PPGI PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA
Universidade Federal do Rio de Janeiro

Universidade Federal do Rio de Janeiro
Instituto de Matemática
Instituto Tércio Pacitti
Programa de Pós-Graduação em Informática

System Identification Attacks, Model-based Offensives and Countermeasures in Networked Control Systems

Alan Oliveira de Sá

Rio de Janeiro
2019

Universidade Federal do Rio de Janeiro

Instituto de Matemática

Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais

Programa de Pós-Graduação em Informática

Alan Oliveira de Sá

System Identification Attacks, Model-based Offensives and Countermeasures in Networked Control Systems

Doctoral Thesis submitted to the Postgraduate Program in Informatics, Institute of Mathematics and Tércio Parcitti Institute of the Federal University of Rio de Janeiro (Concentration area: Adaptive Complex Systems), in partial fulfillment of the requirements for the degree of Doctor in Informatics.

Supervisor: Luiz Fernando Rust da Costa Carmo, Dr. UPS.

Co-supervisor: Raphael Carlos Santos Machado, D.Sc.

Rio de Janeiro

2019

CIP - Catalogação na Publicação

O111s

Oliveira de Sá, Alan
System Identification Attacks, Model-based Offensives and Countermeasures in Networked Control Systems / Alan Oliveira de Sá. -- Rio de Janeiro, 2019.
230 f.

Orientador: Luiz Fernando Rust da Costa Carmo.
Coorientador: Raphael Carlos Santos Machado.
Tese (doutorado) - Universidade Federal do Rio de Janeiro, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Programa de Pós-Graduação em informática, 2019.

1. Segurança. 2. Sistemas Físicos Cibernéticos. 3. Sistemas de Controle em Rede. 4. Ataques. 5. Contramedidas. I. Rust da Costa Carmo, Luiz Fernando, orient. II. Carlos Santos Machado, Raphael, coorient. III. Título.

Elaborado pelo Sistema de Geração Automática da UFRJ com os dados fornecidos pelo(a) autor(a), sob a responsabilidade de Miguel Romeu Amorim Neto - CRB-7/6283.

Alan Oliveira de Sá

System Identification Attacks, Model-based Offensives and Countermeasures in Networked Control Systems

Doctoral Thesis submitted to the Postgraduate Program in Informatics, Institute of Mathematics and Tércio Parcitti Institute of the Federal University of Rio de Janeiro (Concentration area: Adaptive Complex Systems), in partial fulfillment of the requirements for the degree of Doctor in Informatics.

Approved. Rio de Janeiro, May 17th, 2019:

Luiz Fernando Rust da Costa Carmo,
Dr. UPS.(Supervisor)

Raphael Carlos Santos Machado, D.Sc.
(Co-supervisor)

Cláudio Miceli de Farias, D.Sc.

Nival Nunes de Almeida, D.Sc.

António Casimiro Ferreira da Costa,
D.Sc.

Dedico este trabalho às minhas filhas Beatriz e Liz, e à minha esposa Adriana, as quais me alimentaram com amor e carinho em todos os momentos desta jornada, mesmo quando, em virtude do estudo, não pude retribuir na mesma proporção. Vocês são a minha vida e a minha motivação! Aos meus pais Raul e Maria Luisa pelo exemplo, pela inestimável ajuda, pelas palavras de incentivo e pela minha formação em todos os sentidos. Ao meu irmão, Ruy, meu grande amigo, e à minha avó Beliza que sempre torceram por minhas conquistas. Aos meus sogros José Paulo e Elisabeth pelo apreço e por todo o apoio que recebi neste período.

ACKNOWLEDGEMENTS

Agradeço primeiramente a Deus pela vida e pelas oportunidades que tenho.

Agradeço aos professores Luiz Fernando Rust da C. Carmo e Raphael C. S. Machado pela orientação, pelos ensinamentos, pelos incentivos e pela amizade que recebi ao longo deste trabalho.

Agradeço ao Prof. Nival Nunes de Almeida, pela atenção e ajuda, não só no aspecto acadêmico, mas também pessoal e profissional.

Agradeço ao CIAW, em especial ao Capitão de Fragata (RM1) Emanuel Rodrigues de Zacharias, ao Capitão de Corveta Robson Ribeiro Carreira e ao Capitão de Mar e Guerra (RM1) Henrique Luiz Neto pelo grande apoio, o qual permitiu que eu me dedicasse à este projeto. Alguns minutos (ou algumas horas) de conversa podem fazer muita diferença quando encontramos pessoas com a sincera vontade de ajudar.

Uma etapa importante desse trabalho foi realizada junto aos pesquisadores do LaSIGE (Laboratório de Sistemas de Grande Escala), da Faculdade de Ciências da Universidade de Lisboa. Quero agradecer, portanto, ao professor António Casimiro, pela cordialidade, profissionalismo e apoio ao me receber na Universidade de Lisboa. Agradeço também ao professor Altair Santin (PUC-PR), por propiciar essa oportunidade através de um de seus projetos, e também à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), por prover os subsídios necessários à minha estadia em Portugal por meio do Projeto 99999.008512/2014-0.

Agradeço aos meus colegas de pesquisa e amigos que fiz ao longo desta caminhada, tanto no PPGI/UFRJ quanto no LaSIGE/FCUL, pela troca de conhecimentos, pelo companheirismo e pelos momentos de descontração.

Alan Oliveira de Sá

"Um menino caminha
E caminhando chega no muro
E ali logo em frente
A esperar pela gente o futuro está"
(Toquinho)

RESUMO

As vantagens do uso de redes de comunicação para interconectar controladores e plantas físicas têm motivado o crescente número de Sistemas de Controle em Rede, ou *Networked Control Systems* (NCS), na indústria e em infraestruturas críticas. Entretanto, esta integração expõe tais sistemas a novas ameaças, típicas do domínio cibernético. Neste contexto, estudos têm sido realizados com o objetivo de explorar as vulnerabilidades e propor soluções de segurança para NCSs. O presente trabalho, primeiramente, propõe dois ataques de identificação de sistemas: um ataque passivo; e um ataque ativo. Estes ataques, que utilizam metaheurísticas bioinspiradas para estimar os modelos do NCS atacado, são estudados e avaliados como ferramenta para o projeto de ofensivas furtivas/baseadas em modelo. Em seguida, o trabalho apresenta três ataques baseados em modelo: um novo ataque que opera por meio da perda controlada de pacotes no NCS; e dois ataques que operam por meio da injeção de dados no sistema. Os resultados demonstram que a informação fornecida pelos ataques de identificação de sistemas permite o desenvolvimento eficaz dos referidos ataques furtivos/baseados em modelo. Para amparar a discussão sobre a relação entre ataques de Identificação de Sistemas e ataques furtivos/baseados em modelo, este trabalho demandou a formalização de um conjunto de conceitos relacionados à furtividade e inteligência no contexto da segurança de NCSs. Sendo assim, uma contribuição adicional do trabalho é a proposição de uma terminologia que abarca toda uma nova classe de ataques em sistemas físicos cibernéticos. Por fim, esta tese propõe duas contramedidas que visam contribuir para a segurança de NCSs em casos de falha ou ausência de outros mecanismos de segurança convencionais – tais como criptografia, autenticação, e segmentação de redes. A primeira contramedida visa mitigar os ataques de identificação por meio de uma estratégia de controle chaveado. Os resultados indicam que esta contramedida é capaz de mitigar os ataques de Identificação de Sistema propostos – desencorajando a implementação de ataques furtivos/baseados em modelo – ao mesmo tempo em que desempenha um controle satisfatório da planta. A segunda contramedida visa detectar/identificar funções lineares e invariantes no tempo (LTI) executadas por ataques de injeção controlada de dados no NCS. Para aumentar a acurácia da contramedida, é proposta uma técnica de Integração de Impulsos de Ruído, ou *Noise Impulse Integration*, a qual foi desenvolvida utilizando como inspiração a técnica de integração de pulsos radar. Os resultados demonstram que esta contramedida é capaz de identificar funções LTI de ataque, de forma acurada, sem interferir no funcionamento do NCS quando o sistema está em operação normal.

Palavras-chaves: Segurança. Sistemas Físicos Cibernéticos. Sistemas de Controle em Rede. Ataques furtivos. Identificação de Sistemas. Contramedidas.

ABSTRACT

The advantages of using communication networks to interconnect controllers and physical plants motivate the increasing number of Networked Control Systems (NCS) in industrial facilities and critical infrastructures. However, this integration also exposes such control systems to new threats, typical of the cyber domain. In this context, studies have been conducted aiming to explore vulnerabilities and propose security solutions for NCSs. The present work, firstly, proposes two system identification attacks: a passive attack; and an active attack. These attacks, which use bioinspired metaheuristics to estimate the models of the attacked NCS, are studied and evaluated as an attack tool to support the design of covert/model-based offensives. Then, this work presents three model-based attacks: a novel attack that operates causing controlled data loss in the NCS; and two attacks that operate through the injection of false data into the system. The simulation results show that the information provided by these System Identification attacks allow the effective design of the referred covert/model-based offensives. To support the discussion regarding the relationship between System Identification attacks and covert/model-based offensives, this work required the formalization of a number of concepts related to covertness and intelligence in the context of the security of NCSs. Thus, an additional contribution of this work is the proposition of a terminology that encompasses a whole new class of attacks in cyber-physical systems. Finally, this thesis proposes two countermeasures intended to contribute to the security of NCSs in case of failure or absence of other conventional security mechanisms – such as encryption, authentication, and network segmentation. The first countermeasure aims to hinder the system identification attacks through a switching control strategy. The results indicate that this countermeasure is able to mitigate the proposed System Identification attacks – discouraging the implementation of covert/model-based attacks – at the same time that it performs a satisfactory plant control. The second countermeasure aims to detect/identify linear time-invariant (LTI) functions executed by controlled data injection attacks in NCSs. To increase the accuracy of this countermeasure, it is proposed the *Noise Impulse integration* technique, which was developed using the radar pulse integration technique as inspiration. The results demonstrate that this countermeasure is able to accurately identify LTI attack functions, without interfering with the NCS behavior when the system is in its normal operation.

Keywords: Security. Cyber Physical Systems. Networked Control Systems. Covert Attacks. System Identification. Countermeasures.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

BSA	Backtracking Search Optimization Algorithm
CPI	Cyber-physical Intelligence
DC	Direct Current
DES-CBC	Data Encryption Standard - Cipher Block Chaining mode
DMZ	Demilitarized Zone
DoS	Denial-of-Service
IP	Internet Protocol
IT	Information Technology
LAN	Local Area Network
LTI	Linear time-invariant
MitM	Man-in-the-Middle
MOPSO	Multi-objective Particle Swarm Optimization
MTBF	Mean Time Between Failure
NBCS	Network-Based Control System
NCS	Networked Control System
NII	Noise Impulse Integration
OSI	Open System Interconnection
OT	Operational Technology
PDF	Probability Density Function
PHWR	Pressurized Heavy Water Reactor
PI	Proportional-Integral
PID	Proportional-Integral-Derivative
PLC	Programmable Logic Controller

PSO	Particle Swarm Optimization
RPI	Radar Pulse Integration
RSA	Rivest-Shamir-Adleman cryptosystem
RTE	Real-Time Ethernet
SCADA	Supervisory Control and Data Acquisition
SD	Service Degradation
SISO	Single-Input-Single-Output
SLS	Switched Linear System
SNR	Signal-to-Noise Ratio
UDP	User Datagram Protocol
WNCS	Wireless Networked Control System

CONTENTS

1 INTRODUCTION

The integration of systems used to control physical processes via communication networks aims to assign to such systems better operational and management capabilities, as well as reduce its costs (GUPTA; CHOW, 2008). Motivated by these advantages, there is a trend to have an increasing number of industrial process and critical infrastructure systems driven by Networked Control Systems (NCS) (FAROOQUI et al., 2014; GUPTA; CHOW, 2008; GUPTA; CHOW, 2010; TIPSUWAN; CHOW; VANIJJIRATTIKHAN, 2003; ZHANG et al., 2013), also referred to as Network-Based Control Systems (NBCS) (CHOW; TIPSUWAN, 2001; LONG; WU; HUNG, 2005). As detailed in Figure 1, an NCS consists of a controller, which runs a control function $C(z)$, a physical plant, described by its transfer function $G(z)$, and a communication network that interconnects both devices through a forward stream and a feedback stream. The forward stream connects the controller output to the plant actuators. The feedback stream connects the output of the plant's sensors to the controller input.

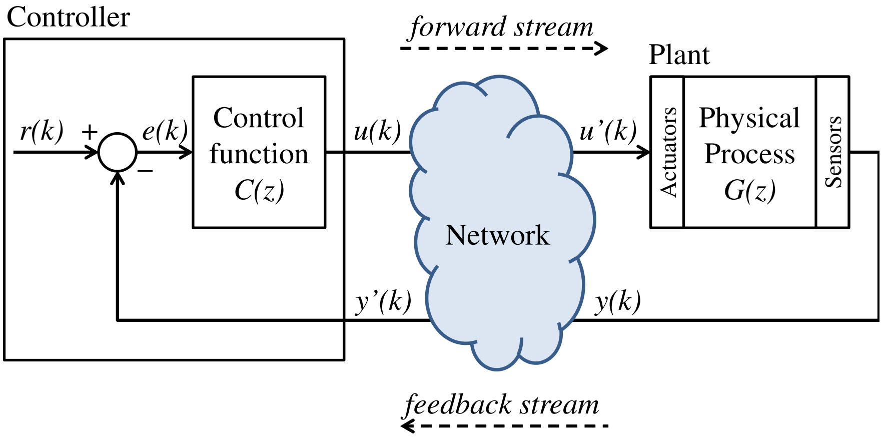


Figure 1 – Networked Control Systems (NCS) – own figure published in (SA; CARMO; MACHADO, 2017c).

At the same time it brings several advantages, the integration of controllers and physical plants in a closed loop through a communication network also exposes such control systems to new threats, typical of the cyber domain. Indeed, the literature (MCLAUGHLIN et al., 2016) reports the execution of real cyber attacks against physical plants since 1982, affecting a wide variety of targets, such as a diesel generator, a gas pipeline, and a steel plant. Among these known cases, the most emblematic example of attack in a cyber-physical system is the Stuxnet worm (LANGNER, 2011), whose strategic purpose was to deny nuclear weapons to Iran (ZETTER, 2014). Specifically, the targets were the uranium enrichment centrifuges installed at Natanz nuclear plant, which were controlled by a Supervisory Control and Data Acquisition (SCADA) system built with Siemens STEP7 programmable logic controllers (PLC) (FALLIERE; MURCHU; CHIEN, 2011; LANGNER, 2011; ZETTER, 2014). To reach the PLCs, which were in an isolated control network, the Stuxnet infected the Windows computers used by programmers to configure the PLCs. Once the infected computer was connected to the target controller, the Stuxnet installed a modified control algorithm into the PLC. The modified algorithm was specially designed to cause subtle and harmful behaviors to the centrifuges, reducing their efficiency and causing damage.

Note that, one possible way to attack an NCS, for example, is by hacking its software (*i.e.* changing the configuration or even the code executed by the controller), following a strategy similar to that used by the Stuxnet worm (LANGNER, 2011). Another possible way for an attacker to negatively affect an NCS is by interfering on its communication process. Basically, an attacker may interfere in the forward and/or feedback streams by three different means: inducing jitter, causing data loss due to packet drop outs, or even injecting false data in the communication process due to failure or absence of security mechanisms in the NCS.

In fact, although some new industrial communication protocols were developed including security features (FERRARI et al., 2013; MULLER; NETTO; PEREIRA, 2011; PETERSEN; CARLSEN, 2011), there are protocols in industry that still lack security mechanisms (COLLANTES; PADILLA, 2015) – such as the Profinet, MODBUS/TCP, and Ethernet/IP. The main issue of these industrial protocols is the lack of encryption and authentication (COLLANTES; PADILLA, 2015) between devices used in automation and control systems. A wide collection of scientific literature about this topic is available, reporting security breaches in all the major Real-Time Ethernet (RTE) protocols used in industry (PESCHKE et al., 2006; GRANAT; HÖFKEN; SCHUBA, 2017; AKPINAR; OZCELIK, 2018; YUNG; DEBAR; GRANBOULAN, 2016; MATHUR; TIPPENHAUER, 2016; PFRANG; MEIER, 2017; AKERBERG; BJORKMAN, 2009; COLLANTES; PADILLA, 2015). Still, even when the NCS uses secure communication protocols, it must be considered the possibility of the security mechanisms being overcome. The security of the NCS communication may be compromised if an attacker, for instance, succeed in obtaining security keys or passwords (used for encryption and authentication) through social engineering attacks (KROMBHOLZ et al., 2015).

Given the feasibility of occurring cyber attacks against physical systems, as evidenced by the real cases already reported in the literature (MCLAUGHLIN et al., 2016; LANGNER, 2011; ZETTER, 2014), studies have been conduced aiming to characterize vulnerabilities and propose security solutions for NCSs. According to (TEIXEIRA et al., 2015), the attacks in NCSs, in general, can be analyzed based on three aspects: the attacker's *a priori* system model knowledge; its disclosure resources; and its disruption resources. Regarding the requirement of model knowledge, from the point of view of control theory, the literature indicates that covert/model-based attacks must be planned based on an accurate knowledge about the NCS models (AMIN et al., 2013a; SMITH, 2011; SMITH, 2015; TEIXEIRA et al., 2015). However, despite the importance of model knowledge for this set of covert/model-based offensives, the literature does not explore attacks intended to reveal NCS models.

To fill this gap, in this work, two System Identification attacks are proposed, studied and evaluated as an attack tool to support the design of covert/model-based attacks. These attacks are: the Passive System Identification attack (SA; CARMO; MACHADO, 2017c); and the Active System Identification attack (SA; CARMO; MACHADO, 2017b). The system identification process, *i.e.* the action of building mathematical models of dynamic systems, is often used to obtain the model of physical processes aiming to support the design of their respective control systems. However, as demonstrated in this work, the system identification process can also be considered a key step for the execution of covert/model-based attacks against NCSs.

It is worth mentioning that the System Identification attacks herein proposed are different from the passive and active attacks performed to identify vulnerabilities of protocols and applications within the OSI model layers, such as the active scanning process used to identify network services (BOU-HARB; DEBBABI; ASSI, 2014). The attacks herein proposed aim to identify the physical model of a plant and the control functions that, in an NCS, lies above the application layer of the OSI model.

In addition to the proposed System Identification attacks, this thesis also presents three model-based offensives and evaluate their performances when supported by the referred System Identification attacks:

- The novel Controlled Data Loss attack, which is proposed in the present work;
- The Controlled Data Injection attack, which was characterized by this research in (SA; CARMO; MACHADO, 2017c);
- The Covert Misappropriation attack proposed in (SMITH, 2011; SMITH, 2015);

To support the discussion regarding the relationship between System Identification attacks and covert/model-based attacks, this work required the formalization of a number of concepts related to covertness and intelligence in the context of cyber-physical security. Thus, an additional contribution of this work is the proposition of a terminology that encompasses a whole new class of attacks in cyber-physical systems.

The analysis of system identification processes as feasible attacks led to the development of a countermeasure intended to inhibit the identification task, in case of failure of other conventional security mechanisms – such as encryption, network segmentation and firewall policies. In this sense, another contribution of this work is the proposal of a switching controller design (SA; CARMO; MACHADO, 2018) to hinder the System Identification attacks proposed in this work – and, therefore, dissuade covert/model-based attacks.

Finally, this work also proposes an identification strategy to estimate linear time-invariant (LTI) functions executed during controlled data injection attacks in NCSs. It consists of a link monitoring strategy, which uses white gaussian noise to excite possible attack functions in the NCS, in order to obtain the information necessary to identify the attack. To increase the accuracy of the attack function identification using white gaussian noise, this work also proposes a Noise Impulse Integration (NII) technique, which is developed inspired by the pulse integration process of radar systems (SKOLNIK, 1990).

It is worth emphasizing that the applications of NCSs can range from cooperative control of vehicles using mobile networks (ÖNCÜ et al., 2014; SABĂU et al., 2017) to wired NCS intended to control devices in Industry 4.0 (JAZDI, 2014; LASI et al., 2014), water canal systems (AMIN et al., 2013a; SMITH, 2015) or even large Pressurized Heavy Water Reactors (PHWR) (DASGUPTA et al., 2013). It includes a vast number of potential – sometimes critical – targets that can suffer from the attacks herein studied, as well as benefit from the countermeasures herein proposed.

1.1 OBJECTIVES

The first objective of this work is to study covert/model-based attacks in NCSs. The second objective of this work is to investigate disclosure attacks – particularly system identification attacks – as a tool to gather information from the plant and control algorithms, as well as the role of these attacks in the design of covert/model-based offensives against NCSs. With the lessons learned from the first two objectives, the third objective of this work is to develop countermeasures to mitigate system identification attacks and model-based offensives in NCSs, while ensuring adequate levels of plant control. It is worth mentioning that the aim of this work is not to facilitate System Identification attacks or covert/model-based offensives in NCSs.

1.2 CONTRIBUTIONS

So far, this research has resulted in contributions that encompass novel System Identification attacks and model-based offensives in NCS, as well as propose countermeasures to improve the security of NCSs against these kinds of attack. In summary, the main contributions of this work are listed below:

I - The proposition of a Passive System Identification attack to support the design of covert/model based attacks against NCSs;

II - The proposition of an Active System Identification attack, which is an alternative to the Passive System Identification attack when the attacker cannot wait for the occurrence of an event that produces the signals needed for the identification process;

III - The proposition of a novel controlled data loss attack, which is built upon the models learned through a System Identification attack. Based on the NCS models, this attack causes the loss of specific network packets to induce harmful behaviors to a plant. The attack uses a bio-inspired metaheuristic to smartly decide which packets the NCS must lose – through malicious interferences – in order to cause the desired effect on the plant;

IV - The evaluation on the effectiveness and accuracy of the joint operation of System Identification attacks and model-based offensives against NCSs, which is done considering an example of a common industrial device – a DC motor – and a nuclear critical infrastructure – a large Pressurized Heavy Water Reactor (PHWR);

V - The introduction of a taxonomy to support the discussion regarding the relationship between System Identification attacks and covert/model-based attacks in NCSs. This taxonomy also sets the requirements for the attacks discussed in this work, which helps on the development of layered defense strategies against System Identification attacks and covert/model-based offensives. Moreover, regarding covert attacks in cyber-physical systems – such as an NCS –, this taxonomy also dismembers the concept of covertness in two different domains: the cyber domain; and the physical domain.

VI - The proposition of a switching controller design to mitigate the proposed Passive and Active System Identification attacks – and, therefore, discourage the implementation of covert/model-based attacks –, while providing an adequate control performance;

VII - The proposition of a countermeasure to detect/identify linear time-invariant (LTI) functions executed by controlled data injection attacks in NCSs. To increase the accuracy of this countermeasure, it is proposed the *Noise Impulse integration* technique, which was developed using the radar pulse integration technique as inspiration.

1.3 ORGANIZATION OF THIS WORK

The rest of this work is organized as follows:

- First, Chapter 2 presents the related works and introduces a taxonomy regarding cyber-physical attacks that may happen in the control loop of an NCS.
- After that, Chapter 3 introduces two System Identification attacks, and then describes three model-based offensives in NCSs. The two System Identification attacks are the Passive System Identification attack and the Active System Identification attack. The three model-based offensives comprehend: a Controlled Data Loss attack; a Controlled Data Injection attack; and a Covert Misappropriation attack.
- Chapter 4 proposes two countermeasures: the switching controller strategy to mitigate the System Identification attacks described in Chapter 3; and the System Identification scheme to identify possible LTI attack functions executed by controlled data injection attacks in the NCS links.
- Chapter 5 presents results obtained through the joint operation of the System Identification attacks and the model-based offensives described in Chapter 3.
- Chapter 6 evaluates the performance of the countermeasures proposed in Chapter 4;
- Finally, Chapter 7 presents the conclusions and directions for possible future works.

A diagram representing simplified roadmap to the thesis is shown in Figure 2.

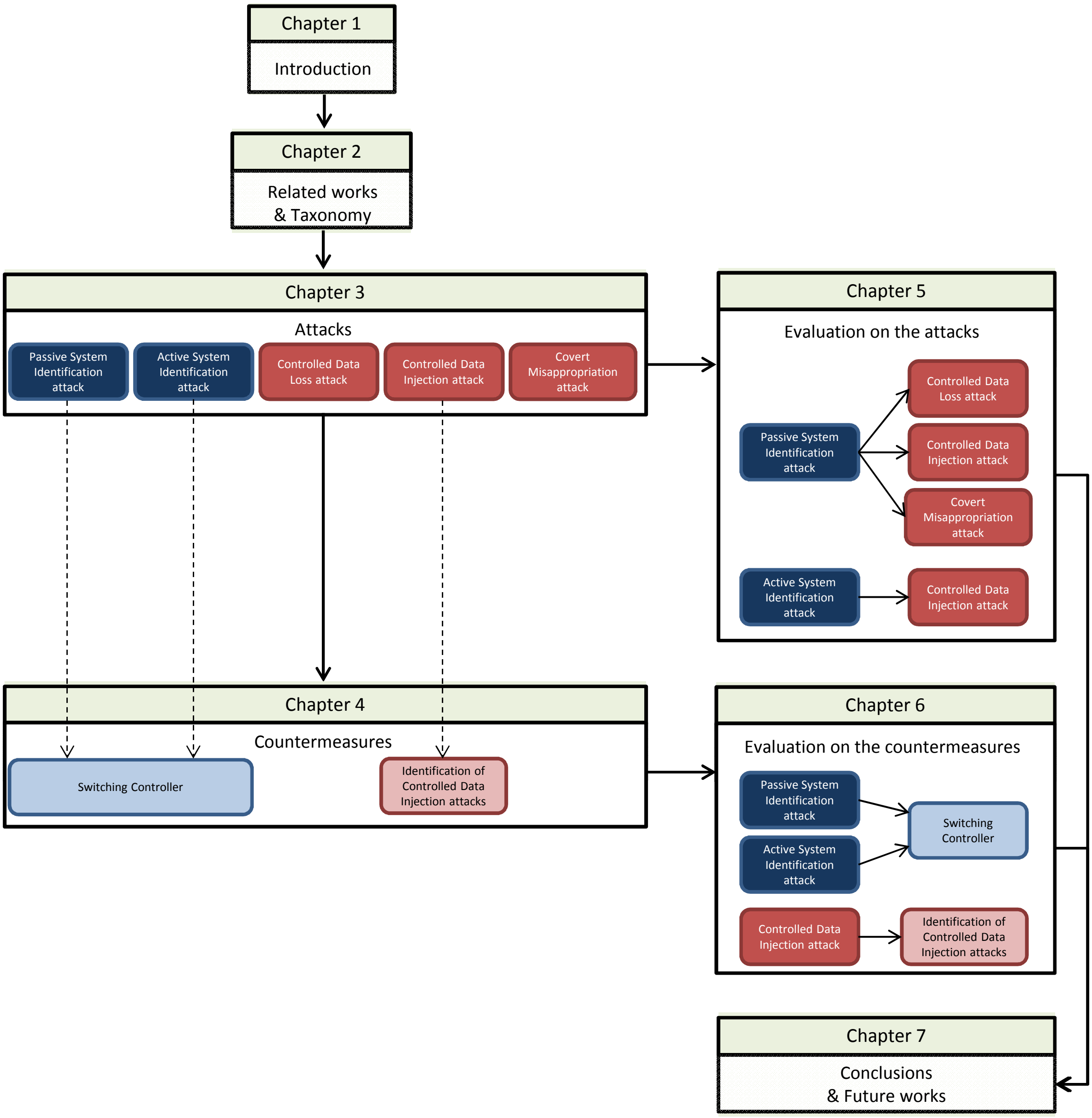

Chapter 1
Introduction
Chapter 2
Related works & Taxonomy
Chapter 3
Attacks
Passive System Identification attack
Active System Identification attack
Controlled Data Loss attack
Controlled Data Injection attack
Covert Misappropriation attack
Chapter 5
Evaluation on the attacks
Passive System Identification attack
Controlled Data Loss attack
Controlled Data Injection attack
Covert Misappropriation attack
Active System Identification attack
Controlled Data Injection attack
Chapter 4
Countermeasures
Switching Controller
Identification of Controlled Data Injection attacks
Chapter 6
Evaluation on the countermeasures
Passive System Identification attack
Active System Identification attack
Switching Controller
Controlled Data Injection attack
Identification of Controlled Data Injection attacks
Chapter 7
Conclusions & Future works

Figure 2 – Simplified roadmap to the thesis.

