Triggered Control. In Marek Miskowicz, editor, *Event-Based Control and Signal Processing*, pages 104–120. CRC Press, 0 edition, September 2018.

[59] Navid Hashemi and Justin Ruths. Generalized chi-squared detector for LTI systems with non-Gaussian noise. In *IEEE International Conference on Cyber Technology in Automation*, pages 404–410, Philadelphia, July 2019. IEEE.

[60] W. P. M. H. Heemels and M. C. F. Donkers. Model-based periodic event-triggered control for linear systems. *Automatica*, 49(3):698–711, March 2013.

[61] W. P. M. H. Heemels, M. C. F. Donkers, and Andrew R. Teel. Periodic Event-Triggered Control for Linear Systems. *IEEE Transactions on Automatic Control*, 58(4):847–861, April 2013. Conference Name: IEEE Transactions on Automatic Control.

[62] W. P. M. H. Heemels, J. H. Sandee, and P. P. J. Van Den Bosch. Analysis of event-driven controllers for linear systems. *International Journal of Control*, 81(4):571–590, April 2008. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00207170701506919.

[63] W.P.M.H. (Maurice) Heemels, M.C.F. Donkers, and A.R. Teel. Periodic event-triggered control based on state feedback. In *IEEE Conference on Decision and Control and European Control Conference*, pages 2571–2576, December 2011.

[64] W.P.M.H. (Maurice) Heemels, Karl Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. In *Proceedings of the IEEE Conference on Decision and Control*, pages 3270–3285, December 2012.

[65] Laurentiu Hetel, Christophe Fiter, Hassan Omran, Alexandre Seuret, Emilia Fridman, Jean-Pierre Richard, and Silviu Iulian Niculescu. Recent developments on the stability of systems with aperiodic sampling: An overview. *Automatica*, 76:309–335, February 2017.

[66] Andreas Hoehn and Ping Zhang. Detection of covert attacks and zero dynamics attacks in cyber-physical systems. In *2016 American Control Conference (ACC)*, pages 302–307, July 2016. ISSN: 2378-5861.

[67] Sahand Hadizadeh Kafash, Jairo Giraldo, Carlos Murguia, Alvaro A. Cardenas, and Justin Ruths. Constraining Attacker Capabilities Through Actuator Saturation. In *2018 Annual American Control Conference (ACC)*, pages 986–991, June 2018. ISSN: 2378-5861.

[68] Asadi Khashooei. *Event-triggered control for linear systems with performance and rate guarantees : an approximate dynamic programming approach.* Phd Thesis, Technische Universiteit Eindhoven, Eindhoven, June 2017.

[69] Jihan Kim, Juhoon Back, Gyunghoon Park, Chanhwa Lee, Hyungbo Shim, and Petros G. Voulgaris. Neutralizing zero dynamics attack on sampled-data systems via generalized holds. *Automatica*, 113:108778, March 2020.

[70] Kosuke Kimura and Hideaki Ishii. Quantized Zero Dynamics Attacks against Sampled-data Control Systems, March 2023. arXiv:2303.11982 [cs, eess].

[71] Corbin Klett, Matthew Abate, Yongeun Yoon, Samuel Coogan, and Eric Feron. Bounding the State Covariance Matrix for Switched Linear Systems with Noise. In *2020 American Control Conference (ACC)*, pages 2876–2881, July 2020. ISSN: 2378-5861.

[72] Tomoki Koga, Mitsuaki Ishitobi, and Masatoshi Nishi. A sampling zero of a sampled-data model for continuous-time systems with relative degree two. In *Proceedings of the 2010 International Conference on Modelling, Identification and Control*, pages 751–755, July 2010.

[73] Jérémie Kreiss and Jean-François Trégouët. Input redundancy: Definitions, taxonomy, characterizations and application to over-actuated systems. *Systems & Control Letters*, 158:105060, December 2021.

[74] Chanhwa Lee, Hyungbo Shim, and Yongsoon Eun. On Redundant Observability: From Security Index to Attack Detection and Resilient State Estimation. *IEEE Transactions on Automatic Control*, 64(2):775–782, February 2019. Conference Name: IEEE Transactions on Automatic Control.

[75] Robert M. Lee, Michael J. Assante, and Tim Conway. Analysis of the Cyber Attack on the Ukrainian Power Grid. White paper 1, E-ISAC, Washington, March 2016.