2 TAXONOMY AND RELATED WORKS

The present chapter aims to introduce a taxonomy developed in this thesis to support the discussion on System Identification attacks and model-based offensives in NCSs, as well as present the works related to this research. The proposed taxonomy is described in Section 2.1. The related works are presented in Section 2.2.

2.1 TAXONOMY

This work motivated the formalization of a number of concepts related to covertness and intelligence in the context of cyber-physical security. Thus, the first contribution of this work is the proposition of a terminology that encompasses a whole new class of attacks on cyber-physical systems in general, and on NCSs in particular. This taxonomy was introduced in (SA; CARMO; MACHADO, 2017c), in order to depict the role that System Identification attacks play in the development of covert/model-based attacks in NCSs. The referred taxonomy was then revised in (SA; CARMO; MACHADO, 2017b), in order to divide the System Identification attacks into two categories, namely: Passive System Identification attacks; and Active System Identification attacks. The proposed taxonomy also establishes a new approach regarding to the covertness of attacks on cyber-physical systems, which must be analyzed from two aspects simultaneously: physical; and cybernetic.

This section presents a taxonomy that enfolds all concepts covered in (SA; CARMO; MACHADO, 2017b; SA; CARMO; MACHADO, 2017c). The taxonomy herein presented establishes a basis for the discussions made in this work, regarding attacks on NCSs and possible countermeasures. In Section 2.1.1, the attacks on NCSs are briefly described and classified according to the way they act in the system. In Section 2.1.2, it is introduced a new approach for the analysis of covert attacks in cyber-physichal systems.

2.1.1 Classification of the Attacks

The attacks to cyber-physichal control systems may take place on its devices – *i.e.* the controller, and the plant's sensors and actuators – and/or on its communication system, affecting the forward and feedback streams. As a premise, we must consider that the *service* intended to be attacked/protected in such system is the work performed by the physical process controlled by the NCS.

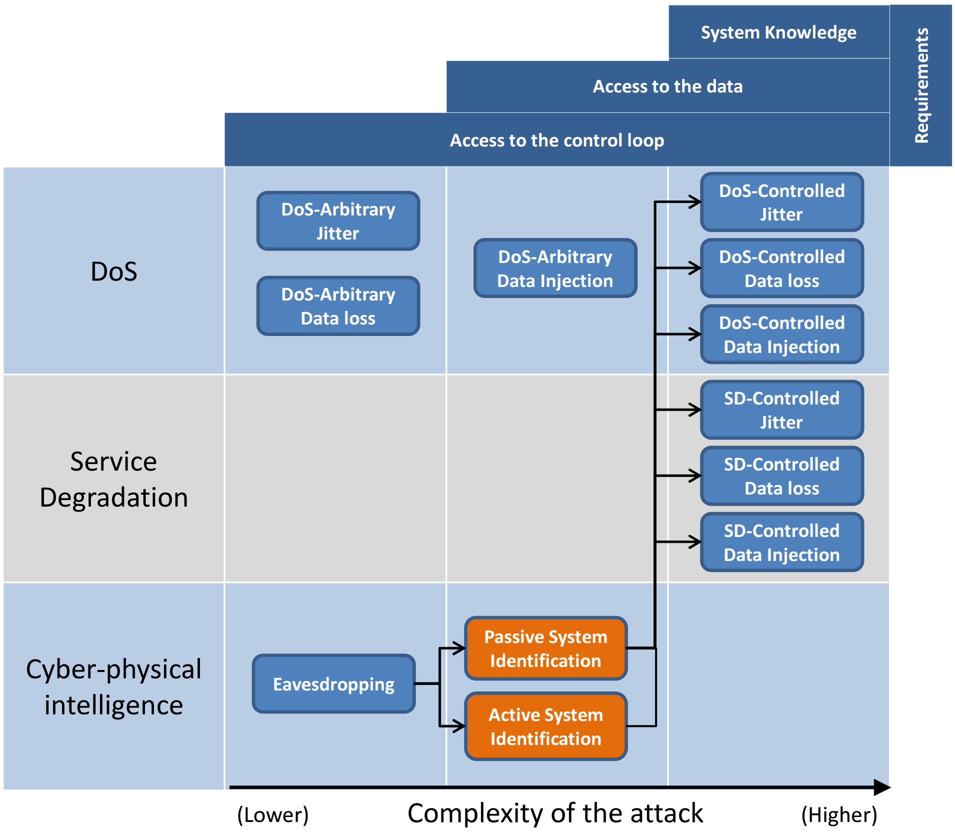


Figure 3 – Classification and requirements of cyber-physical attacks that act in the control loop of an NCS – adapted from own figure published in (SA; CARMO; MACHADO, 2017c).

Considering the aforementioned definition of service in an NCS, the attacks may be classified within three different categories, as shown in Figure 3:

- Denial-of-Service (DoS) (HUSSAIN; HEIDEMANN; PAPADOPOULOS, 2003): in an NCS, the DoS attacks comprehends all kind of cyber-physical attacks that deny the physical process operation, interrupting the execution of the work, or service, that the controlled plant is intended to do. The attack results, for example, in behaviors that may shut the plant down or even destroy it in a short therm.

- Service Degradation (SD) (SA; CARMO; MACHADO, 2017c): the SD attacks consist of malicious interventions that are done in the control loop in order to reduce the service efficiency, *i.e.* the efficiency of the physical process, or even reduce the mean time between failure (MTBF) of the plant in mid therm or long therm.

- Cyber-physical Intelligence (CPI) (SA; CARMO; MACHADO, 2017c): the concept of Cyber-physical Intelligence, herein proposed, is different from the concept where cyber-physical systems are integrated with intelligent systems (RAMOS; VALE;

FARIA, 2011). In the present taxonomy, CPI attacks comprehend actions that are performed in the control loop of an NCS in order to gather information about the system's operation and/or its design. This attacks are intended to acquire the intelligence necessary to plan covert and model-based attacks, or even to provide data for replay attacks (LANGNER, 2011).

In Figure 3, six kinds of DoS attacks are presented, with their respective requirements. From these six DoS attacks, the less complex are the three arbitrary ones:

- DoS-Arbitrary Jitter: in this kind of attack, the delay of the forward and/or feedback stream is arbitrarily changed, without a previous knowledge about the NCS models, in order to lead the system to an instability or to a condition that causes the physical process interruption. This attack only requires access to the control loop, once it may be performed by just consuming the system resources, such as the bandwidth of communication links or the computational resources of the equipments that are in the control loop.

- DoS-Arbitrary Data Loss: in this kind of attack, the attacker prevents data from reaching the actuator and/or the controller of the NCS. The communication channel is arbitrarily jammed, without a previous knowledge about the NCS models, leading the system to an instability or to a condition that causes the physical process interruption. It is worth mentioning that some DoS-Arbitrary Jitter attack may result in a DoS-Arbitrary Data Loss attack, if an eventual higher delay cause packet drop outs. As the DoS-Arbitrary Jitter attack, this attack only requires access to the NCS control loop.

- DoS-Arbitrary Data Injection: in such attacks, the attacker sends arbitrary false data to the controller, as it was sent by the sensors, and/or to the actuators, as it was sent by the controller. The false data is injected in the NCS closed loop without a previous knowledge about the NCS models. This attack is more complex than the DoS-Arbitrary Jitter and DoS-Arbitrary Data Loss attacks, given that it requires access to the data that flows in the NCS control loop.

The attacks classified as DoS-controlled – DoS-Controlled Jitter, DoS-Controlled Data Loss, and DoS-Controlled Data Injection – shown in Figure 3 interfere in the control loop of an NCS by the same means that their respective DoS-Arbitrary attacks. The difference between a DoS-Controlled attack and a DoS-Arbitrary attack is that, in the former, the interference caused by the attacker is precisely planned and executed, in order to achieve the exact desired behavior that leads the physical service to an interruption, in a more efficient way. Thus, to achieve such efficiency, a DoS-Controlled attack requires an accurate knowledge about the NCS models, *i.e.* the plant and controller transfer functions, which have to be analyzed to plan the attack.

Regarding to the SD attacks, we must consider the three different kinds of attack shown in Figure 3: SD-Controlled Jitter, SD-Controlled Data Loss, and SD-Controlled Data Injection. The difference between an SD-Controlled attack and a DoS-Controlled attack is that the former is not intended to interrupt the physical process in a short therm. It aims to keep the process running with reduced efficiency, sometimes also targeting a gradual physical deterioration of the controlled devices. To succeed, the SD-Controlled attacks need to be planned based on an accurate knowledge about the dynamics and the design of the NCS. Otherwise, the attack can eventually interrupt the physical process, due to unpredicted reasons, evolving to a DoS attack.

The system knowledge required to both DoS-Controlled and SD-Controlled attacks, can be gathered through CPI attacks, as shown in Figure 3. The first and simpler CPI attack is the eavesdropping attack (KHATRI et al., 2015; ZOU; WANG, 2016), which consists of simply capturing the data transmitted through the forward and feedback streams of the NCS. In addition to the eavesdropping attack, the CPI attacks also include two kinds of System Identification attacks. These System Identification attacks were proposed in (SA; CARMO; MACHADO, 2017c; SA; CARMO; MACHADO, 2017b), as a result of the present research, and their concepts are described bellow:

- Passive System Identification attack (SA; CARMO; MACHADO, 2017c): this kind of attack estimates the model of an NCS based on the analysis of the signals collected from the input and output of the system's devices. This kind of attack analyzes signals that typically flow through the NCS, as a result of its normal operation. In this case, both input and output signals must carry meaningful information – *i.e.* information enough to estimate the transfer function of the attacked system/device –, and it is not necessary to inject signals into the attacked system.
- Active System Identification attack (SA; CARMO; MACHADO, 2017b): in this kind of attack, the attacker injects an attack signal into the system, in order to estimate the NCS model based on the system response to such signal. From the attacker point of view, this attack is useful, for example, when the system is in steady state and the attacker cannot wait for a signal carrying the meaningful information required for the identification process.

It is noteworthy that an Active System Identification attack is less stealthy than a Passive System Identification attack, given that the former needs to interfere in the system and the latter just needs to listen its signals. In this sense, when performing an Active System Identification attack, the attacker must choose signals that, when injected on the NCS, are more difficult to be perceived by a defense system. From the defender perspective, it is important to be aware of this kind of attack and also learn about the stealthiness of Active System Identification attacks, in order to develop techniques to identify and avoid them.

2.1.2 Cybernetic vs. Physical Covertness

The covertness of an attack regards to its capacity to not be perceived or detected. In the case of cyber-physical attacks on NCSs, the covertness must be simultaneously analyzed in two different domains: the cyber domain; and the physical domain. In this sense, it is presented in this section the definition proposed in (SA; CARMO; MACHADO, 2017c) about what is a *cybernetically covert* attack and what is a *physically covert* attack:

- Cybernetically covert attacks: are the attacks that have low probability to be detected by algorithms that monitor the softwares, communications and data of the NCS, or by systems that monitor the plant dynamics.
- Physically covert attacks: are attacks that cause physical effects that can not be easily noticed or identified by a human observer. The attack slightly modifies some behaviors of the system in a way that it physically affects the plant, but the effect is not easily perceptible or it can eventually be understood as a consequence of another root cause, other than an attack.

The taxonomy available in the literature before the approach provided by this research – published in (SA; CARMO; MACHADO, 2017c) – does not clearly distinguish that an attack may have different degrees of covertness regarding to the cybernetic and physical domains. However, analyzing cyber-physical attacks, it is possible to state that the measures taken to make an attack cybernetically covert do not necessarily guarantee a physically covert behavior, and vice versa. Thus, in order to provide a clear comprehension about these two aspects of covertness of a cyber-physical attack, this research introduces the two aforementioned classifications for covertness. For instance, in (SMITH, 2011; SMITH, 2015), it is proposed an attack architecture, where the attacker eliminates from the feedback signal the interference caused by him on a plant through data injection. That architecture hinders the system's ability to detect the attack through signal analysis, making it cybernetically covert. However, such architecture does not guarantee that the physical effects of the attack will not facilitate its disclosure. Indeed, depending on the plant's behavior, the attack can provide physical evidences that it is being manipulated, drawing the attention for the possibility of a cyber-physichal attack. Thus, to be physically covert, the attacker's control function of (SMITH, 2011; SMITH, 2015) have to be adjusted to meet the requirements of a physically covert attack, as herein defined, independently of the cybernetic covertness provided by the attack architecture.

2.2 RELATED WORKS

This section discusses the works related to the subject of this research. Section 2.2.1 regards to cyber attacks in NCSs, situating the attacks herein proposed in the existing literature. Section 2.2.2, in turn, presents works regarding countermeasures for cyberattacks in NCSs, as well as indicates how the countermeasures herein proposed complement the existing security solutions.

2.2.1 Attacks in NCSs

The launch of cyber-physical attacks in real world systems (MCLAUGHLIN et al., 2016), such as the case of Stuxnet (LANGNER, 2011) worm, raised the concern of governments and NCS owners, and is motivating the research on cybersecurity of industrial facilities and critical infrastructures. In this context, recent studies demonstrate the development of a set of sophisticated attacks that, to achieve a high level of covertness and accuracy, rely on the knowledge about the model of the attacked system. Therefore, this section presents a review on cybersecurity of NCSs, giving special attention to covert/model-based attacks, as well as their inherent need for accurate NCS models.

In (LONG; WU; HUNG, 2005), the authors evaluate the impact of delay jitter and packet loss in an NCS under DoS attack. The conception of such DoS attack does not take into account the models of the controller and physical plant of the attacked NCS (*i.e.* these models are not known by the attacker). Therefore, to affect the physical process, the attacker arbitrarily floods the network, causing jitter and packet loss in the NCS communication links. In this tactic, the excess of packets in the network may reveal the attack, allowing the implementation of countermeasures such as packet filtering (LONG; WU; HUNG, 2005) or blocking of malicious traffic on its origin (SNOEREN et al., 2002). Additionally, as stated in (SA; CARMO; MACHADO, 2017c), an arbitrary intervention in a system which the models are unknown may lead the plant to an extreme physical behavior, which is not desired if a physically covert (SA; CARMO; MACHADO, 2017c) attack is intended.

In (FAROOQUI et al., 2014), the authors demonstrate an attack where false signals are transmitted to the controller and actuators of an NCS. The false signals are randomly generated by the attacker, aiming to cause instability on the plant (a DC motor). To evaluate this arbitrary data injection attack, the authors propose a testbed for SCADA systems, using TrueTime (a MATLAB/Simulink based tool). Such arbitrary data injection attack does not require a previous knowledge about the models of the plant and its controller. Therefore, the desired physical effect and the attack covertness cannot be ensured due to the unpredictable consequences of the injection of random false signals in a system whose model is not known.

In (TEIXEIRA et al., 2015), the authors analyze a wide variety of attacks in NCSs and establish requirements for those attacks in terms of model knowledge, disclosure and disruption resources. In their work, it is stated that the design of covert attacks requires high level of knowledge about the attacked system model. In (SMITH, 2011; AMIN et al., 2013a; SMITH, 2015), examples of covert attacks that agree with the statement provided in (TEIXEIRA et al., 2015) are proposed and analyzed. In (SMITH, 2011; SMITH, 2015), the attacker, acting as a man-in-the-middle (MitM), injects false data in the NCS forward stream to take control of the plant. Then, to make the attack covert, the attacker uses the attacked plant model to compute the data injected in the feedback stream. The covertness of the attack proposed in (SMITH, 2011) is analyzed from the perspective of signals arriving at the controller and, as demonstrated in (SMITH, 2015), it depends on the difference between the actual plant model and the model known by the attacker. In (AMIN et al., 2013a), the attacker, aware of the NCS model, injects data in its communication links to covertly steal water from the Gignac canal system located in Southern France.

In (AMIN et al., 2013a; SMITH, 2011; SMITH, 2015; TEIXEIRA et al., 2015), although the attacks are designed based on the NCS models, the authors do not describe how these models are obtained by the attacker. It is just stated that the models, used to design covert/model-based attacks, are previously known by the attacker. In order to fill this gap, this work proposes two new kinds of attack to estimate the models of the attacked system: the Passive System Identification attack (SA; CARMO; MACHADO, 2017c); and the Active System Identification attack (SA; CARMO; MACHADO, 2017a). According to the taxonomy proposed in section 2.1 – and published in (SA; CARMO; MACHADO, 2017c) –, these attacks belong to the category of Cyber-physical Intelligence attacks.

The Passive System Identification attack (SA; CARMO; MACHADO, 2017c) – formerly referred to as System Identification attack [1] – does not need to inject signals in the NCS to estimate its models. However, the effectiveness of the Passive System Identification attack depends on the occurrence of events – not controlled by the attacker – to produce signals that carry meaningful information for the system identification algorithm. This attack passively estimates the transfer functions of both controller and

[1] The Passive System Identification attack was originally referred in (SA; CARMO; MACHADO, 2017c) as System Identification attack. However, with the introduction of the Active System Identification attack in (SA; CARMO; MACHADO, 2017a), its designation was reviewed to Passive System Identification attack, in order to evince the differences between the two attacks.

plant by simply eavesdropping the forward and feedback streams of the system. On the other hand, the Active System Identification attack (SA; CARMO; MACHADO, 2017a) constitutes an alternative to the Passive System Identification attack in situations where the attacker cannot wait so long for the occurrence of such meaningful signals. In the Active System Identification attack, the attacker estimates the open-loop transfer function of the NCS by injecting an attack signal and eavesdropping its response at a single point of interception.

In this work, the aforementioned System Identification attacks are evaluated in joint operations with the following model-based offensives:

- The SD-Controlled Data Injection attack, characterized in this research (SA; CARMO; MACHADO, 2017c);
- The Covert Misappropriation attack proposed in (SMITH, 2011; SMITH, 2015);
- The novel SD-Controlled Data Loss attack.

The SD-Controlled Data Loss attack, proposed in this work, demonstrates the ability to produce the same accurate and harmful behaviors of the SD-Controlled Data Injection attack (SA; CARMO; MACHADO, 2017c), however, without the need to overcome eventual security mechanisms for data integrity and authenticity that may hinder an SD-Controlled Data Injection attack. Moreover, in contrast with the arbitrary data loss attack shown in (LONG; WU; HUNG, 2005), the SD-Controlled Data Loss attack takes special care to avoid the indiscriminate loss of samples, as well as the complete denial of communication. The SD-Controlled Data Injection attack uses the models estimated by a System Identification attack to smartly cause the loss of a limited number of specific samples, which makes the attack more difficult to be noticed.

A synthesis of the attacks addressed in this section is presented in Table 1. Based on these works, it is possible to verify how the proposed System Identification attacks and model-based offensives are included in the scenario of attacks in NCSs, as well as how they contribute for the study of cybersecurity of NCSs. It is worth mentioning that, given the recent inclusion of the System Identification attacks in the context of NCSs cybersecurity – which was done through this research –, as far was we know, the scientific literature does not present specific countermeasures to mitigate the identification process performed by the proposed attacks.

Table 1 – Synthesis of related attacks

Attack	Method	Knowledge about the system?	How the knowledge is obtained?
Stuxnet *worm* (LANGNER, 2011)	Modifications in PLC code	Yes	Experiments in a real system
Long, *et al.* (LONG; WU; HUNG, 2005)	*Jitter* and packet loss	None	N/A
Farooqui, *et al.* (FAROOQUI et al., 2014)	Data injection	None	N/A
Smith (SMITH, 2011; SMITH, 2015)	Data injection	Yes	Not described
Teixeira (TEIXEIRA et al., 2015)	Packet loss Data injection	None Yes	N/A Not described
Amin (AMIN et al., 2013a)	Data injection	Yes	Not described
SD-Controlled (SA; CARMO; MACHADO, 2017c)	Data injection	Yes	Passive System Identification attack
SD-Controlled (SA; CARMO; MACHADO, 2017b)	Data injection	Yes	Active System Identification attack
SD-Controlled	Data Loss	Yes	Passive System Identification attack

2.2.2 Countermeasures in NCSs

As discussed in Chapter 1, the occurrence of cyber attacks against real-world cyber-physical systems (MCLAUGHLIN et al., 2016) evince the feasibility of launching actual attacks in NCSs. At the same time, the literature on NCS (AMIN et al., 2013a; AMIN et al., 2013b; PANG; LIU, 2012; WANG; LU, 2013) brings theoretical studies and practical experiments that demonstrate the efforts to propose countermeasures for cyberattacks in NCSs. This section presents works reporting security solutions for NCSs, in order to indicate how the countermeasures proposed in this work contribute for the security of these systems – specially against the System Identification attacks and model-based offensives addressed in this thesis.

The straightforward countermeasure to prevent cyberattacks in NCSs (including the System Identification attacks and model-based offensives discussed in Sections 2.1 and 2.2.1) is to avoid unauthorized access to the system control loop. It can be achieved by using, for instance, network segmentation, demilitarized zones (DMZ), firewall policies and implementing specific network architectures, such as recommended in (STOUFFER et al., 2015).

As proposed in (PANG; LIU, 2012), a complementary countermeasure – in case the attacker is able to access the control loop – is to hinder the access to the data flowing in the NCS using, for example, symmetric-key encryption algorithms, hash algorithms and a timestamp strategy to form a secure transmission mechanism between controller and plant. In (WANG; LU, 2013), the use of cyber-security mechanisms in devices endowed with limited computational resources – such as actuators and sensors of an NCS – is quantitatively evaluated through experiments using the communication module TS7250 (200-MHz ARM9 CPU and 32-MB SD-RAM). The results of (WANG; LU, 2013) indicate that a DES-CBC encryption requires $183.81ms$ of processing time, while a RSA encryption requires $228.18ms$ to encrypt the same amount of data from a solid-state transformer, using a 1024-bit key. If a 2048-bit key is used, the processing time of RSA, for example, grows to $1457.14ms$. This may be an issue if it is considered an NCS sensitive to delay. Such processing times exemplify the tradeoff between security and performance, which must be taken into account when dealing with NCS where real-time communication is normally required.

In (AMIN et al., 2013a; AMIN et al., 2013b) the authors report field-operational test attacks, performed at the Gignac canal system (in Southern France), where the attacker pilfers water from the canal, without being noticed, by manipulating the data transmitted by a sensor. The authors indicate that, among all sensors of the attacked canal, there is a set of sensors that are more critical and should receive more investments on cyber-security mechanisms aiming more resilience to tampering. Examples of such cyber-security mechanisms are experimentally assessed in (PANG; LIU, 2012), where the authors propose a recursive networked predictive control (RNPC) technique, combined with a symmetric-key encryption algorithm, a times-tamp and a hash algorithm to ensure data confidentiality and integrity.

However, in spite of these security mechanisms, one must consider the possibility of the mentioned countermeasures fail and the attacker gain access to the data flowing in the NCS. Indeed, the access to the NCS data can be facilitated by alternative attack methods, such as social engineering attacks (KROMBHOLZ et al., 2015) to obtain passwords or encryption keys in use. In this case, specifically for the System Identification attacks and model-based offensives addressed in this work, an alternative to impair them is to prevent the attacker in obtaining the system model by hindering the analysis of the captured data – *i.e.* making the System Identification algorithm inaccurate/ineffective. Therefore, this work proposes a countermeasure to mitigate the mentioned System Identification attacks, in situations where the attacker gets access to the data that is transmitted through the NCS. This countermeasure consists of a switching controller design that hinders the identification process and, at the same time, allows a satisfactory plant control. It is worth emphasizing that this countermeasure is intended to increase the security of NCSs against System Identification attacks and model-based offensives, in case of failure or absence of conventional security mechanisms for data confidentiality, integrity and authenticity (PANG; LIU, 2012; WANG; LU, 2013; STOUFFER et al., 2015).

Additionally, the present work also proposes a countermeasure to detect/identify SD-Controlled Data Injection attacks, in case an attacker gathers all resources needed to implement it in an NCS. The countermeasure consists of a link monitoring strategy that uses white gaussian noise to excite the attack function an obtain the information necessary for the identification process. A Noise Impulse Integration (NII) technique is proposed to increase the accuracy of the countermeasure.

The literature (MO; SINOPOLI, 2009; MO et al., 2012; PASQUALETTI; DORFLER; BULLO, 2015) report countermeasures that analyze the NCS signals to detect/identify possible cyberattacks in the system control loop. A brief discussion on these countermeasures is presented below.

In (MO; SINOPOLI, 2009; MO et al., 2012; PASQUALETTI; DORFLER; BULLO, 2015) the authors report a countermeasure that make use of random Gaussian noise to detect replay attacks in NCSs. According to (PASQUALETTI; DORFLER; BULLO, 2015), a replay attack is carried out by hijacking the NCS sensors, recording the readings for a certain time, and repeating such readings while injecting a malicious signal into the system. In (MO; SINOPOLI, 2009; MO et al., 2012), it is shown that these replay attacks can be detected by injecting a random signal (unknown to the attacker) into the system. The random signal, in this case, acts as an authentication signal. Considering that the random signal is unknown to the attacker, the authors prove that the detector is able to verify whether the received signals derive from a replay attack or not. Note that the countermeasure proposed in (MO; SINOPOLI, 2009; MO et al., 2012) is restricted to the case of replay attacks, not being able to identify an SD-Controlled Data Injection attack. Moreover, it incurs a drawback: as stated by the authors, when the system is under normal operation, the controller is not optimal anymore, which means that in order to detect the attack, the solution needs to sacrifice control performance.

In (PASQUALETTI; DORFLER; BULLO, 2013; PASQUALETTI; DORFLER; BULLO, 2015) the authors address a general class of monitors that, to detect and identify additive attacks in cyberphysical systems, does not inject signals in the communication links. This class of monitors detect and identify attacks based on the presumed knowledge of the system dynamics and measurements. Let (2.1) represent the model of the attacked system:

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t). \end{aligned} \tag{2.1}$$

wherein $x : \mathbb{R} \to \mathbb{R}^n$ and $y : \mathbb{R} \to \mathbb{R}^p$ are the maps describing the evolution of the system state and measurements, respectively, and $E \in \mathbb{R}^{n\times n}$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$ are constant matrices that characterize the system. So, to perform the detection and identification tasks in (PASQUALETTI; DORFLER; BULLO, 2013; PASQUALETTI; DORFLER; BULLO, 2015), the input of the monitoring system is $\Lambda = \{E, A, C, y(t)\}$. By knowing $\{E, A, C\}$, the attack is detected and identified based on its effect in the output measurements $y(t)$. A drawback of this kind of monitor is that a fortuitous and involuntary incident modifying the actual system model (*i.e.*, modifying E, A or C) may affect the attack detection and identification processes. In this case, modifications in the physical domain that reflect changes in E , A or C can cause inconsistencies to the monitor, suggesting an attack and drawing the attention to the cyber domain even when the problem is not there.

Another approach to reveal data deception attacks in NCSs is presented in (TEIXEIRA, 2014). The author tackle the problem of detecting attack signals (added to the NCS links) by modifying the system's structure (2.1). Specifically, to reveal attacks, the method requires modifications in the system dynamics, inputs, and outputs, through changes in matrices A, B, and C, respectively. The results show that such strategy is effective, however, some considerations regarding practical constraints of the method are necessary. First, according to (TEIXEIRA, 2014), to reveal attacks by modifying matrix C, it is necessary to add measurement signals to the system, which consequently increases the network traffic. Then, the strategy to modify the system matrix A may not be convenient, or even feasible, given that it may imply structural changes in the physical process. Last, to modify the input matrix B, the authors propose a coordinated scaling of the control inputs between the controller and actuator. To do so, the authors define a new plant input matrix $\tilde{B} = BW$ and a new control signal $\tilde{u}(t) = W^{-1}u(t)$, where W is an invertible matrix unknown to the attacker. According to the authors, this scheme can be interpreted as a coding or encryption performed between the controller and actuator, where W acts as their secret key. The issue that arises from this scheme is that, when modifying W, an occasional lack of synchronism between the controller and actuators may actually scale the control signal applied to the plant, which may affect the system performance.

Regarding the aspects and characteristics of the monitoring systems discussed in this section, the countermeasure proposed in this work takes special care to avoid some of the reported issues when specifically detecting/identifying SD-Controlled Data Injection attacks:

1. The proposed countermeasure does not interfere in the NCS behavior when the system is in its normal operation. Without the attack the injected noise is canceled, manifesting only in the presence of an attack;

2. Ocasional changes in the physical process does not lead the monitoring system to inappropriately suggest a cyberattack. In the present work, to avoid this problem, the proposed solution injects a noisy signal in the NCS in such way that the noise only manifests itself when an SD-Controlled Data Injection attack is present in the monitored link. In this case, if the injected noise is present in measurements, it is possible to state that an SD-Controlled Data Injection attack is occurring. On the other hand, if the noise is cancelled, it is possible to state that there is no SD-Controlled Data Injection attack in the NCS link, regardless of whether the monitored system is physically modified or not. This way, it is possible to evaluate whether the problem is restricted to the cybernetic domain or not.

3. In the present work, the countermeasure specifically proposed to reveal SD-Controlled Data Injection attacks, does not require modifications in the plant structure, avoiding changes in the physical process, increase of network traffic, and risks associated with occasional lack of coordination when scaling the NCS signals, for instance.

3 SYSTEM IDENTIFICATION ATTACKS AND MODEL-BASED OFFENSIVES

This chapter discusses System Identification attacks and model-based offensives in NCSs, describing the underlying details on how they are built, as well as how they can work together to compose intelligent and sophisticated threats to cyber-physical systems. Section 3.1 presents the two System Identification attacks proposed in this work. Section 3.2 presents the three model-based offensives that can be built with the support of System Identification attacks.

3.1 SYSTEM IDENTIFICATION ATTACKS

This section describes the two system identification attacks defined in the taxonomy discussed in Section 2.1: the Passive System Identification attack; and the Active System Identification attack. These attacks aim NCSs constituted by impulse-response systems, defined by Linear Time Invariant (LTI) transfer functions, such as the NCSs presented in (TIPSUWAN; CHOW; VANIJJIRATTIKHAN, 2003; ZHANG et al., 2013; FAROOQUI et al., 2014; SMITH, 2011; PANG; LIU, 2012; DASGUPTA et al., 2013; AMIN et al., 2013a; SMITH, 2015). Exemples of potential targets with this characteristic can range from non-critical industrial plants controlled by wireless networked control systems (WNCS) (FERRARI et al., 2013; SADI; ERGEN; PARK, 2014), to large Pressurized Heavy Water Reactors (PHWR) (DAS et al., 2006; DASGUPTA et al., 2013; SÁ; CARMO; MACHADO, 2018) or water canal systems (AMIN et al., 2013a; SMITH, 2015) controlled by wired NCSs. The well known vulnerabilities of the cyber domain (UMA; PADMAVATHI, 2013; DRIAS; SERHROUCHNI; VOGEL, 2015; COLLANTES; PADILLA, 2015), which may allow an attacker to have access to the control loop of an NCS, and the typical model of the aforementioned cyber-physical systems, which are consistent with the attack herein proposed, evince why this attack may actually happen. Note that it includes targets with potentially significant impacts, such as a PHWR and water canal systems. Section 3.1.1 presents the details of the Passive System Identification attack, while Section 3.1.2 describes the Active System Identification attack.

3.1.1 Passive System Identification Attack

The Passive System Identification attack (SA; CARMO; MACHADO, 2017c) is intended to assess the coefficients of the plant's transfer function $G(z)$ and the controller's control function $C(z)$ of the generic NCS shown in Figure 1. Both functions are LTI. The attack uses the Backtracking Search Optimization Algorithm (BSA) – a metaheuristic proposed in (CIVICIOGLU, 2013) and briefly described in (SÁ; NEDJAH;

MOURELLE, 2016) – to minimize the fitness function presented in this section. The BSA is specifically chosen to demonstrate the feasibility of System Identification attacks on NCSs. Moreover, it is noteworthy that the use of BSA to perform a system identification process was not reported before in the literature, which constitutes another novelty of this work.

The BSA is an evolutionary algorithm that uses the information obtained by past generations – or iterations – to search for solutions for optimization problems. The algorithm has two parameters that are empirically adjusted: the size of its population P; and η, described in (SÁ; NEDJAH; MOURELLE, 2016), that establishes the amplitude of the movements of the individuals of P. The parameter η must be adjusted aiming to assign to the algorithm both good exploration and exploitation capabilities.

If both input $i(k)$ and output $o(k)$ of an NCS device are known, the model of such device can be assessed by applying the known $i(k)$ in an estimated model, which must be adjusted until its estimated output $\hat{o}(k)$ converge to $o(k)$. In this sense, the BSA is used to iteratively adjust the estimated model, by minimizing a specific fitness function, until the estimated model converge to the actual model of the real device, that can be either a controller or a plant.

To establish the fitness function, firstly, it must be considered a generic LTI system, whose transfer function $Q(z)$ is represented by (3.1):

$$Q(z) = \frac{O(z)}{I(z)} = \frac{a_n z^n + a_{n-1} z^{n-1} + ... + a_1 z^1 + a_0}{z^m + b_{m-1} z^{m-1} + ... + b_1 z^1 + b_0}, \tag{3.1}$$

wherein $I(z)$ is the system input, $O(z)$ is the system output, n and m are the order of the numerator and denominator, respectively, and $[a_n,a_{n-1},...a_1,a_0]$ and $[b_{m-1},b_{m-2},...b_1,b_0]$ are the coefficients of the numerator and denominator, respectively, that are intended to be found by this System Identification attack. Also, it must be considered that $i(k)$ and $o(k)$ represent the sampled input and output of the system, respectively, where $I(z) = \mathcal{Z}[i(k)]$, $O(z) = \mathcal{Z}[o(k)]$, k is the sample number and $\mathcal{Z}$ represents the Z-transform operation.

In this System Identification attack, $i(k)$ and $o(k)$ are firstly captured by an eavesdropping attack (KHATRI et al., 2015; ZOU; WANG, 2016), for exemple, during a monitoring period T. To deal with the eventual loss of samples, that may not be received by the attacker during T, the algorithm holds the value of the last received sample, according to (3.2), wherein $x(k)$ can either be $i(k)$ or $o(k)$:

$$x(k) = \begin{cases} x(k-1) & \text{if sample } k \text{ is lost;} \\ x(k) & \text{otherwise.} \end{cases} \tag{3.2}$$

Then, after acquiring $i(k)$ and $o(k)$, the captured $i(k)$ is applied to the input of an estimated model, that is described by a transfer function whose coefficients $[a_{n,j},a_{n-1,j},...a_{1,j},a_{0,j},b_{m-1,j},\ b_{m-2,j},...b_{1,j},b_{0,j}]$ are the coordinates of an individual j of the BSA. The application of $i(k)$ to the input of the estimated model results in an output signal $\hat{o}_j(k)$. After obtaining $\hat{o}_j(k)$, the fitness f_j of individual j is computed comparing the output $o(k)$, captured from the attacked device, with the output $\hat{o}_j(k)$ of the estimated model, according to (3.3):

$$f_j = \frac{\sum\limits_{k=0}^{N} (o(k) - \hat{o}_j(k))^2}{N}, \tag{3.3}$$

wherein N is the number of samples that exist during the monitoring period T. Note that, if the attacker does not lose any sample of $i(k)$ and $o(k)$ during T, then $\min f_j = 0$ when $[a_{n,j},a_{n-1,j},...a_{1,j},a_{0,j},b_{m-1,j},b_{m-2,j},...b_{1,j},b_{0,j}] = [a_n,a_{n-1},...a_1, a_0,\ b_{m-1},b_{m-2},...b_1, b_0]$, *i.e.* when the estimated model converges to the actual model of the attacked device.

It is possible to establish an analogy between this System Identification attack and the Known Plaintext cryptanalytic attack (STALLINGS, 2006), wherein $i(k)$ and $o(k)$ correspond to the plaintext and ciphertext, respectively, the form of the generic transfer function $Q(z)$ corresponds to the encryption algorithm and the actual coefficients of $Q(z)$ corresponds to the secret key.

Note that, in this Passive System Identification attack, by definition, the attacker does not interfere in – or excite – the system to collect the signals necessary to estimate the model of the attacked system. However, the attack depends on the occurrence of events, that are not controlled by him/her, to produce signals that carry meaningful information for the system identification algorithm. This passive approach can make the system identification more time consuming, until meaningful information transits through the eavesdropped communication links. The situation is even worse if the system is in steady state because no meaningful information may transit through the NCS's communication links for a long time. This results from the fact that the information content of signals measured under steady operating conditions is often insufficient for identification purposes (TULLEKEN, 1990).

3.1.2 Active System Identification Attack

The Active System Identification attack (SA; CARMO; MACHADO, 2017b) is intended to overcome the constraint of the Passive System Identification Attack. It constitutes an alternative to the Passive System Identification attacks in situations where the attacker may not wait so long for the occurrence of meaningful signals. This attack is used to assess the coefficients of the transfer function $G(z) = C(z)P(z)$ of the NCS shown in Figure 4, wherein $C(z)$ is the controller's control function and $P(z)$ is the plant's transfer function. These transfer functions are all LTI. The attack is performed by a MitM that may be located either in the forward or in the feedback link. For the sake of clarity of the analysis presentation, but without loss of generality, we focus on the case where the MitM is in the feedback link, *i.e.* between the plant's sensors and the controller's input.

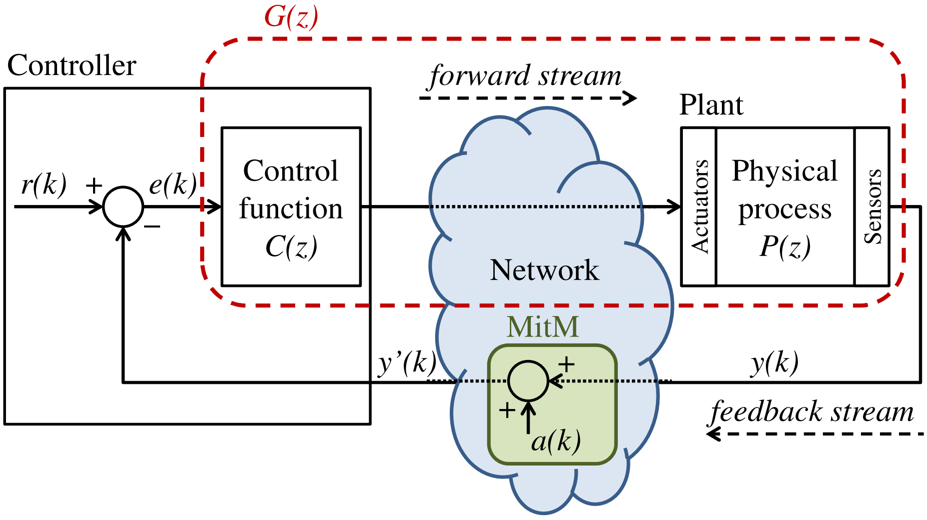


Figure 4 – Active System Identification attack with a MitM in the feedback link – own figure published in (SA; CARMO; MACHADO, 2017b).

To estimate the attacked NCS model, the attacker injects an attack signal $a(k)$ and measures the system response to such signal. The complete response of the generic NCS shown in Figure 4, considering only the inputs $R(z) = \mathcal{Z}[r(k)]$ and $A(z) = \mathcal{Z}[a(k)]$, is expressed in the z domain by (3.4):

$$Y(z) = \frac{G(z)}{1+G(z)}R(z) - \frac{G(z)}{1+G(z)}A(z), \tag{3.4}$$

wherein $Y(z) = \mathcal{Z}[y(k)]$. As a premise, in a normal condition, it is considered that $a(k) = 0$ and the system is designed to make $y(k) \to q$, in such way that $y(k) \approx q$ $\forall k > k_s$, *i.e.* the NCS output $y(k)$ converges and stabilizes at a constant value q after a certain amount of samples k_s. Indeed, it is usually one of the main aims of a control system. Now, considering $a(k) \neq 0$, the output $y(k)$, $\forall k > k_s$, may be defined approximately as (3.5):

$$y(k) = q - \mathcal{Z}^{-1}\left[\frac{G(z)}{1+G(z)}A(z)\right], \forall k > k_s. \tag{3.5}$$

Thus, after k_s, the portion of $y(k)$ caused by $r(k)$ can be eliminated by just subtracting q from (3.5), which leads to (3.6):

$$y_a(k) = y(k) - q = -\mathcal{Z}^{-1}\left[\frac{G(z)}{1+G(z)}A(z)\right], \forall k > k_s. \tag{3.6}$$

wherein $y_a(k)$ represents the portion of $y(k)$ caused by the attack signal $a(k)$. The value of q can be assessed by the attacker through an eavesdropping attack in the feedback stream, by just capturing $y(k)$ after the NCS stabilization. The subtraction of q after k_s makes the system identification attack independent of $r(k)$ $\forall k > k_s$. The Active System Identification attack now just relies on the attack signal $a(k)$ – which can be chosen – and the system response to the attack $y_a(k)$, which can be obtained in accordance with (3.6). The signal $y_a(k)$ starts with the injection of $a(k)$ and has the size of a monitoring period T.

If the attack input $a(k)$ and its consequent output $y_a(k)$ are known, the model of $G(z)$ can be assessed by applying $a(k)$ in an estimated system, defined by (3.7):

$$\hat{y}_a(k) = -\mathcal{Z}^{-1}\left[\frac{G_e(z)}{1+G_e(z)}\right] * a(k), \tag{3.7}$$

wherein $G_e(z)$ is the estimation of $G(z)$ and $\hat{y}_a(k)$ is the output of the estimated system in face of $G_e(z)$. By comparing $\hat{y}_a(k)$ with $y_a(k)$, the attacker is capable to evaluate whether $G_e(z)$ is equal/approximately $G(z)$. Note that $G_e(z)$ is a generic transfer function represented by (3.8):

$$G_e(z) = \frac{\alpha_n z^n + \alpha_{n-1} z^{n-1} + ... + \alpha_1 z^1 + \alpha_0}{z^m + \beta_{m-1} z^{m-1} + ... + \beta_1 z^1 + \beta_0}, \tag{3.8}$$

wherein n and m are the order of the numerator and denominator, respectively, while $[\alpha_n,\alpha_{n-1},...\alpha_1, \alpha_0]$ and $[\beta_{m-1},\ \beta_{m-2},...\beta_1, \beta_0]$ are the coefficients of the numerator and denominator, respectively, that are intended to be found by this Active System Identification attack. Therefore, to find $G(z)$, the coefficients of $G_e(z)$ are adjusted until the estimated output $\hat{y}_a(k)$ converges to the known $y_a(k)$.

In this attack, two bio-inspired metaheuristics – BSA and Particle Swarm Optimization (PSO) (KENNEDY J. E EBERHART, 1995) – are used to iteratively adjust the estimated model, by minimizing a specific fitness function until the estimated model $G_e(z)$ converges to the actual $G(z)$ of the real NCS. To compute the fitness of the individuals of the optimization algorithm (*i.e.* BSA or PSO), the same attack signal $a(k)$ that caused $y_a(k)$ is applied on the estimated system defined by (3.7) and (3.8), where the coefficients of $G_e(z)$ are the coordinates $x_j = [\alpha_{n,j},\alpha_{n-1,j},...\alpha_{1,j},\ \alpha_{0,j},\beta_{m-1,j},\ \beta_{m-2,j},...\beta_{1,j},\beta_{0,j}]$ of an individual j of the BSA/PSO. The output $\hat{y}_{aj}(k)$ is the response of the estimated model (3.7) (3.8) in face of $a(k)$, when the coefficients of $G_e(z)$ are x_j. Then, the fitness f_j of each individual j is obtained comparing $\hat{y}_{aj}(k)$ with $y_a(k)$, according to (3.9):

$$f_j = \frac{\sum\limits_{k=0}^{N} (y_a(k) - \hat{y}_{aj}(k))^2}{N}, \tag{3.9}$$

wherein N is the number of samples that exist during a monitoring period T of $y_a(k)$. Note that, if no other inputs – perturbation or noise – occur in the NCS during T, then $\min f_j = 0$ when $[\alpha_{n,j},\alpha_{n-1,j},...\alpha_{1,j},\alpha_{0,j},\ \beta_{m-1,j},\beta_{m-2,j},...\ \beta_{1,j},\beta_{0,j}] = [\alpha_n,\alpha_{n-1},...\alpha_1,\alpha_0,\beta_{m-1},\ \beta_{m-2},...\beta_1,\beta_0]$, *i.e.* when the estimated $G_e(z)$ converges to $G(z)$.

An analogy may be established between this Active System Identification attack and the Chosen Plaintext cryptanalytic attack (STALLINGS, 2006), wherein $a(k)$ corresponds to the chosen plaintext, $y_a(k)$ represents the ciphertext, equations (3.7) and (3.8) together correspond to the encryption algorithm, and the actual coefficients $[\alpha_n,\alpha_{n-1},...\alpha_1,\alpha_0]$ and $[\beta_{m-1},\beta_{m-2},...\beta_1,\beta_0]$ of $G_e(z)$ correspond to the secret key.

It is worth mentioning that this attack requires the previous knowledge about the order of the numerator and denominator of equation (3.8) (n and m, respectively). Using the analogy with the Chosen Plaintext cryptanalytic attack, it is equivalent to require the knowledge about the size of the secret key of the encryption algorithm. In this Active System Identification attack – such as in the Passive System Identification attack –, the information of n and m is necessary to define the number of dimensions of the search space of the optimization algorithm (or the number of unknown coefficients of $G(z)$) which must be set to $n + m - 1$. Although this is an attack constraint, this information may be inferred if the attacker, at least, knows what the attacked plant is and what type of controller is being used.

3.2 MODEL-BASED OFFENSIVES

This section describes three covert and accurate model-based attacks that, to be implemented, require the support of system identification attacks: the SD-Controlled Data Injection attack; the Covert Misappropriation attack; and the SD-Controlled Data Loss attack. Section 3.2.1 characterizes and explains the SD-Controlled Data Injection attack (SA; CARMO; MACHADO, 2017c) which, according to the taxonomy of Section 2.1, is intended to be physically covert. Section 3.2.2 describes the the Covert Misappropriation attack, which is endowed with a specific architecture that makes it cybernetically covert from the perspective of the signal arriving at the controller. Lastly, Section 3.2.3 presents the novel SD-Controlled Data Loss attack, which aims to cause the same impacts of the SD-Controlled Data Injection attack, however, by causing packets dropouts instead of injecting false signals in the NCS. It is worth mentioning that these attacks require access to the NCS control loop and data. In this Chapter, as a premise, it is assumed that this requirement is satisfied.

3.2.1 SD-Controlled Data Injection

Based on the taxonomy presented in Section 2.1.1, the attack described in this section is classified as an SD-Controlled Data Injection attack (SA; CARMO; MACHADO, 2017c). It is a model-based attack and its purpose is to reduce the plant MTBF and/or reduce the efficiency of the physical process that the plant performs, by inserting false data in the control loop. At the same time, the attack is designed to meet the requirement of being physically covert, as the definition presented in Section 2.1.2.

One way to degrade the physical service of a plant, for example, is through the induction of an overshoot during its transient response. The overshoots, or peaks occurred when the system exceeds the targeted value during its transient response, can cause stress and possibly damage physical systems such as mechanical, chemical and electromechanical systems (EL-SHARKAWI; HUANG, 1989; TRAN; HA; NGUYEN, 2007). Additionally, once they occur in a short period of time, the overshoots are often difficult to be noticed by a human observer. Another way to degrade the service of a plant is causing a constant steady state error on it, *i.e.* producing a constant error when $t \to \infty$. A low proportion steady state error, besides being difficult to be perceived by a human observer, may reduce the efficiency of the physical process or, occasionally, stress and damage the system in mid/long term.

In the present attack, to achieve either of the two mentioned effects, *i.e.* an overshoot or a constant steady state error, the attacker interfere in the NCS's communication process by injecting false data into the system in a controlled way. To do so, the attacker act as a MitM that executes an attack function $M(z)$, as presented in Figure 5, wherein $U'(z) = M(z)U(z)$, $U(z) = \mathcal{Z}[u(k)]$ and $U'(z) = \mathcal{Z}[u'(k)]$.

The function $M(z)$ is designed based on the models of the plant and controller, both obtained through one of the system identification attacks described in Section 3.1. Therefore, the SD-Controlled Data Injection attack is implemented together with a System Identification attack, in a joint operation composed by two stages:

STAGE-I: The system identification attack is executed to provide the attacker an accurate knowledge about the models of the targeted system, *i.e.* the plant's transfer function $G(z)$ and the controller's control function $C(z)$. This knowledge is obtained based on signals that are collected from the input and output of the NCS's devices.

STAGE-II: The Data Injection attack is performed. The attacker, as an MitM, injects false data in the NCS control loop. The injected false data is computed based on the knowledge obtained by the attacker during STAGE-I, in order to covertly and accurately change the plant physical behavior.

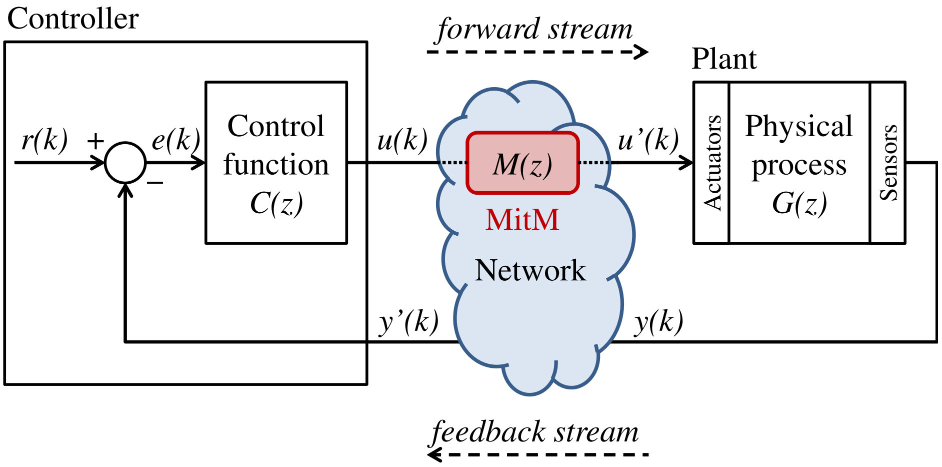


Figure 5 – MitM attack – own figure published in (SA; CARMO; MACHADO, 2017c).

This joint operation is able to degrade the service performed by a plant, through interventions that produce subtle changes on its physical behavior. It is worth mentioning that an uncontrolled intervention in an NCS may lead the plant to an immediate breakdown, or even significantly change its behavior, which may cause the attack disclosure and the eventual failure of the operation. Thus, the changes driven by the attack herein described are dimensioned so that the modifications in the plant's behavior are physically difficult to be perceived. That is why this attack is classified as physically covert.