[76] Tongxiang Li, Zidong Wang, Lei Zou, Bo Chen, and Li Yu. A dynamic encryption–decryption scheme for replay attack detection in cyber–physical systems. *Automatica*, 151:110926, May 2023.

[77] Zishuo Li, Anh Tung Nguyen, André Teixeira, Yilin Mo, and Karl H. Johansson. Secure State Estimation with Asynchronous Measurements against Malicious Measurement-data and Time-stamp Manipulation, March 2023. arXiv:2303.17514 [cs, eess].

[78] Steffen Linsenmayer, Dimos V. Dimarogonas, and Frank Allgöwer. A non-monotonic approach to periodic event-triggered control with packet loss. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 507–512, December 2016.

[79] Steffen Linsenmayer, Dimos V. Dimarogonas, and Frank Allgöwer. Periodic event-triggered control for networked control systems based on non-monotonic Lyapunov functions. *Automatica*, 106:35–46, August 2019.

[80] Hanxiao Liu, Yilin Mo, and Karl Henrik Johansson. Active Detection Against Replay Attack: A Survey on Watermark Design for Cyber-Physical Systems. In Riccardo M.G. Ferrari and André M. H. Teixeira, editors, *Safety, Security and Privacy for Cyber-Physical Systems*, Lecture Notes in Control and Information Sciences, pages 145–171. Springer International Publishing, Cham, 2021.

[81] Hanxiao Liu, Jiaqi Yan, Yilin Mo, and Karl Henrik Johansson. An On-line Design of Physical Watermarks, September 2018. arXiv:1809.05299 [cs, math].

[82] Qinyuan Liu, Zidong Wang, Xiao He, and D.H. Zhou. A survey of event-based strategies on control and estimation. *Systems Science & Control Engineering*, 2(1):90–97, December 2014. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/21642583.2014.880387.

[83] Shixian Luo and Feiqi Deng. On Event-Triggered Control of Nonlinear Stochastic Systems. *IEEE Transactions on Automatic Control*, 65(1):369–375, January 2020. Conference Name: IEEE Transactions on Automatic Control.

[84] Guoqi Ma, Xinghua Liu, Prabhakar R. Pagilla, and Xinghuo Yu. Two-Channel Periodic Event-Triggered Observer-Based Repetitive Control for Periodic Reference Tracking. In *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pages 2469–2474, October 2018. ISSN: 2577-1647.

[85] Manuel Mazo, Adolfo Anta, and Paulo Tabuada. An ISS self-triggered implementation of linear controllers. *Automatica*, 46(8):1310–1314, August 2010.

[86] Manuel Mazo Jr, Adolfo Anta, and P Tabuada. On self-triggered control for linear systems: Guarantees and complexity. *European control conference*, January 2009.

[87] Divyabh Mishra. The Stages Of Industry 4.0: Where Are You Now? *Forbes*, September 2020. Section: Innovation.

[88] Aritra Mitra and Shreyas Sundaram. Byzantine-resilient distributed observers for LTI systems. *Automatica*, 108:108487, October 2019.

[89] Yilin Mo, Emanuele Garone, Alessandro Casavola, and Bruno Sinopoli. False data injection attacks against state estimation in wireless sensor networks. In *49th IEEE Conference on Decision and Control (CDC)*, pages 5967–5972, December 2010. ISSN: 0191-2216.

[90] Yilin Mo and Bruno Sinopoli. Secure control against replay attacks. In *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 911–918, September 2009.

[91] Yilin Mo, Sean Weerakkody, and Bruno Sinopoli. Physical Authentication of Control Systems: Designing Watermarked Control Inputs to Detect Counterfeit Sensor Outputs. *IEEE Control Systems Magazine*, 35(1):93–109, February 2015. Conference Name: IEEE Control Systems Magazine.

[92] Carlos Murguia and Justin Ruths. CUSUM and chi-squared attack detection of compromised sensors. In *2016 IEEE Conference on Control Applications (CCA)*, pages 474–480, September 2016.

[93] Bengt Mårtensson. Zeros of sampled systems. Master's thesis, Lund Institute of Technology, Lund, December 1982.

[94] Mohammad Naghnaeian, Nabil Hirzallah, and Petros G. Voulgaris. Dual rate control for security in cyber-physical systems. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 1415–1420, December 2015.

[95] Mohammad Naghnaeian, Nabil H. Hirzallah, and Petros G. Voulgaris. Security via multirate control in cyber–physical systems. *Systems & Control Letters*, 124:12–18, February 2019.

[96] Rasika B. Naik and Udayprakash Singh. A Review on Applications of Chaotic Maps in Pseudo-Random Number Generators and Encryption. *Annals of Data Science*, January 2022.

[97] Minghui Ou, Zhiyong Yang, Zhenjie Yan, Mingkun Ou, Shuanghong Liu, Shan Liang, and Shengjiu Liu. Stability of Zeros for Sampled-Data Models with Triangle Sample and Hold Implemented by Zero-Order Hold. *Machines*, 10(5):386, May 2022. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

[98] Gyunghoon Park, Chanhwa Lee, Hyungbo Shim, Yongsoon Eun, and Karl H. Johansson. Stealthy Adversaries Against Uncertain Cyber-Physical Systems: Threat of Robust Zero-Dynamics Attack. *IEEE Transactions on Automatic Control*, 64(12):4907–4919, December 2019. Conference Name: IEEE Transactions on Automatic Control.

[99] Syed Ahmed Pasha and Ayesha Ayub. Zero-dynamics attacks on networked control systems. *Journal of Process Control*, 105:99–107, September 2021.

[100] Chen Peng and Fuqiang Li. A survey on recent advances in event-triggered communication and control. *Information Sciences*, 457-458:113–125, August 2018.

[101] Ian R. Petersen and Andrey V. Savkin. *Robust Kalman Filtering for Signals and Systems with Large Uncertainties*. Birkhäuser, Boston, MA, 1999.

[102] Ed Powers, Sean Peasley, Rene Waslo, Byron Fletcher, and David Dinh. Examining the Industrial Control System Cyber Risk Gap. Reader 1, Deloitte, London, January 2015.

[103] S. Prakash, E.P. van Horssen, D. Antunes, and W.P.M.H. Heemels. Self-triggered and event-driven control for linear systems with stochastic delays. In *2017 American Control Conference (ACC)*, pages 3023–3028, May 2017. ISSN: 2378-5861.

[104] Road2CPS. Guide to Cyber-Physical Systems Engineering. Brochure 1, DESTECS, Berlin, December 2016.

[105] Raffaele Romagnoli, Sean Weerakkody, and Bruno Sinopoli. A Model Inversion Based Watermark for Replay Attack Detection with Output Tracking. In *2019 American Control Conference (ACC)*, pages 384–390, July 2019. ISSN: 2378-5861.

[106] R. Sakthivel, S. Mohanapriya, H. R. Karimi, and P. Selvaraj. A Robust Repetitive-Control Design for a Class of Uncertain Stochastic Dynamical Systems. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 64(4):427–431, April 2017. Conference Name: IEEE Transactions on Circuits and Systems II: Express Briefs.

[107] Henrik Sandberg. Cyber-Physical Security. In John Baillieul and Tariq Samad, editors, *Encyclopedia of Systems and Control*, pages 1–8. Springer, London, 2020.

[108] Henrik Sandberg, Vijay Gupta, and Karl H. Johansson. Secure Networked Control Systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):445–464, 2022. _eprint: https://doi.org/10.1146/annurev-control-072921-075953.

[109] Bharadwaj Satchidanandan and P. R. Kumar. Secure control of networked cyber-physical systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 283–289, December 2016.

[110] Bharadwaj Satchidanandan and P. R. Kumar. Dynamic Watermarking: Active Defense of Networked Cyber–Physical Systems. *Proceedings of the IEEE*, 105(2):219–240, February 2017. Conference Name: Proceedings of the IEEE.

[111] D. Sbarbaro, J. M. Gomes da Silva Jr., and L. G. Moreira. Event-Triggered Tracking Control: a Discrete-Time Approach. *IFAC-PapersOnLine*, 53(2):4565–4570, January 2020.

[112] Luca Schenato. To Zero or to Hold Control Inputs With Lossy Links? *IEEE Transactions on Automatic Control*, 54(5):1093–1099, May 2009. Conference Name: IEEE Transactions on Automatic Control.

[113] Rohan Chandra Shekhar. *Variable horizon model predictive control: robustness and optimality.* Phd Thesis, University of Cambridge, Cambridge, July 2012.

[114] Hyungbo Shim, Juhoon Back, Yongsoon Eun, Gyunghoon Park, and Jihan Kim. Zero-dynamics Attack, Variations, and Countermeasures, January 2021. arXiv:2101.00556 [cs, eess].

[115] Takumi Shinohara and Toru Namerikawa. Distributed secure state estimation with a priori sparsity information. *IET Control Theory & Applications*, 16(11):1086–1097, 2022. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/cth2.12287.