To ensure that the attack to an NCS is physically covert, the attacker must plan his offensive based on an accurate knowledge about the system dynamics, otherwise the attack consequences may be unpredictable. One possible way to obtain such knowledge is through conventional intelligence operations, performed to collect information regarding the design and dynamics of the NCS. Another way to gather information about the targeted system is through what we refer in Section 2.1.1 as a *Cyber-Physical Intelligence* attack. Specifically, the CPI attack that supports this SD-Controlled Data Injection attack is a system identification attack, which is performed in the aforementioned STAGE-I.

The attack effectiveness, therefore, depends on the design of $M(z)$ which, in turn, depends on the accuracy of the system identification attack. It is worth mentioning that, in Figure 5, although the MitM is placed in the forward stream, it is also possible perform an attack by interfering in the NCS feedback stream. Moreover, the MitM may act in wired or wireless networks, such as in (HWANG et al., 2008).

3.2.2 Covert Misappropriation

The attack for covert misappropriation of NCSs was introduced in (SMITH, 2015). The aim of this attack is to allow a Man-in-the-Middle (MitM) to perform malicious control actions in a physical plant, while remaining undetectable from the point of view of the signals arriving at the original networked controller. Figure 6 shows an implementation of such covert misappropriation attack, based on the attack architecture proposed in (SMITH, 2015), wherein $A(z)$ is the covert controller and $G'(z)$ is model of the plant's actuation to output response – *i.e.* the plant model, which the attacker is supposed to know. The input $\lambda(k)$ drives the attacker's feedback loop and allows the MitM to lead the actual plant output to the desired offset.

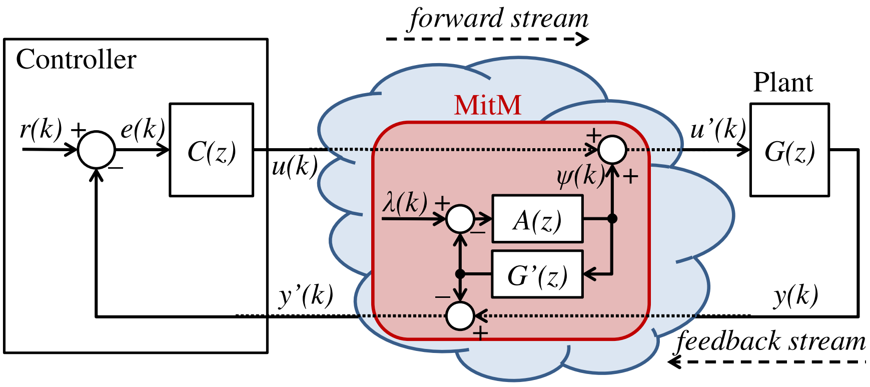


Figure 6 – Covert Misappropriation attack – own figure published in (SÁ; CARMO; MACHADO, 2018).

Note in Figure 6 that, in the forward stream, the MitM performs a data injection attack in which the plant input is given by (3.10):

$$u'(k) = u(k) + \psi(k), \tag{3.10}$$

wherein $\psi(k)$ is the attack signal (3.11):

$$\psi(k) = \lambda(k) * \mathcal{Z}^{-1}\left[\frac{A(z)}{1 + A(z)G'(z)}\right], \tag{3.11}$$

wherein $\mathcal{Z}$ represents the Z-transform operation. Therefore, considering this data injection, the plant output $Y(z) = \mathcal{Z}\left[y(k)\right]$ is defined as (3.12):

$$Y(z) = \mathcal{Z}\left[u(k) + \psi(k)\right] G(z). \tag{3.12}$$

Yet, from Figure 6, it is possible to see that in the feedback stream the MitM also implements a data injection attack in order to manipulate the controller's input signal

$Y'(z) = \mathcal{Z}\left[y'(k)\right]$. With this manipulation, considering (3.12), the signal that arrives at the controller is defined as (3.13):

$$\begin{aligned} Y'(z) &= Y(z) - \mathcal{Z}\left[\psi(k)\right] G'(z) \\ &= \mathcal{Z}\left[u(k) + \psi(k)\right] G(z) - \mathcal{Z}\left[\psi(k)\right] G'(z). \end{aligned} \tag{3.13}$$

In this sense, if the attacker perfectly knows the model of the actual plant – *i.e.* if $G'(z) = G(z)$ –, then (3.13) can be rewritten as (3.14):

$$\begin{aligned} Y'(z) &= \mathcal{Z}\left[u(k) + \psi(k)\right] G(z) - \mathcal{Z}\left[\psi(k)\right] G(z) \\ &= \mathcal{Z}\left[u(k) + \psi(k) - \psi(k)\right] G(z) \\ &= \mathcal{Z}\left[u(k)\right] G(z) \end{aligned} \tag{3.14}$$

which, from the controller's point of view, means that the plant is behaving as in a normal operation, where $Y(z) = \mathcal{Z}\left[u(k)\right] G(z)$. In other words, by analyzing $y(k)$, one should assume that $u'(k) = u(k)$ and $y'(k) = y(k)$ and, therefore, there is no data injection attack in the NCS.

As demonstrated in (SMITH, 2015) and explained in this section, the Covert Misappropriation attack is model-based and its cybernetic covertness depends on how accurate is the plant model known by the attacker. Despite this fact, in (SMITH, 2015) it was not discussed the step that should be taken before implementing the referred attack – *i.e.*, a cyber attack to obtain the plant model (a System Identification attack). Therefore, an evaluation on how effective the joint operation of a System Identification attack and a Covert Misappropriation attack was not done in (SMITH, 2015). In the present work – specifically in Chapter 5 –, it is demonstrated how effective the Covert Misappropriation attack can be when supported by a System Identification attack, enforcing the importance to prevent the disclosure of the NCS models.

3.2.3 SD-Controlled Data Loss

The Controlled Data Loss attack herein proposed is intended to degrade the service performed by a plant. It aims to reduce the efficiency of the physical process being controlled, or even diminish the plant MTBF in mid/long term. For this reason, according to the taxonomy presented in Section 2.1, this attack belongs to the category of Service Degradation (SD) attacks and, therefore, is referred to as an SD-Controlled Data Loss attack. The attack is designed to produce subtle and harmful changes in the plant behavior by causing data loss in the NCS communication process. Additionally, special care is taken to avoid the indiscriminate loss of samples, as well as the complete denial of communication, which could facilitate the attack disclosure.

As discussed in Section 3.2.1, one possible strategy to degrade the service of a plant is by inducing overshoots on it, which can stress and eventually damage the physical system (SA; CARMO; MACHADO, 2017c; EL-SHARKAWI; HUANG, 1989; TRAN; HA; NGUYEN, 2007). In this sense, for the sake of presentation clarity, but without loss of generality, in this work, the SD-Controlled Data Loss attack is designed to cause overshoots on the plant. It would be possible, however, to design other controlled data loss attacks to cause different harmful behaviors, such as increasing the plant settling time.

For an SD-Controlled Data Injection attack to cause an overshoot, the attacker – acting as an MitM – modifies the data flowing in the forward and/or feedback streams of the NCS, based on the models obtained through a System Identification attack. However, to be able to perform an SD-Controlled Data Injection attack and send forged samples to the attacked device, the attacker may have to deal with possibly existing security mechanisms for data integrity and authenticity, which increases the attack complexity.

The SD-Controlled Data Loss attack arises as an alternative to produce the same kind of harmful behavior in a simpler fashion, *i.e.*, without the need to deal with these security mechanisms. In the proposed attack strategy, the overshoot is produced by causing the loss of packets containing specific samples in the forward and feedback streams of the NCS. In a Wireless Networked Control System (WNCS), for instance, loss of samples can be caused by jamming the wireless links during data transmission. To find which samples must be lost to achieve a certain desired overshoot effect, the BSA can once again be used, as it was used in the context of the System Identification attacks presented in Section 3.1. To do so, a fitness function whose minimum value leads to an attack solution must be designed.

To design such fitness function, firstly, it is necessary to consider how the attacked system deals with a lost sample. In NCSs, where real-time communication is normally required, the retransmission of old samples is generally not useful (HESPANHA; NAGHSHTABRIZI; XU, 2007). Therefore, as described in (DASGUPTA et al., 2013; HESPANHA; NAGHSHTABRIZI; XU, 2007), it is often assumed that when the packet containing a sample $s(k)$ is dropped, the NCS uses the value of most recently received sample. This assumption is also made in the design of the present data loss attack, where $s(k)$ may be a sample in either forward or feedback streams.

Now, let $S = \{s(k), s(k+1), \ldots, s(k+h-1)\}$ be a sequence of h samples among which the attacker will choose the ones that should be lost to cause the desired harmful behavior. Based on this, let $W = \{b_0, b_1, \ldots, b_{(h-1)}\}$ be a word of h bits used to indicate which samples of S should be lost or preserved from the attacker's point of view. In this representation, a bit of W is set to 0 if the attacker should cause the loss of the corresponding sample in S. Conversely, a bit of W is set to 1 if the attacker should preserve the corresponding sample in S. We consider the possibility that an attacker will remove samples from forward and/or feedback streams. Therefore, in an attack solution, let the words W_{fw} and W_{fb} indicate which samples should be lost in the sequences $S_{fw} = \{u(k),u(k+1),\ldots,u(k+h-1)\}$ and $S_{fb} = \{y(k),y(k+1),\ldots,y(k+h-1)\}$ of the forward and feedback streams, respectively.

The attacker's task is to find the words W_{fw} and W_{fb} that cause the desired harmful behavior in the NCS. This task is performed using the BSA, taking into account the models previously obtained by a System Identification attack. The coordinates of an individual j of the BSA are $x_j = [W_{fw,j},W_{fb,j}]$, wherein $W_{fw,j}$ and $W_{fb,j}$ are an estimation of W_{fw} and W_{fb}. Aiming an overshoot, the fitness $\mathcal{F}_j$ of each individual j is computed according to (3.15):

$$\mathcal{F}_j = \mathcal{F}_{1,j} + \mathcal{F}_{2,j}, \tag{3.15}$$

in which terms $\mathcal{F}_{1,j}$ and $\mathcal{F}_{2,j}$ are computed as indicated in equations (3.16) and (3.17), respectively:

$$\mathcal{F}_{1,j} = \begin{cases} \left(\max \hat{y}_j(k) - \Upsilon\right)^2, & \text{if } k_1 \leq k_{peak} \leq k_2, \\ \mathcal{P}, & \text{otherwise.} \end{cases} \tag{3.16}$$

$$\mathcal{F}_{2,j} = \sum_{k=k_s}^{k_l} \left(\hat{y}_j(k) - y_{ss}\right)^2, \tag{3.17}$$

wherein $\hat{y}_j(k)$ is the plant output of a simulated NCS, in which the set of samples defined by $x_j = [W_{fw,j},W_{fb,j}]$ are lost. The control function and the plant transfer function of the simulated NCS, used to obtain $\hat{y}_j(k)$, are provided by the System Identification attack. The term $\mathcal{F}_{1,j}$ aims to make the overshoot, *i.e.* the peak of $\hat{y}_j(k)$, as close as possible to the overshoot level aimed by the attacker, defined by Υ. Also, it specifies that k_{peak}, *i.e.* the instant when the peak of $\hat{y}_j(k)$ occurs, must be within a specific period bounded by k_1 and k_2. Otherwise, the fitness of the individual is penalized by $\mathcal{P}$, empirically defined. In turn, $\mathcal{F}_{2,j}$ aims to ensure that the plant output $\hat{y}_j(k)$ converges to y_{ss} after a specific moment defined by k_s. The constant y_{ss} is the value which $y(k)$ should assume in steady state without attack. The attacker can obtain y_{ss} by measuring the actual plant output $y(k)$ in a normal operation. Note that, in $\mathcal{F}_{2,j}$, the steady state error is computed from k_s until the end of the simulation, defined by k_l. It is possible to see, from (3.15), (3.16) and (3.17), that $\min \mathcal{F}_j = 0$ when an overshoot with amplitude Υ occurs between k_1 and k_2, and $\hat{y}_j(k)$ stabilizes at y_{ss} after k_s.

3.3 SUMMARY

This chapter presents two System Identification attacks and three model based offensives that can work together in order to cause harmful effects in an NCS, even if the attacker does not have, in a first moment, the models of the attacked system. The joint operation of these attacks occurs according to the following sequence:

1. First, a System Identification attack estimates the NCS models based on signals eavesdropped in the attacked system;
2. Then, a model based offensive (designed based on the models learned through the System Identification attack) performs malicious interferences in the NCS links in order to cause accurate and covert behaviors that affect the plant service.

The System Identification attacks herein proposed are: the Passive System Identification attack; and the Active System Identification attack. Both attacks use bioinspired metaheuristics to estimate the NCS models. The Passive System Identification attack estimates the NCS models through the analysis of signals eavesdropped in the attacked system. This attack does not interfere in the NCS to obtain data for the identification process. Instead, it passively analyzes signals that typically flow in the NCS in normal operating conditions. The Active System Identification attack, in turn, injects an attack signal into the NCS in order to estimate its models based on the system response to the injected signal. From the attacker's perspective, this attack is useful, for instance, when the system is in steady state and the attacker cannot wait for a signal carrying the meaningful information required for the identification process.

The model-based offensives presented in this chapter are: the SD-Controlled Data Injection attack; the Covert Misappropriation attack; and the SD-Controlled Data Loss attack. First, this chapter characterizes the SD-Controlled Data Injection attack, which modifies the data transmitted through the NCS links in order to cause physically covert behaviors that degrade the plant services. To do so, an MitM executes an attack function $M(z)$ that is designed based on the NCS models. Then, this chapter describes the Covert Misappropriation attack, introduced in (SMITH, 2015), which injects an attack signal in the NCS forward stream and, then, eliminates from the feedback signal the interference it caused to the plant. The attack architecture allows the attacker to be cybernetically covert from the perspective of signals arriving at the controller. To compute the signal that is subtracted from the feedback stream, the attacker uses the models provided by the system identification attack. Finally, this chapter proposes the SD-Controlled Data Loss attack. This novel attack aims to cause subtle and harmful changes in the plant behavior by causing data loss in the NCS communication process. To smartly find a suitable attack solution (*i.e.* a sequence of packets to be dropped through malicious interferences in the communication process), the attacker uses a bioinspired metaheuristic and the models previously learned through the System Identification attack. Chapter 5 presents simulation results of the attacks covered in the present chapter.

4 COUNTERMEASURES

This chapter presents two countermeasures against attacks in NCSs. The first countermeasure, described in Section 4.1, is used to hinder an attacker from obtaining the NCS models through the System Identification attacks proposed in Section 3.1. The second countermeasure, presented in Section 4.2, introduces a link monitoring strategy to identify possible SD-Controlled Data Injection attacks – characterized in Section 3.2.1 – in the NCS links.

4.1 MITIGATION OF SYSTEM IDENTIFICATION ATTACKS

This section presents the use of switching controllers as a technique to hinder system identification attacks. The use of this technique is motivated by the concept that the less the attacker knows the NCS, the more difficult is his task implement a covert/model-based attack. In Section 4.1.1, it is discussed the motivation and the role of this technique as part of a layered defense strategy. Section 4.1.2 presents the underlying details of the referred technique.

4.1.1 Discussion

An NCS owner might think being safe from covert/model-based attacks, supposing that an eventual attacker does not know the plant's design and, thus, its models. Notwithstanding, this work demonstrates how a covert/model-based attack may be built starting from few information about the NCS – here, the only starting information is the structure of the transfer functions of both the plant and controller. Thus, system security must not be relaxed, and countermeasures have to be adopted.

As shown in Figure 3, a complete model-based attack is composed by a sequence of three individual attacks – or stages –, namely: eavesdropping; system identification (active or passive); and a model-based interference (a controlled data injection, packet loss, or jitter). Note that the requirements specified in Figure 3 help in the development of layered defense strategies (HAHN, 2016) for covert/model-based attacks, where both information technology (IT) and operational technology (OT) countermeasures may be involved. Thus, a set of preventive countermeasures can be systematically thought based on the requirements drawn in Figure 3:

I - The first, and straightforward preventive countermeasure, is to increase the difficulties for an attacker to have access to the control loop which, according with

Figure 3, may prevent the execution of the three mentioned stages of the attack. According to (STOUFFER et al., 2015) the most effective architectural concept to protect an NCS is to segregate the control network from other networks. However, sometimes, it is not feasible or even wanted. Then, the possibility of an undesirable access to the control loop can be reduced by applying network segmentation, DMZ, firewall policies and using specific network architectures, such as established in the guidelines described in (STOUFFER et al., 2015). In the case of WNCS, that are techniques designed to minimize the transmitting power of the network devices (SADI; ERGEN; PARK, 2014) that should be used in order to reduce the probability of an attacker getting access to the control loop. Note that, minimizing the transmitting power of the WNCS's devices also minimizes the area from where the control loop can be accessed, which preventively reduces the probability to have the proposed attack launched on the WNCS.

II - In addition to the countermeasures aimed to prevent access to the control loop, other countermeasures are recommended to deny the access to the data that flows through the NCS, in case the former fails. In (PANG; LIU, 2012), it is proposed a countermeasure that integrates a symmetric-key encryption algorithm, a hash algorithm and a timestamp strategy to form a secure transmission mechanism between the controller side and plant side, which is responsible for enforcing the data confidentiality and checking its integrity and authenticity. The use of such countermeasure should hinder the access to the NCS data, which, according to Figure 3, is required for the system identification attack and the model-based data injection (an SD-Controlled Data Injection attack, for instance).

III - Another way to avoid a covert/model-based attack is preventing the attacker to obtain the required knowledge about the system. If the attacker eventually gets access to the NCS's control loop and data, then it is necessary to make the system identification process harder and/or less accurate. Thus, the third preventive countermeasure lies on the use of control functions harder to be accurately identified, such as switching controllers (ZHANG; FAN; HAO, 2012; SA; CARMO; MACHADO, 2018; SA; CARMO; MACHADO, 2017d), for instance. The strategy of using switching controllers to mitigate system identification attacks is presented in Section 4.1.2.

4.1.2 Mitigation using Switching Controllers

As discussed in Section 4.1.1, one possible strategy to mitigate system identification attacks is to build the NCS with specific transfer functions that are harder to be identified. Therefore, it is necessary to analyze the two transfer functions $C(z)$ and

$G(z)$, shown in Figure 1, to verify what can be done to hinder the NCS identification. Regarding the plant, it is not desired or even feasible to modify its transfer function $G(z)$ just to make it harder to be identified. This follows from the simple fact that the plant's transfer function is a consequence of the physical structure of the controlled system. In other words, modify $G(z)$ means to modify the physical process being controlled, which is not convenient. However, it is reasonable to think about the design of controllers that are capable to meet, simultaneously, two objectives:

Objective I - Comply with the plant control requirements. In general, the primary requirement is to preserve the system stability. However, additional requirements – such as low settling time, low overshoot, etc. – may be considered depending on the process being controlled.

Objective II - Hinder the identification of the controller, so that the model obtained by the attacker is imprecise or ambiguous, in such a way that the attacker hesitates to launch covert or model-based attacks against the NCS.

Considering these two objectives, this work proposes the use of randomly switching controllers to mitigate system identification attacks and, thus, prevent the design of a set of covert/model-based attacks. Note that, the use of a switching controller does not avoid the identification of the plant's transfer function $G(z)$ by the Passive System Identification attack described in Section 3.1.1. Regardless of the controller switchings, the plant's transfer function is still an LTI system that can be identified by the mentioned System Identification attack, based on the analysis of the plant's input and output signals.

A Switching Controller, shown in Figure 7, is composed by a set of N control functions $C_i(z)$, $i \in \mathcal{I} = \{1,...,N\}$, that are switched by a switching rule S, to perform the control of a plant $G(z)$. If all control functions $C_i(z)$ and the plant's transfer function $G(z)$ are linear, as the NCS herein discussed, then the system is referred as a *switched linear system* (SLS). For the sake of clarity, but without loss of generality, in the present work, the switching controller is represented and discussed with only two control functions $C_1(z)$ and $C_2(z)$ – *i.e.*, $N = 2$.

In a conventional switching controller (SKAFIDAS et al., 1999; LIBERZON; MORSE, 1999; SAFAEI et al., 2014; FERRARA; SACONE; SIRI, 2015), whose sole objective is to control the plant, the switching rule S, in general, orchestrates the switching events based on the plant and/or network behaviors. However, in the solution proposed in this work, the switching rule is not driven by the plant and/or network

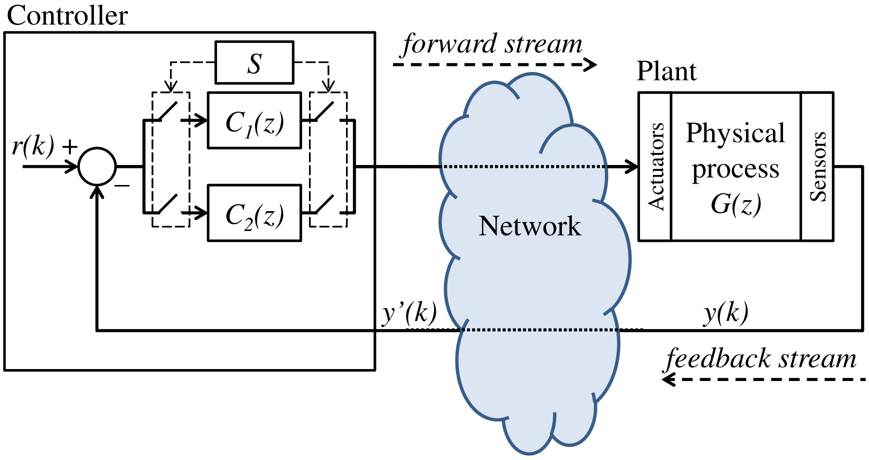


Figure 7 – NCS with a switching controller – own figure published in (SA; CARMO; MACHADO, 2018).

behaviors. To achieve both Objectives I and II, the switching rule herein proposed operates as the Markov chain shown in Figure 8. In this scheme, the control functions are switched at random intervals, in accordance with probabilities $p_{11}(l)$, $p_{12}(l)$. $p_{21}(l)$ and $p_{12}(l)$, wherein l is the number of sampling intervals occurred since the last switch. The probabilities, $p_{12}(l)$ and $p_{21}(l)$ are taken from the probability density function (PDF) shown in Figure 9, wherein a is the minimum number of sampling intervals that the system have to remain in the same state and b is the maximum number of sampling intervals that the system can remain in the same state. Note that $p_{11}(l) = 1 - p_{12}(l)$ and $p_{22}(l) = 1 - p_{21}(l)$.

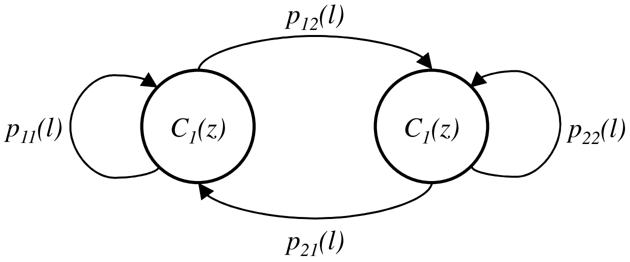


Figure 8 – Markov chain switching rule – own figure published in (SA; CARMO; MACHADO, 2018).

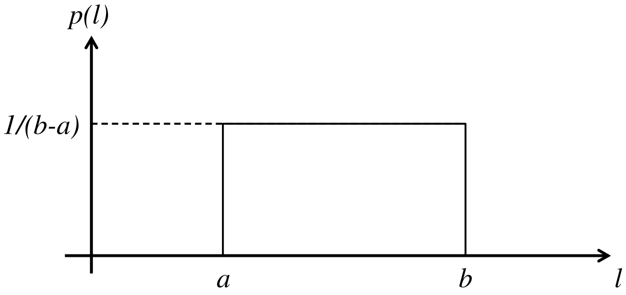


Figure 9 – PDF of p_{12} and p_{21} – own figure published in (SA; CARMO; MACHADO, 2018).

The reason to switch at random intervals is that, according to (WANG, 2013), if the switching times are known, the SLS identification is straightforward. However, when the switching times are not available, the SLS identification turns into a nontrivial task. Moreover, even if the attacker obtain the plant's transfer function $G(z)$ and – somehow – discovers the control functions $C_i(z)$, the random switching rule still hinders the set of attacks that depend on the controller model. This follows from the simple fact that it is more difficult to synchronize the interference caused by these attacks with the controller states, which are switched at random intervals.

However, despite the benefits that the switchings can bring from the point of view of a countermeasure, it can affect the stability of the NCS. Even if all subsystems of an SLS are stable, there are situations in which the switching events can make the SLS unstable. According to (LIN; ANTSAKLIS, 2009; DASGUPTA et al., 2013), to be stable under arbitrary and unrestricted switchings, the SLS must meet two conditions:

1. All its subsystems must be asymptotically stable; and
2. There must exist a common Lyapunov function for all of its subsystems.

Note that, in the case of the NCS shown in Figure 7, each subsystem is constituted by the plant transfer function $G(z)$ arranged in a closed loop with one control function $C_i(z)$. So, to make the NCS stable under arbitrary and unrestricted switching, all control functions $C_i(z)$, $i \in \mathcal{I} = \{1,2\}$, have to be designed in order to meet the two aforementioned conditions.

Another valid strategy to obtain stability in an SLS with stable subsystems is by restricting the switching events. This can be done, for example, by establishing a minimum *dwell time* – *i.e.* the time between two consecutive switches. In an SLS, the instability generated when switching among two – or more – stable subsystems is caused by the failure to absorb the energy increase, caused by the switchings (LIN; ANTSAKLIS, 2009). Intuitively, it is reasonable to think that if an SLS stays at stable subsystems long enough – using a slow switching rule – it becomes able to avoid the energy increase caused by the switchings, maintaining the desired stability. As proved in (MORSE, 1996), it is always possible to preserve the stability of an SLS when all the subsystems are stable and the dwell time is sufficiently large. Actually, it is not critical if the SLS occasionally have a smaller dwell time, provided this does not occur too frequently. As demonstrated in (HESPANHA; MORSE, 1999), if all subsystems are exponentially stable, then the SLS remains exponentially stable provided that the *average dwell time* is sufficiently large. In (ZHAI et al., 2002), this concept of *average dwell-time* is extended to the discrete-time switched systems – which is the case of an NCS endowed with the proposed countermeasure.

In the present work, instead of designing $C_1(z)$ and $C_2(z)$ to make the SLS stable under arbitrary and unrestricted switchings – *i.e.* meeting both conditions 1 and 2 – the restricted switching strategy is used. Thus, $C_1(z)$ and $C_2(z)$ are firstly designed based on the root-locus analysis (DORF; BISHOP, 2011), in order to make each subsystem stable. Then, the overall stability of the SLS is obtained by adjusting the parameters a and b of the PDF shown in Figure 9, aiming an *average dwell-time* that makes the NCS stable.

Besides being adjusted for stability, parameters a and b also have to be adjusted to hinder the system identification attack. So, concerning Objective I, specifically for the sake of stability, a and b are increased as much as possible to ensure the minimum *average dwell-time* required for stability. On the other hand, concerning Objective II, a and b are adjusted to make the system identification attack as much imprecise/ambiguous as possible, which not necessarily occur with high dwell times. In this sense, in this work, a and b are empirically adjusted in order to satisfy the two potentially conflicting objectives.

Note that the use of switching controllers does not prevent the plant identification when it is individually identified. For instance, if the Passive System Identification attack is launched directly on the plant, the identification process is not impacted by the switching controller given that the plant output only depends on the plant's input and its transfer function. Therefore, the use of a switching controller does not prevent the design of attacks that require the knowledge of only the plant transfer function. An example is the Covert Misappropriation attack described in Section 3.2.2. If the attacker aims a cybernetically covert attack, he/she can achieve this goal by using the attack architecture shown in Figure 6, which only requires the knowledge of the plant transfer function. On the other hand, as evaluated in Chapter 6, the proposed countermeasure hinders the design of attacks that depend on the knowledge of $C(z)$ or rely on the knowledge of an open-loop transfer functions composed by $C(z)$.

4.2 IDENTIFICATION OF CONTROLLED DATA INJECTION ATTACKS

This section proposes a link monitoring strategy to identify the LTI transfer function that is performed by a MitM during an SD-Controlled Data Injection attack (described in Section 3.2.1). From the NCS owner perspective, the knowledge about the attack function may be useful, for instance, to:

- provide information for an autonomous process intended to redesign the NCS control function, in order to mitigate the attack effects in the plant behavior;

- reveal the attacker intentions, for forensic purposes, helping to estimate the possible impacts of the attack on the plant and its services.

Section 4.2.1 describes the proposed link monitoring strategy, which uses white gaussian noise to excite the attack function and obtain the information necessary for the identification process. To increase the accuracy of the attack identification using white gaussian noise, this work proposes a Noise Impulse Integration (NII) technique, which is presented in Section 4.2.2.

4.2.1 Strategy to Identify the Attack

This section describes a link monitoring strategy to identify the LTI attack functions used by a MitM during the SD-Controlled Data Injection attack characterized in Section 3.2.1. Consider, for instance, the SD-Controlled Data Injection attack shown in Figure 10, where the attacker only has access to the data flowing in the feedback stream.

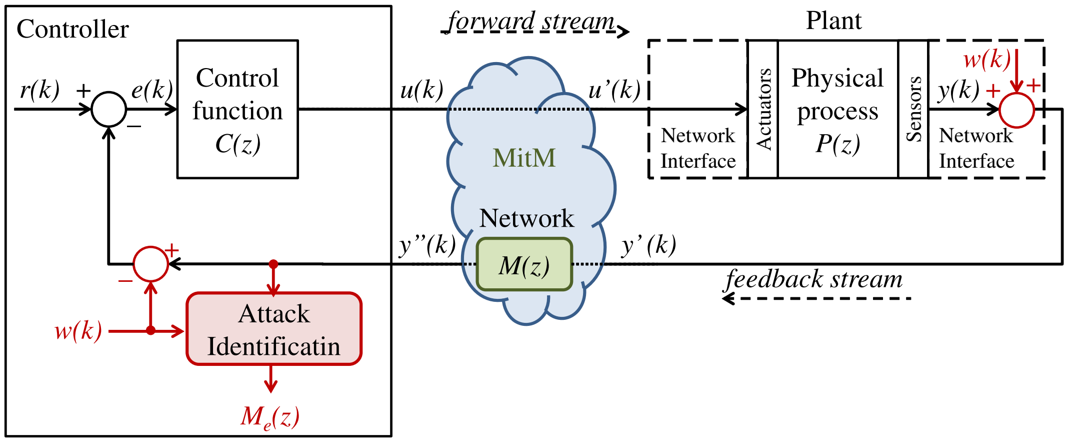


Figure 10 – Identification of an SD-Controlled Data Injection attack.

As discussed in Section 3.1, the LTI system to be identified – in the present problem, the attack function $M(z)$ – has to be excited by an input signal, in order to produce meaningful information for the identification process. If the system is in steady operating conditions, for instance, the information content of measured signals is often insufficient for identification purposes (TULLEKEN, 1990). Considering this, one possible strategy to identify an attack function is to use typical variations in the NCS signals – such as a variation caused by a change in the setpoint $r(k)$ – to estimate $M(z)$. However, depending on the system, this variations may not occur often, which can make the identification of $M(z)$ time consuming. Furthermore, causing arbitrary variations in such signals in order to identify $M(z)$ may not be convenient as it may affect the behavior of the plant.

The architecture shown in Figure 10 is proposed as a solution that can be used to excite $M(z)$ at any time, without affecting the plant behavior when the system is working in normal conditions – *i.e.*, without attack. To do so, as shown in Figure 10, a white gaussian noise $w(k)$ is injected (added) in the signal to be transmitted through the monitored link. To avoid interfering in the controlled plant when the system in not under attack, the same noise signal $w(k)$ is subtracted from the monitored NCS signal at the other end of the link. In Figure 10, where the feedback link is the one being monitored, $w(k)$ is injected at the sensor's network interface, and subtracted at the controller input. In this system, the NCS output $Y(z) = \mathcal{Z}[y(k)]$ is defined as (4.1):

$$Y(z) = \frac{C(z)P(z)}{1 + C(z)P(z)M(z)} \left[R(z) + W(z)\left(1 - M(z)\right)\right], \tag{4.1}$$

wherein $R(z) = \mathcal{Z}[r(k)]$ and $W(z) = \mathcal{Z}[w(k)]$. Note that, if $w(k)$ is exactly the same signal at both ends of the monitored link and the system is not under attack (*i.e.*, $M(z) = 1$), then the injection of $w(k)$ is cancelled and does not influence in $y(k)$. In this case, based on (4.1), the plant output $Y(z)$ is defined as (4.2):

$$Y(z) = \frac{C(z)P(z)}{1 + C(z)P(z)} R(z). \tag{4.2}$$

The white gaussian noise $w(k)$ is chosen to excite the attack function due to its unpredictability, which makes it harder for an attacker to estimate the noise that will be added to the link at any given moment. The white gaussian noise $w(k)$ is obtained from a normal distribution, such that $w(k) \sim N(\mu,\sigma)$, wherein $\mu = 0$ is the mean and σ is the standard deviation. To have the same noise signal $w(k)$ at both ends of the monitored link, it is considered that these two sources of noise are synchronized and both signals are produced based on the same seed. Moreover, to avoid an attacker to predict the noise values, the seed is exchanged among both devices – *i.e.*, the transmitter and receiver – using a secure key exchange method, such as the Diffie-Hellman algorithm (STALLINGS, 2006).

Now, if the system is under attack (*i.e.*, $M(z) \neq 1$), then, according to (4.1), the noise is not cancelled. In this case, the the signal observed at the controller input $y''(k)$ is given by (4.3):

$$\begin{aligned} y''(k) = {} & \underbrace{w(k) * \mathcal{Z}^{-1}\left[M(z)\left(\frac{1 + C(z)P(z)}{1 + C(z)P(z)M(z)}\right)\right]}_{y_1''(k)} + \\ & \underbrace{r(k) * \mathcal{Z}^{-1}\left[\frac{C(z)P(z)M(z)}{1 + C(z)P(z)M(z)}\right]}_{y_2''(k)}. \end{aligned} \tag{4.3}$$

In the present countermeasure, the identification of $M(z)$ is performed by observing the variations produced by $w(k)$ in $y''(k)$ when $M(z) \neq 1$. Note, in Figure 10, that both $w(k)$ and $y''(k)$ are provided to the Attack Identification process. The effect of $w(k)$ in $y''(k)$ is specifically indicated in (4.3) as $y_1''(k)$. To have the identification relying on $y_1''(k)$, and independent from variations in $y_2''(k)$, it is executed when the system is in steady state with regard to $r(k)$. In other words, the identification occurs when $y_2''(k)$ – driven by the setpoint $r(k)$ – converges to a constant value ρ. In this case, considering the time window defined by $k_s < k < k_u$ in which $y_2''(k)$ is in its steady state, (4.3) can be rewritten as (4.4) without initial conditons:

$$y''(k) = \underbrace{w(k) * \mathcal{Z}^{-1}\left[M(z)\left(\frac{1 + C(z)P(z)}{1 + C(z)P(z)M(z)}\right)\right]}_{y_1''(k)} + \underbrace{\rho}_{y_2''(k)}, \qquad \forall k_s < k < k_u, \tag{4.4}$$

wherein ρ can be estimated by computing the average $\bar{y}''$ of $y''(k)$ during a certain amount of samples $\tau \leq k_u - k_s$ starting at k_s, as indicated in (4.5):

$$\bar{y}'' = \sum_{k_s}^{k_s+\tau} \frac{y''(k)}{\tau} = \underbrace{\sum_{k_s}^{k_s+\tau} \frac{w(k) * \mathcal{Z}^{-1}\left[M(z)\left(\frac{1+C(z)P(z)}{1+C(z)P(z)M(z)}\right)\right]}{\tau}}_{\bar{y}_1''(k)} + \underbrace{\sum_{k_s}^{k_s+\tau} \frac{\rho}{\tau}}_{\bar{y}_2''(k)}, \tag{4.5}$$

Considering that $w(k) \sim N(\mu,\sigma)$, wherein $\mu = 0$, as previously stated, then $\bar{y}_1''(k) \rightarrow 0$ when $\tau \rightarrow \infty$. In this case, for a sufficiently large τ, (4.5) can be simplified to (4.6):

$$\bar{y}'' \approx \rho, \tag{4.6}$$

Thus, by applying (4.6) in (4.4), we may define (4.7):

$$y_1''(k) \approx y''(k) - \bar{y}'', \qquad \forall k_s < k < k_u, \tag{4.7}$$

wherein $y_1''(k)$ – obtained through measurements of $y''(k)$ – is the output of the model defined by (4.8) when the noise $w(k)$ is applied to its input:

$$y_1''(k) = w(k) * \mathcal{Z}^{-1}\left[M(z)\left(\frac{1 + C(z)P(z)}{1 + C(z)P(z)M(z)}\right)\right]. \tag{4.8}$$

Based on (4.8), if $C(z)$ and $P(z)$ are known, the Attack Identification process can estimate $M(z)$ by applying $w(k)$ in an estimated system, defined by (4.9):

$$\hat{y}_1''(k) = w(k) * \mathcal{Z}^{-1}\left[M_e(z)\left(\frac{1 + C(z)P(z)}{1 + C(z)P(z)M_e(z)}\right)\right], \tag{4.9}$$

wherein $M_e(z)$ is the estimation of $M(z)$ and $\hat{y}_1''(k)$ is the output of the estimated system in face of $M_e(z)$. By comparing $\hat{y}_1''(k)$ with $y_1''(k)$, the Attack Identification process is able to evaluate whether $M_e(z)$ is equal/approximately $M(z)$. Note that $M_e(z)$ is a generic LTI attack function represented by (4.10):

$$M_e(z) = \frac{\alpha_n z^n + \alpha_{n-1} z^{n-1} + ... + \alpha_1 z^1 + \alpha_0}{z^m + \beta_{m-1} z^{m-1} + ... + \beta_1 z^1 + \beta_0}, \tag{4.10}$$

wherein n and m are the order of the numerator and denominator, respectively, while $[\alpha_n,\alpha_{n-1},...\alpha_1, \alpha_0]$ and $[\beta_{m-1}, \beta_{m-2},...\beta_1, \beta_0]$ are the coefficients of the numerator and denominator, respectively, that are intended to be found by Attack Identification algorithm. Therefore, to find $M(z)$, the coefficients of $M_e(z)$ are adjusted until the estimated output $\hat{y}''_1(k)$ converges to $y''_1(k)$ – obtained from measurements of $y''(k)$ in the real NCS.

As in the System Identification attacks described in Section 3.1, the BSA is also used here to iteratively adjust the estimated model, by minimizing a specific fitness function until $M_e(z)$ converges to the actual $M(z)$. To compute the fitness of the BSA individuals, the noise $w(k)$ – recorded while $y''(k)$ was being captured – is applied on the estimated system defined by (4.9) and (4.10), where the coefficients of $M_e(z)$ are the coordinates $x_j = [\alpha_{n,j},\alpha_{n-1,j},...\alpha_{1,j},\ \alpha_{0,j},\beta_{m-1,j},\ \beta_{m-2,j},...\beta_{1,j},\beta_{0,j}]$ of an individual j of the BSA. Let $\hat{y}''_{1j}(k)$ be the output of the estimated model (4.9) (4.10) in face of $w(k)$, when the coefficients of $M_e(z)$ are x_j. Then, the fitness f_j of each individual j is obtained comparing $\hat{y}''_{1j}(k)$ with $y''_1(k)$, according to (4.11):

$$f_j = \frac{\sum\limits_{k=0}^{N} (y''_1(k) - \hat{y}''_{1j}(k))^2}{N}, \tag{4.11}$$

wherein N is the number of samples that exist during a monitoring period T of $y''_1(k)$. Note that, $\min f_j$ occurs when $[\alpha_{n,j},\alpha_{n-1,j},...\alpha_{1,j},\alpha_{0,j},\ \beta_{m-1,j},\beta_{m-2,j},...\ \beta_{1,j},\beta_{0,j}] \rightarrow [\alpha_n,\alpha_{n-1},...\alpha_1, \alpha_0,\beta_{m-1},\ \beta_{m-2},...\beta_1, \beta_0]$, *i.e.* when the estimated $M_e(z)$ converges to $M(z)$.

The attack identification process described in this section, without the use of the Noise Impulse Integration technique (to be described in Section 4.2.2), is summarized in Algorithm 1.

Algorithm 1: Attack Identification without the NII technique

```
begin
    if $y''(k)$ is in steady state with regard to $r(k)$ then
        Record $w(k)$ and $y''(k)$ during $T$ seconds;
        Compute $\bar{y}''$ according to (4.5);
        Compute $y''_1(k)$ according to (4.7);
        Execute BSA, using $w(k)$ and $y''_1(k)$ to find $M_e(z)$ based on (4.9), (4.10)
        and (4.11).
    end if
end
```

4.2.2 Integrating Impulses of Noise

This section presents the Noise Impulse Integration (NII) technique, which is added to the attack identification process described in Section 4.2.1 to improve its accuracy. This technique was reported in (SA et al., 2019) as part of the present research. It is inspired by the Pulse Integration technique (SKOLNIK, 1990), used in pulse radar systems to improve the probability of detection and reduce the probability of false alarms of those systems. To allow a clear comprehension on the inspiration obtained from the radar Pulse Integration technique, it is necessary to provide a brief explanation on how a pulse radar system works and what is the main idea behind the pulse integration process. The explanation on the radar pulse integration process is provided in Section 4.2.2.1. Section 4.2.2.2, then, introduces the NII technique.

4.2.2.1 Radar Pulse Integration

In a pulse radar system, the radar transmits electromagnetic pulses to the environment in order to detect and obtain information about targets. When a pulse reaches a reflective surface – of a target or other objects in the environment –, it is reflected producing an echo that travels back to the radar antenna, allowing the target detection. To increase the probability of detection, the radar does not transmit a single pulse during the detection process. Instead, as depicted in Figure 11, the radar transmits a series of pulses, one at each pulse repetition interval T_R. Also, as shown in Figure 11, between two consecutive transmissions, there is a silence period T_L in which the radar remains listening the ecoes that arrive from the monitored environment. These echoes may represent a target or another reflective body situated within the line of sight of the radar antenna.

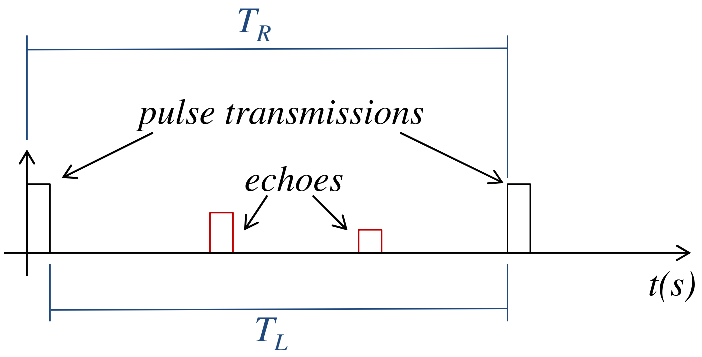


Figure 11 – Pulse transmissions.

Note that, while the radar scans the environment by rotating its antenna, for each antenna pointing angle θ, several pulses are transmitted in sequence as shown in Figure 12. Naturally, for each pulse p transmitted from a given antenna pointing angle

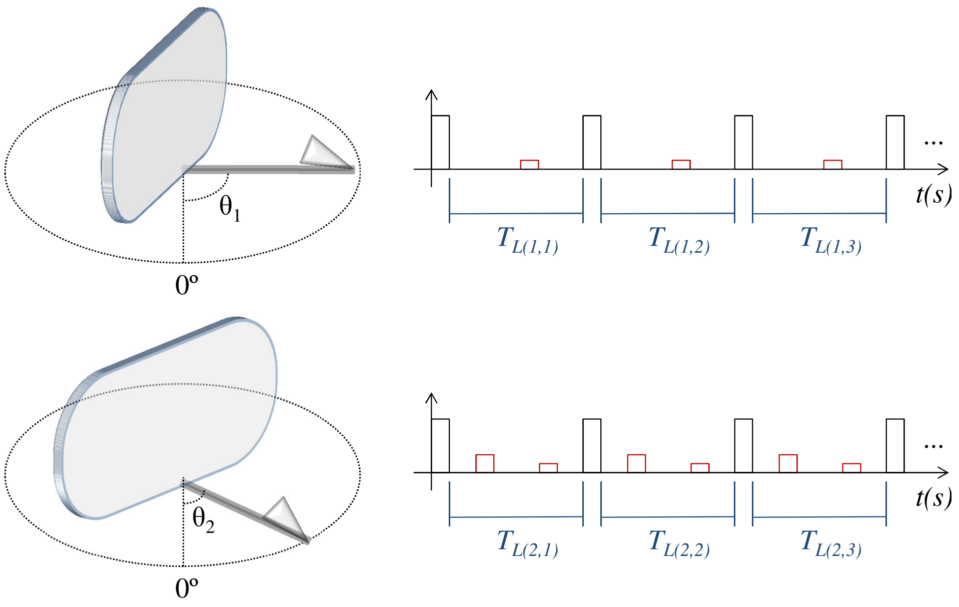


Figure 12 – Radar scan process, in which a sequence of pulses is transmitted for each antenna pointing angle.

θ_d, there will be a listening period $T_{L(d,p)}$ to receive echoes. It happens that, in a real system, the signal received during each listening period $T_{L(d,p)}$ does not contain only target echoes. Typically, as represented in Figure 13, the received signal also contains uncorrelated signal fluctuations (noise), whose amplitude follows a gaussian distribution with zero mean (AHMED, 2015; SCHWARTZ, 1956).

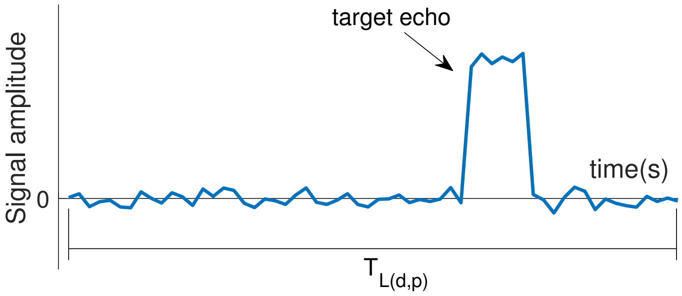


Figure 13 – Noisy signal typically received during a given listening period $T_{L(d,p)}$.

To increase signal-to-noise ratio (SNR), the radar Pulse Integration (RPI) technique combines the signals received in multiple listening periods $T_{L(d,p)}$, in a given θ_d, taking advantage of the mentioned noise properties – *i.e.* uncorrelated fluctuations with gaussian distribution and zero mean. Basically, all signals $S_{(d,p)}(t)$ received in a sequence of listening periods $T_{L(d,p)}$ are integrated by computing their mean according to (4.12):

$$I(t) = \frac{\sum_{p=1}^{h} S_{(d,p)}(t)}{h}, \tag{4.12}$$

wherein $I(t)$ is the integrated signal and h is the amount of signals buffered in a sequence of listening periods. A representation of this computation is shown in Figure 14, where the signals received in a sequence of four listening periods (*i.e.* $h = 4$) are buffered and integrated according to (4.12). Note that, the integrated signal has a better SNR when compared to the other signals. The uncorrelated noise, is minimized (almost cancelled) thanks to its gaussian distribution with zero mean. On the other hand, the target echo (constantly present with non-zero mean amplitude) is reinforced. Ideally, the noise of the integrated signal is completely cancelled when $h \to \infty$. In this case, $I(t)$ would contain only echoes.

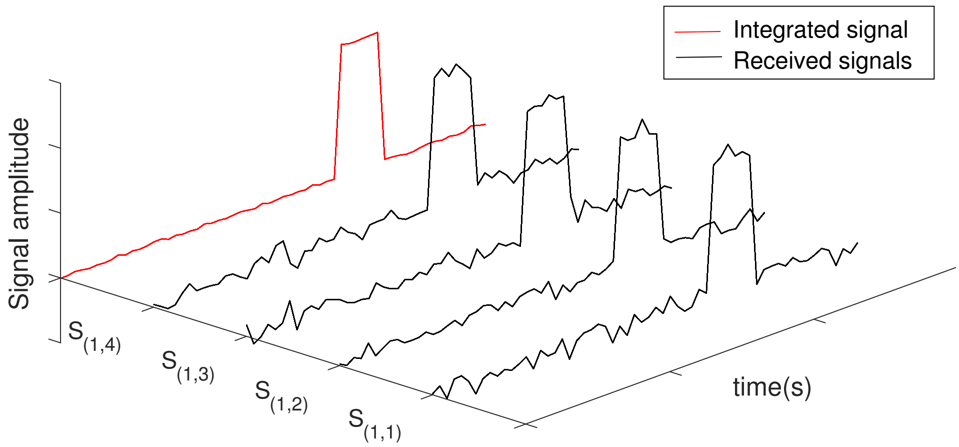


Figure 14 – Radar pulse integration.

4.2.2.2 Noise Impulse Integration Technique

The NII technique described in this section works similarly to the RPI process described in Section 4.2.2.1. Basically, it integrates portions of noisy signals to cancel information that may disturb the identification process, and extract the information that is useful to obtain accurate models. Despite the inspiration obtained from the RPI, it is worth mentioning the following diferences between both techniques:

- **Goal:** The goal of the RPI technique is to minimize the uncorrelated noise contained in signals received by the radar, and reinforce the echoes reflected by bodies within the radar antenna's line of sight – *i.e.*, produce a signal with grater SNR. The goal of the NII technique is to obtain a clear impulse response function of an LTI system, when it is excited by a white gaussian noise;

- **Integrated signals:** The RPI technique integrates signals received between consecutive pulse transmissions, containing, in general, reflected pulses and noise. The

NII technique integrates portions of the signal produced by an LTI system when white gaussian noise is injected into it.

- **Selection of signals to be integrated:** In the RPI technique, the selection of signals to be integrated is straightforward. As explained in Section 4.2.2.1, it integrates signals received between the transmission of consecutive radar pulses. This selection provides a synchronism between the signals to be integrated, which, as shown in Figure 14, aligns the information that must be reinforced by the RPI – *i.e.*, reinforce echoes that are constantly present in the received signal. The RPI's signal selection cannot be used in the NII technique, given that the latter is not triggered by pulses. Therefore, it is necessary to use another criteria to select the portions of signal to be integrated, which is explained in the remainder of this section.

The white gaussian noise $w(k)$, herein used to excite the LTI transfer function to be identified, can be defined as a sum of time-shifted impulses with uncorrelated random weights (amplitudes) according to (4.13):

$$w(k) = \sum_{i=-\infty}^{\infty} \omega(i)\delta(k-i), \tag{4.13}$$

in which the amplitudes $\omega(i) \sim N(\mu,\sigma)$, N is a normal distribution, μ is its mean and σ is its non-zero standard deviation. When a weighted time-shifted impulse $\omega(i)\delta(k-i)$ of $w(k)$ is individually applied to a given LTI system $H(z) = \mathcal{Z}\{h(k)\}$, it produces an output signal $y_i(k)$ defined by (4.14):

$$\begin{aligned} y_i(k) &= \omega(i)\delta(k-i) * h(k) \\ &= \omega(i)h(k-i). \end{aligned} \tag{4.14}$$

Note that $y_i(k)$ is the impulse response of $h(k)$ – *i.e.*, $h(k)$ itself –, weighted by the impulse's amplitude $\omega(i)$ and time-shifted by i samples. However, when $w(k)$ is applied to $h(k)$, the output signal is no more composed by a single weighted time-shifted impulse response function. In this case, the discrete-time output $y(k)$ produced when $h(k)$ is excited by a white gaussian noise $w(k)$ is determined by the discrete convolution (4.15):

$$y(k) = w(k) * h(k). \tag{4.15}$$

Considering (4.13), equation (4.15) can be rewritten as (4.16) and (4.17):

$$y(k) = \sum_{i=-\infty}^{\infty} \omega(i)\delta(k-i) * h(k) \tag{4.16}$$

$$\begin{aligned} y(k) = \sum_{i=-\infty}^{-1} &\omega(i)\delta(k-i) * h(k) \\ &+\omega(0)\delta(k) * h(k) \\ +&\sum_{i=1}^{\infty}\omega(i)\delta(k-i) * h(k). \end{aligned} \tag{4.17}$$

which means that the output $y(k)$ is composed by a sum of randomly weighted time-shifted impulse responses of $h(k)$. Evidently, by observing (4.17), it is possible to verify that $y(k)$ could result in a weighted impulse response of $h(k)$ if conditions (4.18) and (4.19) were met:

$$\omega(0) \neq 0 \tag{4.18}$$

$$\omega(i) = 0, \quad \forall i \neq 0, \tag{4.19}$$

which would make straightforward to reveal $h(k)$ by measuring $y(k)$. However, although condition (4.18) is possible, condition (4.19) is not feasible, given that $\omega(i) \sim N(\mu,\sigma)$, and $\sigma \neq 0$, as previously defined. Thus, the task of the NII technique is to overcome the constraint imposed by condition (4.19). Its goal is to produce a signal derived from $y(k)$ that can reveal $h(k)$ in the same way as if conditions (4.18) and (4.19) were met.