[116] Joris Sijs, Mircea Lazar, and W.P.M.H. Heemels. On integration of event-based estimation and robust MPC in a feedback loop. In *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, HSCC '10, pages 31–40, New York, NY, USA, April 2010. Association for Computing Machinery.

[117] Jill Slay and Michael Miller. Lessons Learned from the Maroochy Water Breach. In *Post-Proceedings of the First Annual IFIP Working Group*, volume 253, pages 73–82, New Hampshire, March 2007. IFIP.

[118] Rhett Smith. Cryptography Concepts and Effects on Control System Communications. *Sensible Cybersecurity for Power Systems: A Collection of Technical Papers Representing Modern Solutions*, 2018.

[119] Straits Research. Industrial Wireless Sensor Network Market Size is projected to reach USD 8.62 billion by 2030, growing at a CAGR of 7.13%. *GlobeNewswire News Room*, June 2023.

[120] Hongtao Sun, Chen Peng, Weidong Zhang, Taicheng Yang, and Zhiwen Wang. Security-based resilient event-triggered control of networked control systems under denial of service attacks. *Journal of the Franklin Institute*, 356(17):10277–10295, November 2019.

[121] Yuan-Cheng Sun and Guang-Hong Yang. Periodic event-triggered resilient control for cyber-physical systems under denial-of-service attacks. *Journal of the Franklin Institute*, 355(13):5613–5631, September 2018.

[122] Paulo Tabuada. Event-Triggered Real-Time Scheduling of Stabilizing Control Tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, September 2007. Conference Name: IEEE Transactions on Automatic Control.

[123] André Teixeira, Iman Shames, Henrik Sandberg, and Karl H. Johansson. Revealing stealthy attacks in control systems. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1806–1813, October 2012.

[124] André Teixeira, Iman Shames, Henrik Sandberg, and Karl Henrik Johansson. A secure control framework for resource-limited adversaries. *Automatica*, 51:135–148, January 2015.

[125] Michel Verhaegen and Vincent Verdult. *Filtering and System Identification: A Least Squares Approach*. Cambridge University Press, 1 edition, April 2007.

[126] Xiaofeng Wang and Michael D. Lemmon. Self-Triggered Feedback Control Systems With Finite-Gain L2 Stability. *IEEE Transactions on Automatic Control*, 54(3):452–467, March 2009. Conference Name: IEEE Transactions on Automatic Control.

[127] Zhao Wang, Jian Sun, and Yongqiang Bai. Stability Analysis of Event-Triggered Networked Control Systems with Time-Varying Delay and Packet Loss. *Journal of Systems Science and Complexity*, 34(1):265–280, February 2021.

[128] David E. Whitehead, Kevin Owens, Dennis Gammel, and Jess Smith. Ukraine cyber-induced power outage: Analysis and practical mitigation strategies. In *2017 70th Annual Conference for Protective Relay Engineers (CPRE)*, pages 1–8, April 2017. ISSN: 2474-9753.

[129] Wei Wu. *Event-triggered Control of Linear Systems with Application to Embedded Control Systems*. Phd Thesis, Technische Universität Kaiserslautern, Kaiserslautern, 2014.

[130] Junlin Xiong and James Lam. Stabilization of linear systems over networks with bounded packet loss. *Automatica*, 43(1):80–87, January 2007.

[131] Bahram Yaghooti, Raffaele Romagnoli, and Bruno Sinopoli. Physical watermarking for replay attack detection in continuous-time systems. *European Journal of Control*, 62:57–62, November 2021.

[132] Hao Yu and Fei Hao. The existence of Zeno behavior and its application to finite-time event-triggered control. *Science China Information Sciences*, 63(1):139201, December 2019.

[133] Hao Yu and Fei Hao. Set-point output tracking problem for linear plants via periodic event-triggered control. *IET Control Theory & Applications*, 14, April 2020.

[134] Amirreza Zaman, Behrouz Safarinejadian, and Wolfgang Birk. Security Analysis and Fault Detection Against Stealthy Replay Attacks. *International Journal of Control*, 95:1–22, December 2020.

[135] Cheng Zeng, Shan Liang, Yuzhe Zhang, Jiaqi Zhong, and Yingying Su. Improving the stability of discretization zeros with the Taylor method using a generalization of the fractional-order hold. *International Journal of Applied Mathematics and Computer Science*, 24(4):745–757, December 2014.