Inspired by the RPI, the NII technique consists of separating portions of $y(k)$ that, when integrated, reinforce selected impulse responses of $h(k)$ and minimize (cancel) the interferences produced other weighted time-shifted impulse responses of $h(k)$ contained in $y(k)$. So, let $y_j(k)$ be a portion of signal extracted from $y(k)$, wherein j is a reference number used to identify each $y_j(k)$. The instances $y_j(k)$ are extracted from the output $y(k)$ based on the amplitudes of the input signal $w(k)$, which is evaluated during a monitoring period staring in sample k_f and ending in sample k_l. This said, each $y_j(k)$ is obtained according Algorithm 2:

Algorithm 2: Generation of signals $y_j(k)$

```
begin
    for $k = k_f$ to $k_l$ do
        if $w(k) \geq \Omega$ then
            $j \leftarrow k$;
            $y_j(k) = y(k + j)$.
        end if
    end for
end
```

According to Algorithm 2, each j is a value of k in which the input $w(k)$ is grater or equal than an amplitude threshold Ω. Note that $y_j(k)$ is an instance of $y(k)$ advanced (left-shifted) by j samples. Thus, in the same way that $y(k)$ is defined by (4.17), $y_j(k)$ can be written as (4.20):

$$\begin{aligned} y_j(k) = \sum_{i=-\infty}^{-1} \omega_j(i)\delta(k-i) * h(k) \\ +\omega_j(0)\delta(k) * h(k) \\ +\sum_{i=1}^{\infty} \omega_j(i)\delta(k-i) * h(k) \end{aligned} \tag{4.20}$$

wherein $\omega_j(i)$, defined according to (4.21), are the advanced (left-shifted) amplitudes of the white gaussian noise (4.13):

$$\omega_j(i) = \omega(i+j). \tag{4.21}$$

Considering that Algorithm 2 is intended to produce a collection of $y_j(k)$ – which is necessary for the NII technique –, let J be the set of all j, and $|J|$ be the total number of elements $j \in J$. So, analogously to the RPI process, the mean $\Upsilon(k)$ of all $y_j(k)$ is computed according to (4.22):

$$\Upsilon(k) = \frac{\sum\limits_{j \in J} y_j(k)}{|J|}, \tag{4.22}$$

Thus, considering (4.20), equation (4.22) can be rewritten as (4.23):

$$\begin{aligned} \Upsilon(k) = \underbrace{\frac{\sum\limits_{j \in J} \left[\sum\limits_{i=-\infty}^{-1} \omega_j(i)\delta(k-i) * h(k) \right]}{|J|}}_{\Upsilon_1(k)} \\ + \underbrace{\frac{\sum\limits_{j \in J} \omega_j(0)\delta(k) * h(k)}{|J|}}_{\Upsilon_2(k)} \\ + \underbrace{\frac{\sum\limits_{j \in J} \left[\sum\limits_{i=1}^{\infty} \omega_j(i)\delta(k-i) * h(k) \right]}{|J|}}_{\Upsilon_3(k)} \end{aligned} \tag{4.23}$$

Note that $\omega_j(i)$ has the same probability distribution function of $\omega(i)$ (*i.e.* $\omega_j(i) \sim N(\mu,\sigma)$) since that, according to (4.21), $\omega_j(i)$ consists of the same amplitudes of $\omega(i)$, however left-shifted. Thus, considering that $\mu = 0$, then $\Upsilon_1(k) \to 0$ and $\Upsilon_3(k) \to 0$

when $|J|$ increases. It means that, for a given $i \neq 0$ the impulse responses produced by all $\omega_j(i)\delta(k-i)$ are canceled when the average of $y_j(k)$ is computed among all $j \in J$.

On the other hand, $\Upsilon_2(k) \neq 0$ since that the mean of $\omega_j(0)$, among all $j \in J$, is different from zero. Note that, according to (4.21) $\omega_j(0) = \omega(j)$. From Algorithm 2, $w(j) \geq \Omega$ which, according to (4.13), means that $\omega(j) \geq \Omega$. So, $\omega_j(0) \geq \Omega$, $\forall j$. This reasoning demonstrates that the mean of all $\omega_j(0)$ is grater than Ω and, therefore, $\Upsilon_2(k) \neq 0$. In this case, the responses produced by all $\omega_j(0)\delta(k)$ are the impulses responses of $h(k)$ selected to be reinforced through the NII technique. This reinforcement is analogous to what the RPI technique does with target echoes. This said, (4.23) can be simplified as (4.24):

$$\Upsilon(k) = \bar{\omega}_j(0)\delta(k) * h(k), \tag{4.24}$$

wherein $\bar{\omega}_j(0)$ is the mean of all $\omega_j(0)$, according to (4.25):

$$\bar{\omega}_j(0) = \frac{\sum\limits_{j \in J} \omega_j(0)}{|J|}. \tag{4.25}$$

An example of the computation performed by the NII technique is represented in Figures 15 and 16. Figure 15 shows a set of signals $y_j(k)$ aligned to be integrated, similarly to the representation shown in Figure 14 for the RPI process. Figure 16 shows the signal $\Upsilon(k)$ produced by the computation of (4.22) using the set of signals represented in Figure 15. The signal $\Upsilon(k)$, highlighted in red, is the result of the integration of all $y_j(k)$ which, according to (4.24), reveals the impulse response of the system as it was excited by the impulse $\bar{\omega}_j(0)\delta(k)$.

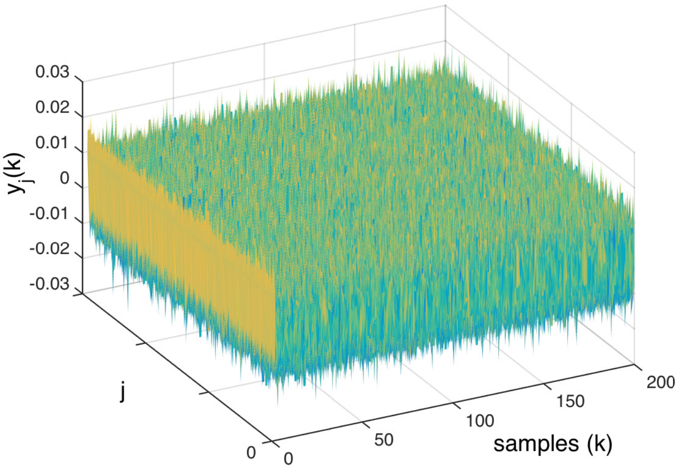


Figure 15 – Signals $y_j(k)$ aligned to be integrated.

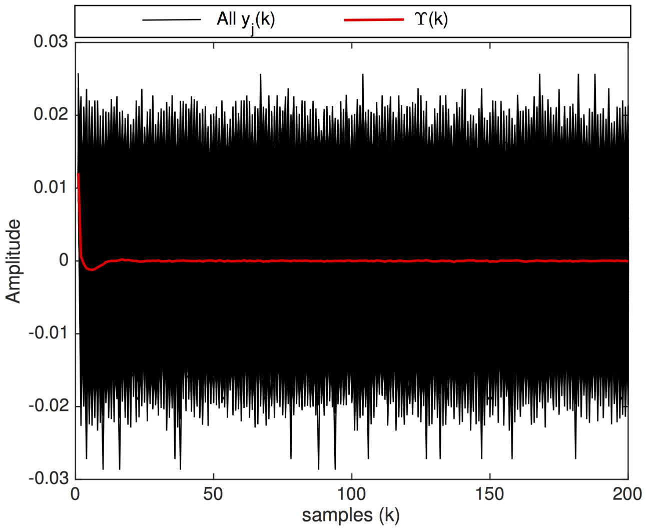


Figure 16 – The impulse response $\Upsilon(k)$ (in red) produced by the NII technique after the integration of a set of signals $y_j(k)$ (shown overlapped in black).

As previously discussed, the NII technique is herein used to complement the attack identification strategy described in Section 4.2.1 in order to improve its accuracy. To do so, lets consider that:

- $\bar{\omega}_j(0)$ and $\Upsilon(k)$ are obtained through the NII technique, by processing signals $w(k)$ and $y_1''(k)$ – specified in Section 4.2.1;

- $h(k)$ is the transfer function between $w(k)$ and $y_1''(k)$ which, according to (4.8), is defined as (4.26):

$$h(k) = \mathcal{Z}^{-1}\left[M(z)\left(\frac{1+C(z)P(z)}{1+C(z)P(z)M(z)}\right)\right]. \tag{4.26}$$

Doing so, (4.24) can be rewritten as (4.27):

$$\Upsilon(k) = \bar{\omega}_j(0)\delta(k) * \mathcal{Z}^{-1}\left[M(z)\left(\frac{1+C(z)P(z)}{1+C(z)P(z)M(z)}\right)\right], \tag{4.27}$$

which can now be used to estimate $M(z)$ in the same way as in Section 4.2.1 for equation (4.8). Note that, the diferences between (4.8) and (4.27) are:

- the input of (4.8) is a white gaussian noise and its output is a white gaussian noise filtered by $h(k)$;

- the input of (4.27) is a weighted impulse signal and its output is a weighted impulse response of $h(k)$.

Now, given (4.27), the attack function $M(z)$ can be estimated by an optimization algorithm (*e.g.* the BSA), such as described in Section 4.2.1. In this case, if $C(z)$ and $P(z)$ are known, $M(z)$ can be estimated by applying $\bar{\omega}_j(0)\delta(k)$ in an estimated system, defined by (4.28):

$$\hat{\Upsilon}(k) = \bar{\omega}_j(0)\delta(k) * \mathcal{Z}^{-1}\left[M_e(z)\left(\frac{1+C(z)P(z)}{1+C(z)P(z)M_e(z)}\right)\right], \tag{4.28}$$

wherein $M_e(z)$ is the estimation of $M(z)$ and $\hat{\Upsilon}(k)$ is the output of the estimated system in face of $M_e(z)$. Recall that $M_e(z)$ is the generic LTI attack function represented by (4.10) wherein $[\alpha_n,\alpha_{n-1},...\alpha_1, \alpha_0]$ and $[\beta_{m-1},\ \beta_{m-2},...\beta_1, \beta_0]$ are the coefficients of the numerator and denominator, respectively, that are intended to be found by Attack Identification algorithm. By comparing $\hat{\Upsilon}(k)$ with $\Upsilon(k)$, the Attack Identification process is able to evaluate whether $M_e(z)$ is equal/approximately $M(z)$.

In the same way that in Section 4.2.1, to discover $M(z)$, the coefficients of $M_e(z)$ are adjusted by the BSA until the estimated output $\hat{\Upsilon}(k)$ converges to $\Upsilon(k)$ – obtained by the NII technique from measurements of $y''(k)$ and $w(k)$ in the real NCS. Let $\hat{\Upsilon}j(k)$ be the output of the estimated model (4.28) (4.10) in face of the input $\bar{\omega}_j(0)\delta(k)$, when the coefficients of $M_e(z)$ are the coordinates $x_j = [\alpha_{n,j},\alpha_{n-1,j},...\alpha_{1,j},\ \alpha_{0,j},\beta_{m-1,j},$ $\beta_{m-2,j},...\beta_{1,j},\beta_{0,j}]$ of an individual j of the BSA. In this case, the fitness f_j of each individual j of the BSA is obtained comparing $\hat{\Upsilon}j(k)$ with $\Upsilon(k)$, according to (4.29):

$$f_j = \frac{\sum\limits_{k=0}^{\mathcal{N}}(\Upsilon(k) - \hat{\Upsilon}j(k))^2}{\mathcal{N}}, \tag{4.29}$$

wherein $\mathcal{N}$ is the number of samples that exist in $\Upsilon(k)$. As already discussed in Section 4.2.1, $\min f_j$ occurs when $[\alpha_{n,j},\alpha_{n-1,j},...\alpha_{1,j},\alpha_{0,j},\ \beta_{m-1,j},\beta_{m-2,j},...\ \beta_{1,j},\beta_{0,j}] \rightarrow$ $[\alpha_n,\alpha_{n-1},...\alpha_1, \alpha_0,\beta_{m-1},\ \beta_{m-2},...\beta_1, \beta_0]$, *i.e.* when the estimated $M_e(z)$ converges to $M(z)$.

The complete attack identification process described in this section, performed with the Noise Impulse Integration technique, is summarized in Algorithm 3. Note that the differences between Algorithms 1 and 3 is that the former does not have the NII stage. This way, while Algorithm 1 uses $w(k)$ and $y_1''(k)$ as input signals to the BSA-based identification, Algorithm 3 uses $\bar{\omega}_j(0)\delta(k)$ and $\Upsilon(k)$ as input signals to the BSA-based identification.

Algorithm 3: Attack Identification with the NII technique

```
begin
    if $y''(k)$ is in steady state with regard to $r(k)$ then
        Record $w(k)$ and $y''(k)$ during $T$ seconds;
        Compute $\bar{y}''$ according to (4.5);
        Compute $y_1''(k)$ according to (4.7);
        NII stage:
            Obtain a set of $y_j(k)$ from $y_1''(k)$ and $w(k)$ using Algorithm 2;
            Compute $\Upsilon(k)$ according to equation (4.22);
            Compute $\bar{\omega}_j(0)$ according to equation (4.25);
        end
        Execute BSA, using $\bar{\omega}_j(0)\delta(k)$ and $\Upsilon(k)$ to find $M_e(z)$ based on (4.10),
        (4.28) and (4.29).
    end if
end
```

4.3 SUMMARY

This chapter presents two countermeasures intended to contribute to the security of NCSs in case of failure or absence of conventional security mechanisms – such as encryption, authentication, network segmentation, etc. Specifically, these countermeasures target the mitigation of the system identification attacks described in Section 3.1, and the SD-Controlled Data Injection attack described in Section 3.2.1.

Concerning the System Identification attacks described in Section 3.1, the first countermeasure consists of a switching controller design that aims to meet, simultaneously, two objectives:

Objective I - Comply with the plant control requirements;

Objective II - Hinder the identification of the controller, so that the model obtained by the attacker is imprecise or ambiguous, in such a way that the attacker hesitates to launch covert or model-based attacks against the NCS.

To achieve both objectives, the switching controller uses a random switching rule described by a Markov chain where the switching events follow a specific PDF configured by two parameters: the minimum number of sampling intervals that the system have to remain in the same state; and the maximum number of sampling intervals that the system can remain in the same state.

The second countermeasure proposed in this chapter aims to identify the LTI attack function executed by the SD-Controlled Data Injection attacks described in Section 3.2.1. It consists of a link monitoring strategy that uses white gaussian noise to excite the attack function and, thus, produce signals with the information necessary for the identification process. Its is demonstrated that in normal operating conditions – *i.e.* without attack – the injected white gaussian noise is cancelled and does not affect the plant output. The injected white gaussian noise only manifests itself in the plant output when an attack is present. To increase the accuracy of this countermeasure, this chapter introduces the NII technique, which is developed using the radar pulse integration process as inspiration. It is proven that the NII technique is able to reveal the impulse response of the attack based on the signals produced by the white gaussian noise injected in the NCS.

Chapter 6 presents simulation results of the two countermeasures described in the present chapter.

5 EVALUATION ON THE ATTACKS

This chapter presents the results obtained through the joint operation of the System Identification attacks introduced in Section 3.1, with the covert/model-based attacks described in Section 3.2. The aim of these simulations is to study and evaluate how effective and accurate are those covert/model-based attacks when supported by the proposed System Identification attacks. It is also evaluated the resilience of these metaheuristic based System Identification attacks when they are impaired by data loss or noise. The simulations explore two types of plants, namely: a DC motor – which has broad applications in industry and real world systems; and a large Pressurized Heavy Water Reactor (PHWR) – which is an example of a nuclear critical infrastructure. The rest of this chapter is organized as follows:

- Section 5.1 evaluates the joint operation of the Passive System Identification attack with an SD-Controlled Data Injection offensive.
- Section 5.2 analyses the joint operation of the Passive System Identification attack with an SD-Controlled Data Loss offensive.
- Section 5.3 evaluates the joint operation of the Passive System Identification attack with a Covert Misappropriation offensive.
- Section 5.4 examines the performance of the SD-Controlled Data Injection attack when supported by the Active System Identification attack.

5.1 PASSIVE SYSTEM IDENTIFICATION WITH SD-CONTROLLED DATA INJECTION ATTACK

This section presents the results obtained through simulations that combine the Passive System Identification attack with a physically covert SD-Controlled Data Injection attack. First, Section 5.1.1 describes the model of the attacked system. Then, Section 5.1.2 presents the results obtained by the Passive System Identification attack. After that, Section 5.1.3 evaluates the results achieved by simulations of physically covert SD-Controlled Data Injection attacks, planned based on the data gathered by the Passive System Identification attack.

5.1.1 The Attacked System: DC Motor

The attacked NCS has the same architecture of the NCS shown in Figure 1, and consists of a Proportional-Integral (PI) controller that controls the rotational speed of a DC motor. This example is chosen due to the use of DC motors in a vast number of real world control systems. Moreover, DC motors have been widely used in previous works about NCS (CHEN; SONG; YU, 2012; SA; CARMO; MACHADO, 2017c; LONG; WU; HUNG, 2005; SHI; HUANG; YU, 2013; SI et al., 2010). It is noteworthy that the model herein chosen as an example does not exhaust the potential targets for this attack. NCSs composed by another kinds of LTI devices may also be a target. However, it must be taken into account that the computational cost of the attack, when launched over different LTI systems, may vary with the number of their unknown coefficients – *i.e.* the number of dimensions of the search space explored by the optimization algorithms.

The PI control function $C(z)$ and the DC motor transfer function $G(z)$ are the same as in (LONG; WU; HUNG, 2005). The equations of this NCS are represented by (5.1):

$$C(z) = \frac{c_1 z + c_2}{z - 1} \qquad\qquad G(z) = \frac{g_1 z + g_2}{z^2 + g_3 z + g_4} \tag{5.1}$$

wherein $c_1 = 0.1701$, $c_2 = -0.1673$, $g_1 = 0.3379$, $g_2 = 0.2793$, $g_3 = -1.5462$ and $g_4 = 0.5646$. The sample rate of the system is 50 samples/s and the set point $r(k)$ is an unitary step function. The network delay is not taken into account in the simulations shown in Sections 5.1.2 and 5.1.3.

5.1.2 Results of the Passive System Identification Attack

In this Section, the performance of the Passive System Identification attack is evaluated through a set of simulations performed in MATLAB. The SIMULINK tool is used to compute the output $\hat{o}_j$ of the estimated models, whose coefficients are the coordinates of an individual j of the BSA – as defined in Section 3.1.1.

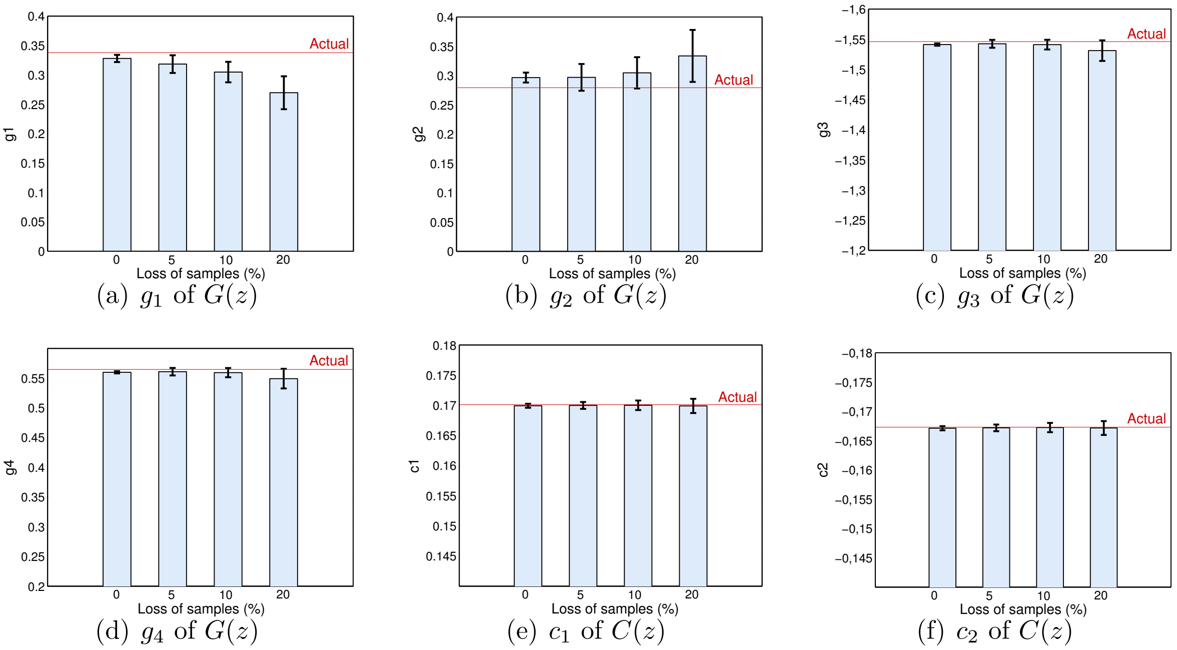
The structure of the equations represented in (5.1) are previously known by the attacker once that, as a premise, it is known that the target is an NCS that controls a DC motor using a PI controller. In these simulations, the goal of the Passive System Identification attack is to discover g_1, g_2, g_3, g_4, c_1 and c_2, also taking into account scenarios in which the attacker occasionally loses samples of the forward and feedback streams.

Each time that the DC motor is turned on, the forward and the feedback streams are captured by the attacker during a period $T = 2s$. All initial conditions are considered 0, by the time that the motor is turned on. The coefficients of $G(z)$, $[g_1,g_2,g_3,g_4]$, and the coefficients of $C(z)$, $[c_1,c_2]$, are computed separately considering that, albeit the closed loop, $G(z)$ and $C(z)$ are independent transfer functions. To assess $[g_1,g_2,g_3,g_4]$, the attacker considers the forward stream as the input and the feedback stream as the output of the estimated plant. In the opposite way, to assess $[c_1,c_2]$, the attacker considers the feedback stream as the input and the forward stream as the output of the estimated controller.

To simulate the loss of samples, it is considered four different rates l of sample loss: 0, 0.05, 0.1 and 0.2. Thus, a sample is lost by the attacker every time that $l < \mathcal{L}$, where $\mathcal{L} \sim U(0{,}1)$ and U is the uniform distribution. There are executed 100 different simulations for each rate of sample loss.

In the BSA, the population is set to 100 individuals and η, empirically adjusted, is 1. To assess the coefficients of the controller $[c_1{,}c_2]$, the algorithm is executed for 600 iterations. To assess the coefficients of the plant $[g_1{,}g_2{,}g_3{,}g_4]$, the number of iterations is increased to 800, due to the higher number of dimensions of the search space in this case. The limits of each dimension of the search space are $[-10{,}10]$.

Figure 17 shows the means of 100 estimated values of g_1, g_2, g_3, g_4, c_1 and c_2, with a Confidence Interval (CI) of 95%, considering different rates of sample loss. The actual values of the coefficients of $C(z)$ and $G(z)$ are also depicted in Figure 17. Note that the scales of Figures 17(a), 17(b), 17(c) and 17(d) are different from the scales of Figures 17(f) and 17(f), due to the smaller amplitude of the CI of c_1 and c_2. In Addition, some statistics of the obtained results are presented in Table 2.



(a) g_1 of $G(z)$ (b) g_2 of $G(z)$ (c) g_3 of $G(z)$

(d) g_4 of $G(z)$ (e) c_1 of $C(z)$ (f) c_2 of $C(z)$

Figure 17 – Mean of the estimated coefficients of $G(z)$ and $C(z)$, with a CI of 95%, in face of different rates of sample loss – own figure published in (SA; CARMO; MACHADO, 2017c).

According to Table 2 the distributions of g_1, g_2, g_3 and g_4 have a high skewness, while the distributions of c_1 and c_2 have a moderate skewness. Table 2 also provides the kurtosis of all coefficients of $G(z)$ and $C(z)$. The kurtosis, computed in accordance with (SACHS, 2012), is a statistical information used to evaluate whether the distribution is tall and thin (leptokurtic) or flat (platykurtic) when compared with the normal distribution. Based on the criteria defined in (SACHS, 2012), the distributions of all coefficients of $G(z)$ and $C(z)$ are leptokurtic, which means that these distributions have more results closer to the mean than the normal distribution. However, analyzing Table 2, it is not possible to state a clear general pattern of an increasing/decreasing skewness or kurtosis, in face of the growth of sample loss.