[136] Heng Zhang, Peng Cheng, Ling Shi, and Jiming Chen. Optimal Denial-of-Service Attack Scheduling With Energy Constraint. *IEEE Transactions on Automatic Control*, 60(11):3023–3028, November 2015. Conference Name: IEEE Transactions on Automatic Control.

[137] Jiaxuan Zhang. Defense Against Malicious Parameter Identification. Master's thesis, Delft University of Technology, Delft, June 2023.

[138] Jinhui Zhang and Gang Feng. Event-driven observer-based output feedback control for linear systems. *Automatica*, 50(7):1852–1859, July 2014.

[139] Xian-Ming Zhang, Qing-Long Han, and Xinghuo Yu. Survey on Recent Advances in Networked Control Systems. *IEEE Transactions on Industrial Informatics*, 12(5):1740–1752, October 2016. Conference Name: IEEE Transactions on Industrial Informatics.

[140] Xiao-Guang Zhang, Guang-Hong Yang, and Xiu-Xiu Ren. Network steganography based security framework for cyber-physical systems. *Information Sciences*, 609:963–983, September 2022.

[141] Bin Zhou. On the relative degree and normal forms of linear systems by output transformation with applications to tracking. *Automatica*, 148:110800, February 2023.

[142] K. J. Åström, P. Hagander, and J. Sternby. Zeros of sampled systems. *Automatica*, 20(1):31–38, January 1984.

[143] Karl Johan Åström and B.M. Bernhardsson. Comparison of Riemann and Lebesgue sampling for first order stochastic systems. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 2, pages 2011–2016 vol.2, December 2002. ISSN: 0191-2216.

Glossary

List of Acronyms

ARE	algebraic Riccati equation	61
BMI	bilinear matrix inequality	18
CETC	continuous event-triggered control	12
CPS	cyber-physical system	31
DARE	discrete algebraic Riccati equation	71
DoS	denial-of-service	25
ETC	event-triggered control	69
FDI	false data injection	54
FIR	finite impulse response	31
GES	globally exponentially stable	72
ICS	industrial control system	52
IET	inter-event time	95
i.i.d.	independent and identically distributed	46
IoT	internet of things	1
ISS	input-to-state stable	72
IT	information technology	23
KKT	Karush–Kuhn–Tucker	38
KL	Kullback-Leibler	94
LI	linear impulsive	70
LLS	linear least squares	53
LMI	linear matrix inequality	72
LQG	linear–quadratic–Gaussian	42
LQR	linear–quadratic–regulator	60
LTI	linear time-invariant	52

MIMO	multiple-input and multiple-output	76
MSS	mean square stable	34
MPC	model predictive control	16
NCPS	networked cyber-physical system	29
NCS	networked control system	9
PD	positive-definite	37
PETC	periodic event-triggered control	72
PID	proportional–integral–derivative	22
PMF	probability mass function	94
PWL	piecewise linear	18
QP	quadratic program	37
SCADA	supervisory control and data acquisition	5
s.e.	standard error	46
RHS	right-hand side	20
SISO	single-input and single-output	51
SL	switched linear	71
STC	self-triggered control	72
TTC	time-triggered control	50
VHMPC	variable horizon model predictive control	69
ZDA	zero dynamic attack	76
ZOH	zero-order hold	52

List of Symbols

This list is in alphabetical order with the Latin alphabet first followed by the Greek alphabet.

Table B-1: List of symbols

Symbol	Description	Page
$\boldsymbol{a}$	Attack vector	24
$\underset{\sim}{\boldsymbol{A}}$	State matrix, continuous-time	6
$\boldsymbol{A}$	State matrix, discretized	9
$\boldsymbol{A}_\kappa$	State matrix, switched	14
$\boldsymbol{A}_0$	State matrix at absence of event	18
$\boldsymbol{A}_1$	State matrix at occurrence of event	18
$\boldsymbol{B}$	Input matrix	6
$\boldsymbol{B}_\kappa$	Input matrix, switched	14
$\boldsymbol{c}$	Controller state	7
$\boldsymbol{c}$	Measurement row	76