Table 2 – Statistics of the results obtained with different rates of sample loss – own table published in (SA; CARMO; MACHADO, 2017c)

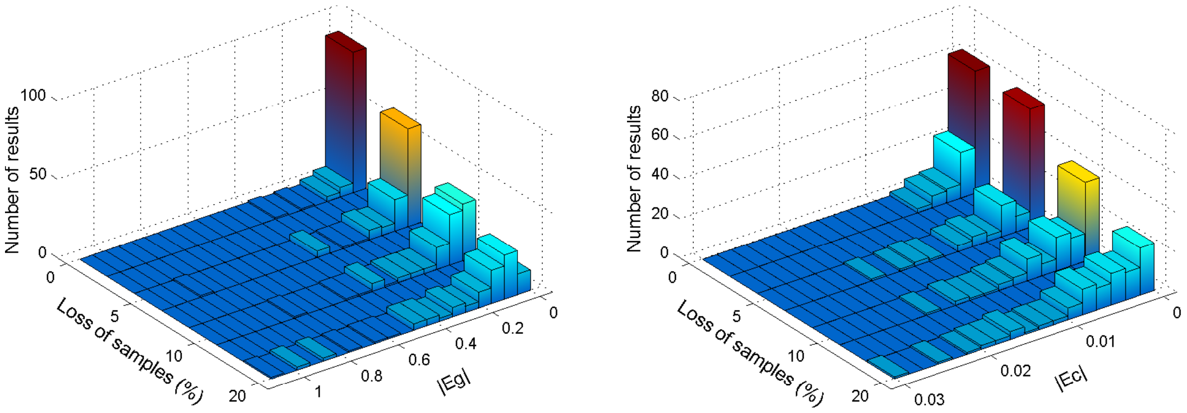
Loss of samples	Mean					
	g_1	g_2	g_3	g_4	c_1	c_2
0%	0.32793	0.29652	-1.54121	0.55983	0.16991	-0.16712
5%	0.31835	0.29689	-1.54251	0.56085	0.16997	-0.16719
10%	0.30473	0.30461	-1.54110	0.55925	0.16999	-0.16724
20%	0.26963	0.33352	-1.53119	0.54916	0.16989	-0.16716
Loss of samples	Standard deviation					
	g_1	g_2	g_3	g_4	c_1	c_2
0%	0.03097	0.04288	0.00986	0.00944	0.00167	0.00178
5%	0.07572	0.11523	0.03322	0.03194	0.00287	0.00287
10%	0.08781	0.13483	0.04076	0.03922	0.00397	0.00399
20%	0.14120	0.22378	0.08596	0.08313	0.00596	0.00598
Loss of samples	Skewness(*)					
	g_1	g_2	g_3	g_4	c_1	c_2
0%	-1.21214	1.23278	1.75298	-1.73202	-0.64331	0.79458
5%	-2.34607	1.64875	1.35284	-1.41346	-0.42288	0.36037
10%	-2.52938	1.97711	1.18018	-1.26045	-0.23379	0.13377
20%	-3.24122	1.75186	1.68335	-1.71055	-0.40055	0.37927
Loss of samples	Kurtosis(**)					
	g_1	g_2	g_3	g_4	c_1	c_2
0%	0.18846	0.19433	0.21259	0.21218	0.15119	0.16472
5%	0.08094	0.10527	0.09412	0.09802	0.02540	0.03118
10%	0.16833	0.17123	0.25041	0.24811	0.24361	0.23429
20%	0.21292	0.21127	0.25054	0.24932	0.23883	0.24441

(*) Computed in accordance with the Pearson's 2^{nd} Coefficient of Skewness.

(**) Computed in accordance with (SACHS, 2012)

In Figure 17, it is possible to verify that, in all cases, the ICs tend to grow with the increase of the sample loss. The same thing occurs with the standard deviations shown in Table 2. Regarding to the coefficients of $G(z)$, Figure 17 shows that the difference between the mean and the actual value of g_1, g_2, g_3 and g_4 also tends to raise with the increase of sample loss. It is worth mentioning that the performance of the algorithm when computing g_3 and g_4 is better then when computing g_1 and g_2, regarding the means and their CIs. This behavior results from the higher sensitivity that the output of $G(z)$ has to the variation of its poles than to the variations of its zeros. It means that, in this problem, f_j grows faster for errors in g_3 and g_4 than for errors in g_1 and g_2, making the BSA population converge more accurately in dimensions g_3 and g_4.

In Figure 17 it is also possible to note that the accuracy obtained with the coefficients of $C(z)$ is better than the accuracy of the coefficients of $G(z)$, for all rates of sample loss. The means of c_1 and c_2 are closer to their actual values, with a smaller CI. In fact, the optimization process is more effective when computing the coefficients of $C(z)$ due to its smaller search space, that has only two dimensions instead of the four dimensions of the $G(z)$ problem.



(a) Distribution of $|E_g|$

(b) Distribution of $|E_c|$

Figure 18 – Histograms of $|E_g|$ and $|E_c|$ in face of different rates of sample loss – own figure published in (SA; CARMO; MACHADO, 2017c).

As an additional metric to evaluate the performance of the algorithm, it is computed $|E_g| = |\mathcal{G}_r - \mathcal{G}_e|$ and $|E_c| = |\mathcal{C}_r - \mathcal{C}_e|$, that synthesize the error of the estimated coefficients of $G(z)$ and $C(z)$, respectively, for each solution found. $\mathcal{G}_r$ and $\mathcal{G}_e$ are vectors that contain the actual and the estimated coefficients of $G(z)$, respectively. Similarly, $\mathcal{C}_r$ and $\mathcal{C}_e$ are vectors that contain the actual and the estimated coefficients of $C(z)$, respectively. The histograms of $|E_g|$ and $|E_c|$ are presented in Figure 18, considering the mentioned rates of sample loss. The histograms graphically show that $|E_g|$ and $|E_c|$, which correspond to the modulus of the error of the estimated coefficients of $G(z)$ and $C(z)$, respectively, tend to present higher values as the loss of samples grows. It can also be confirmed by the increase of the standard deviation of the coefficients of $G(z)$ and $C(z)$ presented in Table 2. However, according to Figure 18, the mode of this errors remain close to zero for all considered rates of sample loss.

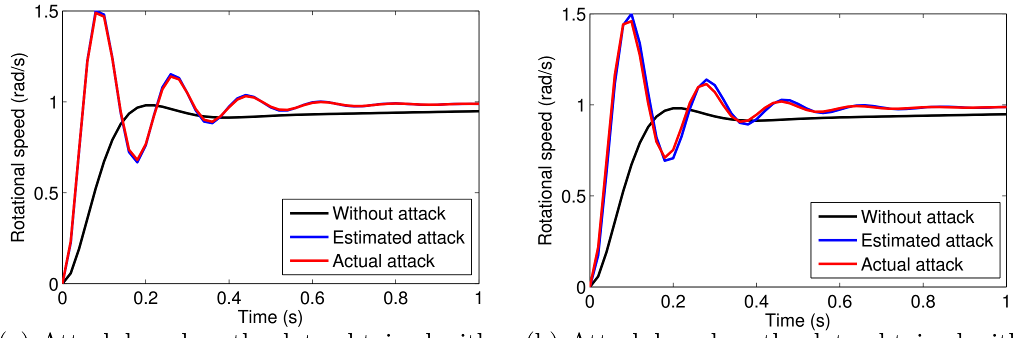
5.1.3 Results of the Service Degradation Attacks

In this section, the results obtained through simulations of SD-Controlled Data Injection attacks are presented, performed by a MitM acting in the control link of the NCS, as shown in Figure (5). The attacks were simulated in MATLAB, aiming to evaluate their accuracy when planned based on the results provided in Section 5.1.2, obtained by the Passive System Identification attack. Two sets of attack were performed. The first one, aims to cause an *overshoot* of 50% in the rotational speed of the motor. The second one, aims to cause a stationary error of -10% in the rotational speed of the motor when it is on the steady state.

In the attack aiming the overshoot, the function executed by the attacker is $M(z) = \mathcal{K}_o$. Performing a root locus analysis considering the obtained models, the attacker adjusts $\mathcal{K}_o$ to make the system underdamped, with a peak of rotational speed 50% higher than its steady state speed. The values of $\mathcal{K}_o$ are adjusted considering the average of the coefficients estimated in Section 5.1.2. Table 3 shows the values of $\mathcal{K}_o$, estimated considering different rates of sample loss during the Passive System Identification attack, as well as the overshoots obtained with the respective $\mathcal{K}_o$ in the real model. In Figure 19 it is possible to compare the response of the system without attack, with the response of the system with an attack aiming the overshoot of 50%. The curves referred as *estimated attack*, represent the results predicted by the attacker when applying the designed attack function $M(z)$ on the estimated model – *i.e.* the model discovered by the attacker through to the Passive System Identification attack. On the other hand, the curves referred as *actual attack* represent the response of the actual system in face of the same attack function $M(z)$. In other words, the curve *estimated attack* is the result achieved in a first moment, during the design stage of the attack, and the curve *actual attack* is the result obtained in a second moment, when the designed attack is launched over the actual system. It is noteworthy that the attack to the actual model – represented by the *actual attack* curve – presents, in the time domain, a response quite similar to the attack estimated with the model obtained by the Passive System Identification attack – represented by the *estimated attack* curve. This can be verified not only in the case where the system is identified with 0% of sample loss, but also in the worst considered case, *i.e.* with 20% of sample loss. It is worth mentioning that all responses presented in Figure 19 converge to 1 rad/s.

Table 3 – Values of $\mathcal{K}_o$, $\mathcal{K}_{Ess}$ and the results obtained with the attacks – own table published in (SA; CARMO; MACHADO, 2017c).

	Sample loss in the System Identification attack			
	0 %	5 %	10 %	20 %
$\mathcal{K}_o$	4.0451	4.0745	4.0828	3.796
Overshoot in the real model	48.90 %	49.43 %	49.57 %	45.94 %
$\mathcal{K}_{Ess}$	5.7471	5.7803	5.8140	5.8823
Stationary error in the real model	-10%	-10%	-9.9%	-9.8%



(a) Attack based on the data obtained without loss of samples

(b) Attack based on the data obtained with 20% of sample loss

Figure 19 – Response of the system to SD-Controlled Data Injection attacks planned to cause an overshoot of 50% in the rotational speed of the motor – own figure published in (SA; CARMO; MACHADO, 2017c).

In the attack where objective is to cause a stationary error of -10% on the rotational speed of the motor, the attacker executes (5.2):

$$M(z) = \frac{\mathcal{K}_{Ess}(z-1)}{z-0.94}, \tag{5.2}$$

wherein $\mathcal{K}_{Ess}$ is adjusted based on the data obtained with the System Identification attack. The pole of $M(z)$ is added aiming to allow a stationary error in the system. The zero of $M(z)$ is intended to format the root locus in order to guarantee the existence of a stable $\mathcal{K}_{Ess}$ that leads the system to a stationary error of -10%. Table 3 shows the $\mathcal{K}_{Ess}$ resultant from different rates of sample loss during the System Identification attack, as well as the stationary errors obtained with the respective $\mathcal{K}_{Ess}$ in the real model.

According to the data presented in Table 3, it is possible to state that the SD-Controlled Data Injection attack, designed based on the data gathered by the Passive System Identification attack, is capable to modify, in an accurate way, the response of the physical system, considering all the evaluated rates of sample loss. In the worst case, *i.e.* with 20% of sample loss, it is obtained an overshoot of 45.94% and a stationary error of -9.8%, quite close to the desired values of 50% and -10%, respectively. Such accuracy allows the attacker to keep his offensive under control, leading the system to a behavior that is predefined as physically covert and capable to degrade the service performed by the plant under attack.

These simulations provide conclusive data regarding to the effectiveness and potential impacts of the joint operation of Passive System Identification and SD-Controlled Data Injection attacks on cyber-physical systems. However, the following issues, not explored in this section, should be considered in case of actual experiments or real attacks: the presence of noise, coming from the physical process, actuator and sensors, as well as possible jitter on the network (ZHANG; GAO; KAYNAK, 2013), which might influence both the Passive System Identification and SD-Controlled Data Injection attacks; the delay unwittingly introduced by the MitM in the control loop during the SD-Controlled Data Injection, which, depending on the magnitude, may influence the system dynamics; and last, but not least, the existing techniques/systems for communication security that must be overcome to allow the attacker get access to the NCS's control loop and data.

5.2 PASSIVE SYSTEM IDENTIFICATION WITH SD-CONTROLLED DATA LOSS ATTACK

This section evaluates the performance of the joint operation of the Passive System Identification attack and the SD-Controlled Data Loss offensive. The attacked NCS consists of a DC motor and a proportional-integral (PI) controller. As in Section 5.1, this NCS example is chosen considering the application of DC motors in many real systems, as well as its common use in the literature on NCS (SA; CARMO; MACHADO, 2017c; SA; CARMO; MACHADO, 2017b). To compare with the results of the SD-Controlled Data Injection attack shown in Section 5.1, the DC motor transfer function $G(z)$ and the PI control function $C(z)$ are defined as (5.1) – the same models used in the referred section.

The controller setpoint is a unitary step function and the sample rate is 50 samples/s. The attack sequence is organized as follows:

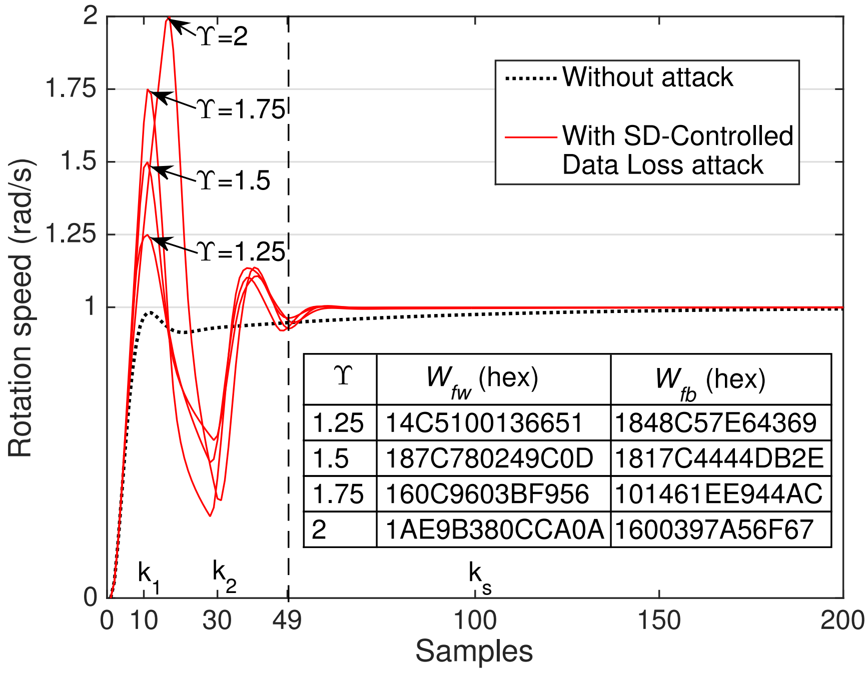
- First, the Passive System Identification attack is performed to estimate the DC motor model $G(z)$ and the control function $C(z)$;
- Then, the SD-Controlled Data Loss is performed to induce a controlled overshoot on the rotation speed of the DC motor, by causing loss of samples in the NCS. To select which samples should be lost, the attacker uses the models learned through the Passive System Identification attack.

For the sake of briefness, it is considered that the models estimated by the Passive System Identification attack are the same as those obtained by the same attack in Section 5.1.2, assuming the scenario in which the attacker does not lose samples during the system identification process. Therefore, the estimated plant model $G_e(z)$ and the estimated control function $C_e(z)$ used to design the SD-Controlled Data Loss attack are represented in (5.3):

$$C_e(z) = \frac{0.16991z - 0.16712}{z - 1} \qquad G_e(z) = \frac{0.32793z + 0.29652}{z^2 - 1.54121z + 0.55983} \tag{5.3}$$

To evaluate the accuracy and adjustability of the SD-Controlled Data Loss attack, the simulations consider four different overshoot levels, which are configured in (3.16): $\Upsilon = 1.25rad/s$, $\Upsilon = 1.5rad/s$, $\Upsilon = 1.75rad/s$ and $\Upsilon = 2rad/s$. The other parameters of (3.16) and (3.17) are configured as follows: $k_1 = 10$, $k_2 = 30$, $\mathcal{P} = 10000$, $k_s = 100$, $k_l = 200$, and y_{ss} – obtained by measuring $y(k)$ in a normal operation – is $1\ rad/s$. The beginning of the attack is triggered by the DC motor startup, which is used as reference for k_1, k_2, k_s and k_l. Also, both S_{fw} and S_{fb} begin when the DC motor starts up and are constituted by a sequence of $h = 49$ samples each.

Figure 20 shows the results obtained by the SD-Controlled Data Loss attack, considering the four different overshoot levels Υ. The response of the system without attack is also depicted in this figure for comparison. The words W_{fw} and W_{fb} – found by the BSA to cause the corresponding overshoots – are also shown in Figure 20, in hexadecimal radix. Note that, a vertical dashed line indicates the end of the period during which the attacker causes the loss of samples in the forward and feedback streams, based on W_{fw} and W_{fb}. The simulation results indicate the high degree of accuracy provided by the proposed attack, as well as its adjustability to different overshoot levels.



ϒ	W_{fw} (hex)	W_{fb} (hex)
1.25	14C5100136651	1848C57E64369
1.5	187C780249C0D	1817C4444DB2E
1.75	160C9603BF956	101461EE944AC
2	1AE9B380CCA0A	1600397A56F67

Figure 20 – Performance of the SD-Controlled Data Loss Attack for different overshoot levels Υ.

The proposed attack gives the attacker the ability to cause overshoots on the plant and accurately adjusts its intensity to the level that the attacker considers harmful/covert, depending on the characteristics of the attacked system. At the same time, the attack does not need to cause the indiscriminate loss of samples, which makes it quieter than an attack that completely denies the NCS communication. Moreover, the results indicate that the plant behavior accurately meets what the attacker planned, not evolving to an unwanted behavior that could be either extreme – which could cause the attack disclosure – or ineffective.

It is possible to compare the performance of the attack herein proposed with the performance of the SD-Controlled Data Injection attack, shown in Section 5.1, by analyzing the simulation where $\Upsilon = 1.5 rad/s$. The overshoot level caused by the attack herein proposed is $1.4994 rad/s$ while the overshoot level obtained by the SD-Controlled Data Injection attack in Section 5.1 is $1.4890 rad/s$. Based on this result, it is possible to state that the performance of the SD-Controlled Data Loss is equivalent to the performance obtained by the SD-Controlled Data Injection attack. However, the attack herein proposed does not need to overcome possibly existing security mechanisms for data integrity and authenticity. From the attacker perspective, this is an advantage of the present attack when compared to the SD-Controlled Data Injection attack.

5.3 PASSIVE SYSTEM IDENTIFICATION WITH COVERT MISAPPROPRIATION ATTACK

This section presents the results obtained through simulations that combine the Passive System Identification attack with a Covert Misappropriation attack. The results of both attacks are obtained using MATLAB/SIMULINK. Besides evaluating the ability of the Passive System Identification attack in supporting the Covert Misappropriation, this section also explores another system as target. Instead of being a DC motor, the target here is the large PHWR described in Section 5.3.1. In Section 5.3.2, the Passive System Identification attack is performed, in order provide the attacker with an estimate of the model $G'(z)$ of the attacked PHWR zone. After that, in Section 5.3.3, the Covert Misappropriation attack is carried out using the data provided by the mentioned Passive System Identification attack.

5.3.1 The Attacked System: Pressurized Heavy Water Reactor

It is known that, in nuclear power plants, the reactor power is controlled by changing the reactivity input of the reactor using reactivity control devices like control rods, liquid poisons and light water. Similarly, a Pressurized Heavy Water Reactor (PHWR) – which is fuelled by natural Uranium – is cooled and moderated by Heavy Water (D_2O, or ^2H_2O). According to (BANERJEE et al., 2015), in a nuclear reactor, the control system generates the inputs which modulate the reactivity control devices to alter the reactivity input to the reactor. An increased reactivity increases the neutron flux and hence burn-up of fissile material. On the other hand, a reduction in reactivity input reduces the burn-up of fissile material.

The literature on nuclear science (DAS et al., 2006; DASGUPTA et al., 2013; DASGUPTA et al., 2015) demonstrates the feasibility of controlling a large Pressurized Heavy Water Reactor (PHWR) through an NCS. In (DAS et al., 2006), the satisfactory control of a large PHWR is achieved using a state-feedback controller and a 100 Mbps Ethernet LAN, using UDP/IP. More recently, in (DASGUPTA et al., 2013; DASGUPTA et al., 2015), the authors demonstrate the feasibility of using PID controllers to control a large PHWR through an UDP/IP Ethernet communication.

As in (DAS et al., 2006; DASGUPTA et al., 2013; DASGUPTA et al., 2015), the reactor model used in the present simulations assumes a 540 MWe Indian PHWR. As described in (DAS et al., 2006) this 540 MWe PHWR consists of 14 zones which can be de-coupled into 14 individual Single-Input-Single-Output (SISO) Systems. For the sake of simplicity, we choose one of these 14 zones to evaluate the impact of the joint operation of the Passive System Identification attack and the Covert Misappropriation attack. The transfer function of the attacked zone and its PID controller, both obtained from (DASGUPTA et al., 2015), are defined by (5.4) and (5.5), respectively:

$$G(z) = \frac{0.0001889z}{z^2 - 1.289z + 0.2891}, \tag{5.4}$$

$$C(z) = k_p + T_s k_i \left(\frac{z}{z-1}\right) + \frac{k_d}{T_s}\left(\frac{z-1}{z}\right), \tag{5.5}$$

wherein the sample time is $T_s = 500ms$, $k_p = 348.52$, $k_i = 17.25$ and $k_d = 10.79$. This plant transfer function has been derived from practical plant data (DASGUPTA et al., 2013; DASGUPTA et al., 2015). In a PHWR, the power of a specific zone is controlled by either filling in or draining out water from a compartment using a control valve. Therefore, the equation (5.4) represents the transfer function between power P and valve input v for the zone 6 of the PHWR reported in (DASGUPTA et al., 2013; DASGUPTA et al., 2015), which has a full power of $132.75MWt$. As in (DASGUPTA et al., 2015), the set point of the controller is ramped up at the rate of $0.66MWt/s$ – *i.e.* of 0.5% of the full power – for $10s$ and then kept steady. This ramping rate is the maximum allowable rate of power increase for the class of PHWR considered.

5.3.2 Results of the Passive System Identification Attack

As described in Section 3.1.1 the Passive System Identification attack aims to estimate the coefficients of the attacked plant which, according to (5.4), are: $\alpha_1 = 1.889 \times 10^{-4}$, $\beta_1 = 1.289$ and $\beta_0 = 0.2891$. The monitoring time of the attack is $T = 200s$, starting when the power P of the attacked zone begins to increase – *i.e.* when the ramp setpoint specified in Section 5.3.1 starts. The parameters of the BSA are the same as in Section 5.1.2: the population has 100 individuals; the limits of each dimension of the search space are $[-10{,}10]$; and η – that establishes the amplitude of the movements of the individuals of the BSA – is set to 1. The accuracy of the Passive System Identification attack is evaluated considering three different numbers of iterations of the BSA: 200, 400 and 600. For each number of iterations, there were executed 100 attack simulations.

The statistics of the Passive System Identification attack in the PHWR zone are shown in Table 4. It is possible to see that, when the attacker increases the number of BSA iterations, he/she improves the performance of the attack – the mean estimated coefficients become closer to their actual values and the standard deviation decreases. Note that, a high level of accuracy is achieved when the attacker runs the BSA for 600 iterations.

Table 4 – Statistics of the Passive System Identification attack in the PHWR zone – own table published in (SÁ; CARMO; MACHADO, 2018).

BSA Iterations	Mean			Standard Deviation		
	α_1 ($\times 10^{-4}$)	β_1	β_0	α_1 ($\times 10^{-6}$)	β_1 ($\times 10^{-2}$)	β_0 ($\times 10^{-2}$)
200	4.331	-0.383	-0.617	87.64	31.18	31.18
400	2.013	-1.242	0.242	19.55	7.51	7.51
600	1.890	-1.289	0.288	0.40	0.15	0.15

5.3.3 Results of the Covert Misappropriation attack

To evaluate how the accuracy of the Passive System Identification attack may contribute for the covertness of the misappropriation attack, $G'(z)$ is configured with the mean estimated coefficients shown in Table 4. Recall that the architecture of the Covert Misappropriation attack is shown in Figure 6. Here, the aim of the Covert Misappropriation attack is to reduce 1MWt of attacked zone power, modifying as less as possible the controller input signal $y'(k)$ (comparing with a normal operation scenario). The input $\lambda(k)$ of the MitM is a ramp signal that starts at $30s$, decreases at the rate of -0.2 during $5s$ and then is kept steady. The covert controller $A(z)$ computes the same PID function defined in (5.5), however, using the following configuration: $k_p = 310$, $k_i = 40$ and $k_d = 10$.

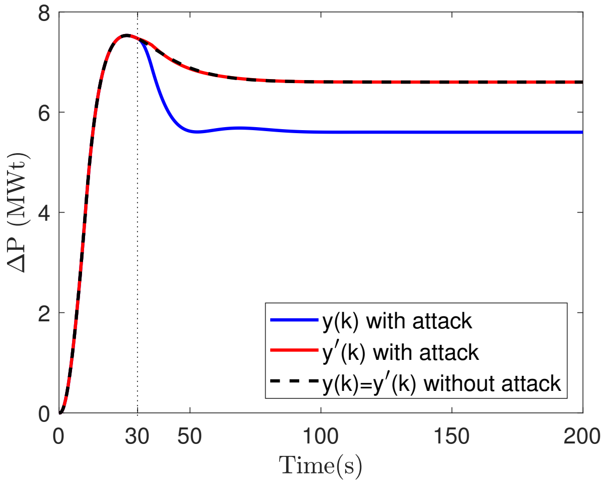


Figure 21 – PHWR responses with and without the Covert Misappropriation attack ($G'(z)$ estimated by 200 BSA iterations) – own figure published in (SÁ; CARMO; MACHADO, 2018).

Figure 21 shows the responses of the PHWR zone with and without the influence of the Covert Misappropriation attack, considering the worst estimated model – *i.e.* when $G'(z)$ is estimated through 200 BSA iterations. The time when the covert misappropriation begins is indicated by the dotted line, placed at $30s$. It is possible to see that the attacker is able to make the output $y(k)$ of the plant converge to a power $1MWt$ lower than in its normal operation (*i.e.* without the Covert Misappropriation attack). Additionally, by comparing the controller input signals $y'(k)$ with and without the attack, it is possible to verify that both are quite similar. It indicates the high degree of covertness achieved using the model estimated by the Passive System Identification attack – even executing only 200 BSA iterations. When $G'(z)$ is estimated using 400 and 600 iterations, the covertness of the misappropriation attack is better than the covertness obtained with 200 iterations. The difference between $y'(k)$ with attack and $y'(k)$ without attack decreases as the number of BSA iterations increases. It is difficult to perceive the differences of covertness if the three cases (using 200, 400 and 600 iterations) are represented as in Figure 21. Thus, to compare the covertness of these three attack conditions, we compute $\xi(k)$ (5.6):

$$\xi(k) = y'_A(k) - y'_N(k). \tag{5.6}$$

wherein $y'_A(k)$ and $y'_N(k)$ are the controller input signal $y'(k)$ with and without the Covert Misappropriation attack, respectively. Figure 22 shows the differences $\xi(k)$ in the controller input, considering Covert Misappropriation attacks where $G'(z)$ is estimated through 200, 400 and 600 BSA iterations.