Symbol	Description	Page
$\boldsymbol{C}$	Measurement matrix	6
$\mathbb{C}$	Set of all complex numbers	51
$\boldsymbol{D}$	Feed-forward matrix	6
D_{KL}	Kullback-Leibler (KL) divergence	36
D_{J}	Jeffreys divergence	36
$\boldsymbol{E}$	Disturbance matrix	6
$\boldsymbol{f}_i$	Adversary state vector	55
$\tilde{\boldsymbol{f}}_i$	Adversary error vector	56
$\boldsymbol{F}$	Adversary feedback matrix	55
$\underset{\sim}{\boldsymbol{F}}$	Perturbation matrix	6
g_i	Detection signal	32
h	Sampling period	9
$\boldsymbol{H}_{\kappa}$	Time-varying observer gain	32
i	Event index	11
$\boldsymbol{J}_0$	Jump matrix at absence of event	71
$\boldsymbol{J}_1$	Jump matrix at occurrence of event	71
k	Discrete-time index	9
k_i	i-th discrete-time event index	7
n_{u}	Number of inputs	6
n_{x}	Number of states	6
n_{y}	Number of outputs	6
n_{z}	Relative degree, discrete-time	52
n_{ν}	Relative degree, continuous-time	52
n_{ξ}	Number of augmented state	17
N_{w}	Watermarker filter order	42
N_{i}	Total number of events	43
$\boldsymbol{O}$	Observability matrix	54
$\boldsymbol{N}(\kappa)$	Trigger evolution matrix	14
m	Length of limiting cycle	59
p	Inter-event index probability mass function (PMF)	34
$\boldsymbol{p}$	Augmented state vector, discrete-time	13
$\boldsymbol{p}$	Probability distribution vector	35
$\hat{p}_{\mathrm{fp}}$	Estimated false alarm rate	46
P	Transfer function	57
P^{-1}	Inverse regularized lower incomplete gamma function	32
p_{fp}	False alarm rate	32
$\boldsymbol{q}[k]$	Augmented state vector, discrete-time	18
$\boldsymbol{Q}$	Triggering matrix	11
$\boldsymbol{r}$	Reference vector	19
$\boldsymbol{R}$	Rosenbrock system matrix	51
$\mathbb{R}$	Set of all real numbers	9
s	Elapsed time	11
t	Continuous-time	6
t_i	i-th event time	11
T_{a}	Attack start time	29

Symbol	Description	Page
T_r	Reference signal period	40
$\boldsymbol{u}$	Input vector	6
$\boldsymbol{w}$	Load disturbance vector	6
$\boldsymbol{v}$	Measurement noise vector	6
$\boldsymbol{x}$	State vector, discrete-time	9
$\boldsymbol{x}_i$	i-th sampled state vector	11
$\tilde{\boldsymbol{x}}_i$	i-th observation error	21
$\hat{\boldsymbol{x}}_i$	i-th state estimate	21
$\boldsymbol{y}$	Output vector	6
$\boldsymbol{y}'$	Output vector, received	15
$\boldsymbol{\gamma}$	Output vector, encrypted	29
$\boldsymbol{z}_i$	Residual	26
	- - - - - - - - - Greek - - - - - -	
γ	Early triggering penalty	36
γ	Gain	17
Γ	Event prediction function	15
Δi	Loop length	29
ΔT	Delay length	29
ϵ	Margin parameter	15
ϵ_0	Numerically positive threshold	37
η	Detector threshold	32
$\eta(s)$	Dynamic buffer variable	47
ρ	Decay rate	17
σ	Triggering parameter	11
$\boldsymbol{\Sigma}$	Covariance matrix	9
$\bar{\kappa}$	Upper bound of inter-event index	14
$\vec{\kappa}_\mathrm{avg}$	Average inter-event index	43
κ_i	i-th inter-event index	14
$\bar{\kappa}_i$	i-th inter-event index deadline	34
κ_max	Largest inter-event index	14
κ_min	Smallest inter-event index	14
$[\![\kappa_i]\!]$	i-th watermarked inter-event index	34
$\boldsymbol{\nu}$	Measurement noise, continuous-time	6
$\boldsymbol{\xi}$	Augmented state vector, continuous-time	11
Ξ	Jamming signal	24
$\bar{\tau}$	Upper bound of the $\underline{\text{i}}$nter-$\underline{\text{e}}$vent $\underline{\text{t}}$ime (IET)	11
$\vec{\tau}_\mathrm{acg}$	Average IET	43
τ_i	i-th inter-event time	11
$\boldsymbol{\upsilon}$	Input vector, continuous-time	6
ϕ	Triggering function	11
$\boldsymbol{\Phi}$	State evolution function	16
$\boldsymbol{\chi}$	State vector, continuous-time	6
$\boldsymbol{\omega}$	Load disturbance, continuous-time	6