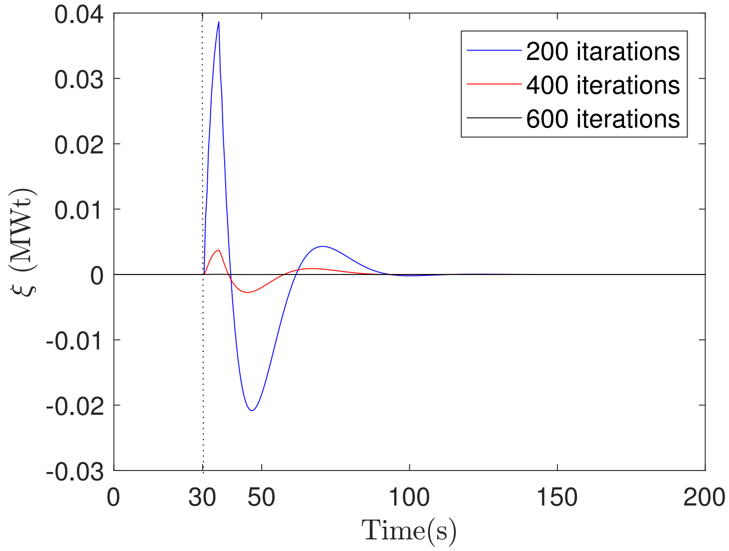


Figure 22 – Differences in the controller's input signal – own figure published in (SÁ; CARMO; MACHADO, 2018).

Note in Figure 22 that the highest amplitude of $\xi(k)$ is obtained when $G'(z)$ is estimated with 200 iterations. With 200 iterations $\max|\xi(k)| = 3.9 \times 10^{-2} MWt$ (during the transient regime of the attack), while with 400 iterations $\max|\xi(k)| = 3.8 \times 10^{-3} MWt$. From the attacker point of view, the best covertness is achieved when $G'(z)$ is estimated with 600 iterations. In this case, $\max|\xi(k)| = 2.9 \times 10^{-5} MWt$, which is a quite small deviation in the controller input, considering the magnitude of the zone power.

These results provide an idea on how covert and harmful may be the joint operation of the these two attacks in a PHWR. The attacker is able to achieve his/her goal, reducing 1MWt of attacked zone power, while causing low levels of $\xi(k)$ – especially when the Passive System Identification attack is performed with 600 iterations. These low levels of $\xi(k)$ may be considered in the development of standards and requirements for PHWR monitoring systems.

5.4 ACTIVE SYSTEM IDENTIFICATION WITH SD-CONTROLLED DATA INJECTION ATTACK

As described in Sections 2.1 and 3.1.2, the Active System Identification attack is an alternative to the Passive System Identification attack, when the attacker cannot wait so long for a signal that carry meaningful information for the identification process. In this sense, this Section aims to evaluate the performance of the Active System Identification attack and its importance in the design of a covert/model-based attack.

It is clear from Section 3.1 that both Passive and Active System Identification attacks proposed in this work rely on metaheuristic-based algorithms to iteratively find the model of the attacked systems. In Sections 5.1, 5.2 and 5.3, the Passive Identification attack is implemented and evaluated using the BSA metaheuristic. In this section, the Active System Identification attack is implemented and evaluated using two different matahеutistics, namely: the BSA; and the PSO. In Section 5.4.1, the results obtained by both BSA-based and PSO-based Active System Identification attacks are analyzed, in order to provide a demonstration of the degree of accuracy that the attacker may obtain with the proposed attack. Additionally, to increase model accuracy, the results of the Active System Identification attack are submitted to a process for eliminating outliers. Section 5.4.2 presents a data injection attack designed based on the models estimated by the Active System Identification attack. The purpose of these simulations is to demonstrate how an Active System Identification attack may contribute for the accuracy of model-based attacks.

5.4.1 Active System Identification Attack

The targeted system, shown in Figure 23, consists of a DC motor whose rotational speed is controlled by a Proportional-Integral (PI) controller – as the system attacked in Section 5.1. The PI control function $C(z)$ and the DC motor transfer function $P(z)$, obtained from (LONG; WU; HUNG, 2005), are represented by (5.7) and (5.8), respectively:

$$C(z) = \frac{0.1701z - 0.1673}{z - 1}, \tag{5.7}$$

$$P(z) = \frac{0.3379z + 0.2793}{z^2 - 1.5462z + 0.5646}. \tag{5.8}$$

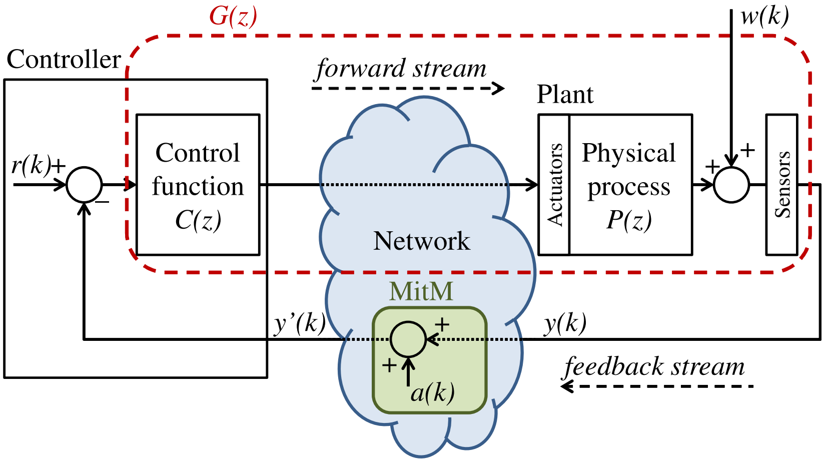


Figure 23 – Attack on a noisy NCS – own figure published in (SA; CARMO; MACHADO, 2017b).

Recall that in this attack, according to Section 3.1.2, the data is collected at only one point of the NCS (be it in the forward or in the feedback stream). In the present simulations, the data is collected only at the feedback stream. Thereby, the transfer function to be identified $G(z)$ – which is also the open-loop transfer function of the NCS – is defined by (5.9):

$$G(z) = C(z)P(z) = \frac{g_1 z^2 + g_2 z + g_3}{z^3 + g_4 z^2 + g_5 z + g_6}, \tag{5.9}$$

wherein $g_1 = 0.0575$, $g_2 = -0.0090$, $g_3 = -0.0467$, $g_4 = -2.5462$, $g_5 = 2.1108$ and $g_6 = -0.5646$. The sample rate of the system is 50 samples/s and the set point $r(k)$ is an unitary step function. Network delay and packet loss are not taken into account in these simulations.

The structure of the equations (5.1), and so the structure of (5.9), are previously known by the attacker once that, as a premise, it is known that the target is an NCS that controls a DC motor using a PI controller. Thus, in these simulations, the goal of the Active System Identification attack is to discover g_1, g_2, g_3, g_4, g_5 and g_6.

The chosen attack signal $a(k)$ is a discrete-time unit impulse (5.10):

$$a(k) = \begin{cases} 1 & \text{if } k = k_a; \\ 0 & \text{otherwise,} \end{cases} \tag{5.10}$$

wherein k_a is the single sample in which the attacker interfere in the system by adding 1 to the feedback stream. Note that the discrete-time unit impulse is chosen to excite the NCS due to its short active time – *i.e.*, one sample –, which increases the stealthiness of the attack in the time domain. Moreover, the Fourier transform of an impulse function has an uniform – flat – density in the frequency domain, which is easily masked by the frequency distribution of a white Gaussian noise. This fact also increases the stealthiness of the attack signal in the frequency domain.

The effectiveness of the Active System Identification attack is evaluated with and without noise. To simulate the noise, $w(k) \sim N(\mu,\sigma)$ is inserted in the NCS as indicated in Figure 23. Note that $w(k)$ is a white Gaussian noise wherein N is a normal distribution, μ is its mean and σ is its standard deviation. In all simulations, the mean is $\mu = 0 \; rad/s$. The standard deviation is adjusted in such manner that 95% of the amplitudes of $w(k)$ are within $\pm I$ ($I = 2\sigma$). The simulations consider four different noise intensities I: 0 (no noise), 0.0025 rad/s, 0.005 rad/s and 0.01 rad/s. For each noise intensity I, 100 different simulations are executed using each of the mentioned metaheuristics. In each simulation, the feedback stream is captured by the attacker during a period $T = 2s$ (100 samples), starting at sample $k_a + 1$.

The attack model was implemented in MATLAB, where the simulations were carried out. The SIMULINK tool was used to compute $y_a(k)$ and $\hat{y}_{aj}(k)$ – the latter, for each individual j of the optimization algorithms. The parameters of the BSA and PSO were empirically adjusted through a set of simulations without noise ($I = 0$). These parameters are then used for all noise conditions. In the BSA-based attacks, the parameter η – that establishes the amplitude of the movements of the BSA individuals – is set to 1. In the PSO-based attacks, the following parameters are used:

- Inertial coefficient: $\omega = 0.4$,
- Cognitive coefficient: $\varphi_1 = 1.5$
- Social Coefficient: $\varphi_2 = 1.5$
- Velocity limit coefficient: $\delta = 0.1$

In both algorithms, the population is set to 100 individuals and the limits of each dimension of the search space are $[-10,10]$. In each simulation, the BSA and the PSO are executed for 4500 iterations.

Let S_u be the solution of an attack simulation u, and $g_{i,u}$ the value estimated for the i^{th} coefficient of $G(z)$ in the u^{th} attack simulation. Each attack simulation provides a solution $S_u = [g_{1,u},g_{2,u},g_{3,u},g_{4,u},g_{5,u},\ g_{6,u}]$ containing estimated values for the six coefficients of $G(z)$. In a preliminary work (SA; CARMO; MACHADO, 2017a) – published as part of this research –, for a given coefficient g_i of $G(z)$, if an estimated value $g_{i,u}$ was beyond two standard deviation from the mean, then $g_{i,u}$ was considered an outlier and eliminated from the set of values found for g_i. After that, the estimated value of each g_i was assumed to be the mean of the remaining $g_{i,u}$. However, in the present work, as in (SA; CARMO; MACHADO, 2017b)[1], the process for eliminating outliers is modified to improve the accuracy of the estimated model. In this work, if an estimated value $g_{i,u}$ is beyond two standard deviation from the mean, the whole solution S_u (to which $g_{i,u}$ belongs) is considered as an outlier and eliminated from the set of solutions. Doing so, the estimated value of each g_i is assumed to be mean of all $g_{i,u}$ contained in the set of remaining S_u. Table 5 presents a summary that compares the results achieved with the outliers elimination process used in (SA; CARMO; MACHADO, 2017b) with the results obtained in (SA; CARMO; MACHADO, 2017a), in both BSA-based and PSO-based attacks. The most accurate results are highlighted. Note that in all cases the most accurate results were achieved by the BSA-based attacks. According to Table 5, the outliers elimination process used (SA; CARMO; MACHADO, 2017b), in general, improves the accuracy of the results obtained by the BSA-based attacks. This improvement is more evident in Section 5.4.2, where the performance of other attacks designed with the data presented in Table 5 is analyzed. Note that, in Table 5, for the PSO-based attacks, the results obtained in (SA; CARMO; MACHADO, 2017b)

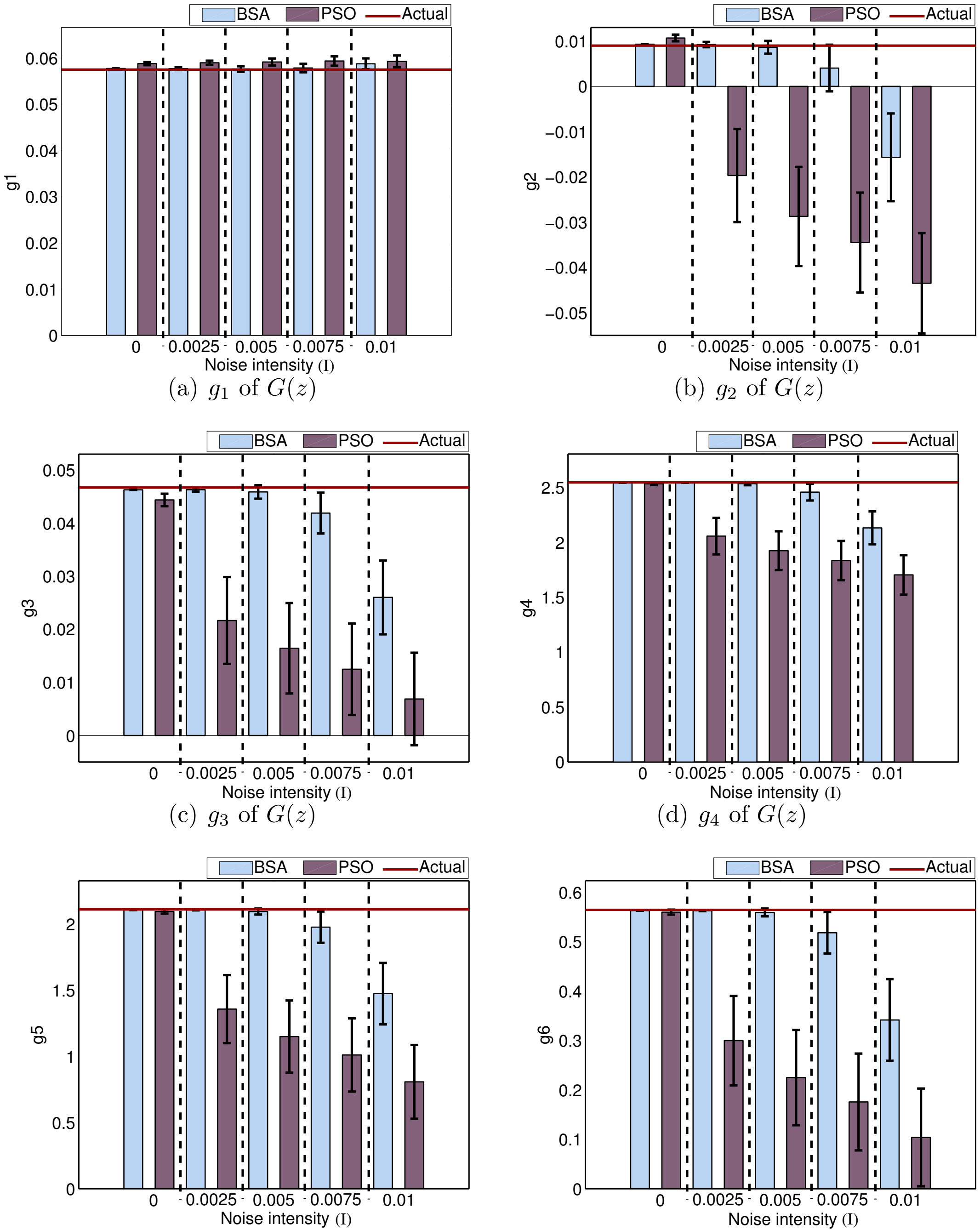
[1] The paper (SA; CARMO; MACHADO, 2017b) constitutes a part of the present research and is an extended version of (SA; CARMO; MACHADO, 2017a).

are the same as the results obtained in (SA; CARMO; MACHADO, 2017a). It occurs because in PSO-based attacks all outlier coefficients belong to solutions wherein all other coefficients are also outliers – *i.e.* beyond two standard deviations from their means. Thus, in the PSO-based attacks, the whole solution S_u which contains an outlier is eliminated from the set of solutions even when the outliers elimination process of (SA; CARMO; MACHADO, 2017a) is applied.

Table 5 – Mean estimated coefficients of $G(z)$ after the processes to eliminate outliers – own table published in (SA; CARMO; MACHADO, 2017b).

			Mean of the estimated coefficients					
		Noise	g_1	g_2	g_3	g_4	g_5	g_6
		(I)	$\times 10^{-2}$	$\times 10^{-3}$	$\times 10^{-2}$			$\times 10^{-1}$
Outliers eliminated as in (SA; CARMO; MACHADO, 2017a)	BSA	0	5.7756	-9.3337	-4.6261	-2.5431	2.1063	-5.6319
		0.0025	5.7736	-9.2001	-4.6301	-2.5428	2.1058	-5.6305
		0.005	5.7826	-9.0411	-4.5528	-2.5345	2.0937	-5.5924
		0.0075	5.8215	-0.7908	-3.4930	-2.4023	1.8911	-4.7857
		0.01	5.8561	20.7982	-2.5371	-2.0906	1.3852	-3.1095
	PSO	0	5.8799	-10.6784	-4.4361	-2.5341	2.0940	-5.5989
		0.0025	5.8987	19.7038	-2.1653	-2.0568	1.3567	-2.9982
		0.005	5.9148	28.7309	-1.6431	-1.9242	1.1493	-2.2507
		0.0075	5.9357	34.5026	-1.2472	-1.8347	1.0102	-1.7552
		0.01	5.9288	43.4950	-0.6878	-1.7036	0.8073	-1.0370
Outliers eliminated as in (SA; CARMO; MACHADO, 2017b)	BSA	0	5.7750	-9.3128	-4.6268	-2.5431	2.1063	-5.6319
		0.0025	5.7714	-9.2299	-4.6294	-2.5428	2.1059	-5.6306
		0.005	5.7628	-8.6145	-4.5870	-2.5350	2.0944	-5.5931
		0.0075	5.7843	-4.0346	-4.1886	-2.4578	1.9761	-5.1824
		0.01	5.8763	15.6817	-2.6009	-2.1322	1.4738	-3.4164
	PSO	0	5.8799	-10.6784	-4.4361	-2.5341	2.0940	-5.5989
		0.0025	5.8987	19.7038	-2.1653	-2.0568	1.3567	-2.9982
		0.005	5.9148	28.7309	-1.6431	-1.9242	1.1493	-2.2507
		0.0075	5.9357	34.5026	-1.2472	-1.8347	1.0102	-1.7552
		0.01	5.9288	43.4950	-0.6878	-1.7036	0.8073	-1.0370

The mean estimated values of g_1, g_2, g_3, g_4, g_5 and g_6, after applying the outliers elimination process of (SA; CARMO; MACHADO, 2017b), are shown in Figure 24 with a Confidence Interval (CI) of 95%, for different values of noise intensity I. Note that the actual values of these coefficients are also depicted in Figure 24. In this Figure, it is possible to compare the results achieved by the BSA-based and the PSO-based attacks. According to Figure 24, it is possible to verify that, for all coefficients of $G(z)$, both BSA-based and PSO-based attacks present good accuracy when $I = 0$ (*i.e.* without noise, the mean values of the estimated coefficients are close to their actual values). Despite the similar and accurate performance of the two metaheuristics without noise, it is possible to state that the BSA presented a slightly better performance than the PSO in this noise condition ($I = 0$), specially with regard to the coefficients g_1, g_2 and g_3. Note that the



(a) g_1 of $G(z)$

(b) g_2 of $G(z)$

(c) g_3 of $G(z)$

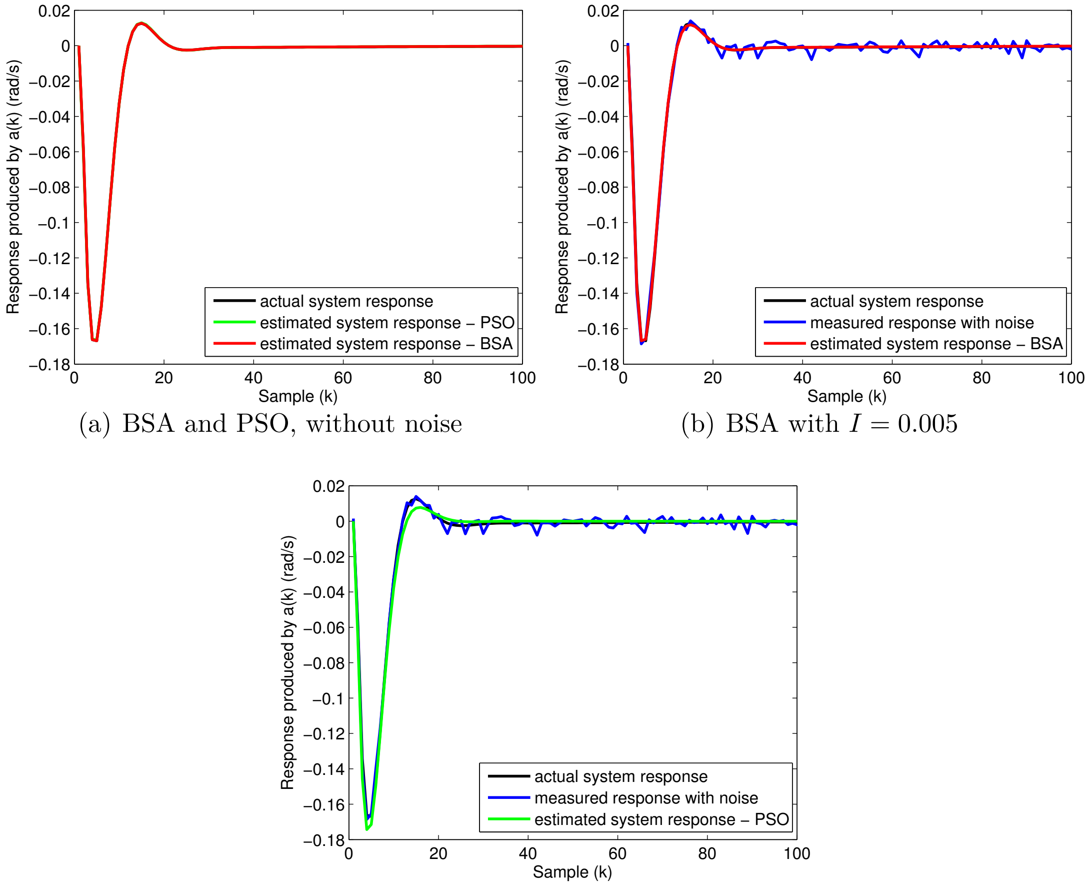
(d) g_4 of $G(z)$

(e) g_5 of $G(z)$

(f) g_6 of $G(z)$

Figure 24 – Mean of the estimated coefficients of $G(z)$, with CI of 95%, in face of different noise intensities I – own figure published in (SA; CARMO; MACHADO, 2017b).

performance of the PSO-based attack is degraded when noise is added to the system. This performance degradation of the PSO occurs for $I \geq 0.0025$ and tends to be more expressive with the increase of I. On the other hand, it is possible to verify in Figure 24 that the BSA-based attack still present good accuracy for noise intensities up to 0.005. When $I \leq 0.005$, all coefficients estimated by the BSA-based attack present a mean close to their actual values and with a small CI. When $I \geq 0.0075$, the performance of the BSA-based attack decreases with the raise of noise in a more expressive way, being at its worst when $I = 0.01$. In general, among the six coefficients of $G(z)$, the estimation of g_2 presents the lowest accuracy for both BSA-based and PSO-based attacks. This behavior is attributed to a lower sensitivity that the output $\hat{y}_a(k)$ of the estimated system has to the variation of g_2. This means that, in this problem, f_j grows faster for errors in g_1, g_3, g_4, g_5 and g_6 than for errors in g_2, making the BSA population converge less accurately in dimension g_2.



(a) BSA and PSO, without noise

(b) BSA with $I = 0.005$

(c) PSO with $I = 0.005$

Figure 25 – Response of actual and estimated systems produced by $a(k)$, in face of different noise intensities – own figure published in (SA; CARMO; MACHADO, 2017b).

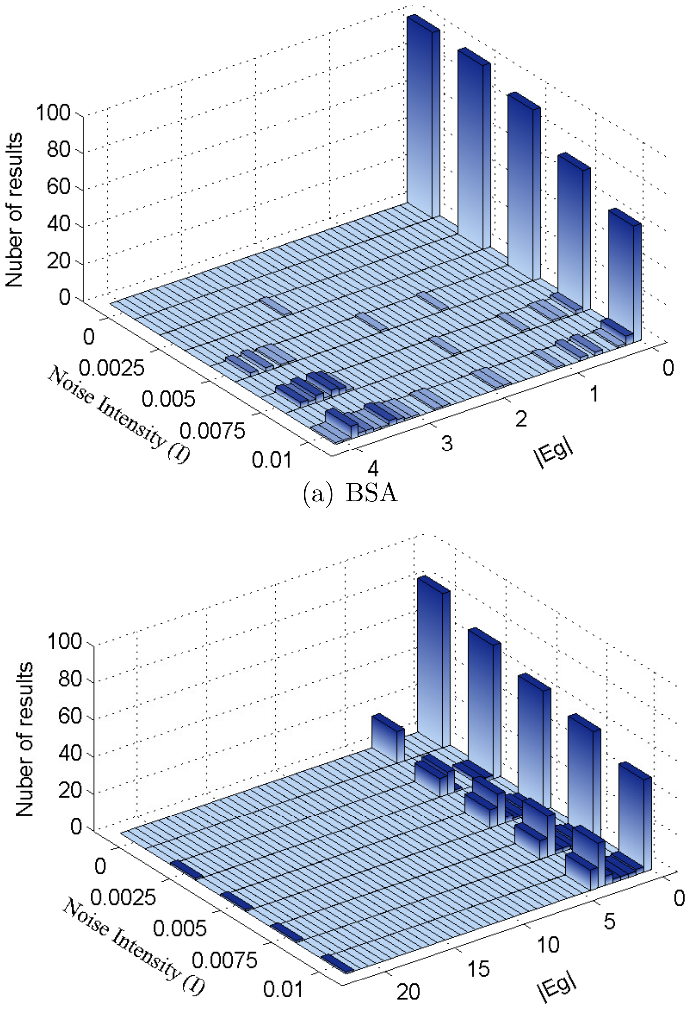
The performance of the attacks can also be evaluated in the k domain through the examples provided in Figure 25, considering two different intensities of noise: without noise, in Figure 25(a); and with $I = 0.005$, in Figures 25(b) and 25(c). Figure 25(a) shows that, without noise, the response of the system estimated by both BSA-based and PSO-based attacks matches the response of the actual system with high accuracy. In Figure 25(b), even with a noise intensity of $I = 0.005$, the response of the system estimated by the BSA-based attack still matches the response of the actual system, indicating the convergence of $G_e(z)$ to $G(z)$ and ratifying the statistics shown in Figure 24 for the BSA with such noise intensity. On the other hand, when applying the PSO-based attack with the same noise, as exemplified in Figure 25(c), there is a slight difference between the response of the estimated system and the response of the actual system, produced by the mismatch of the estimated coefficients in the presence of such noise intensity. This exemplifies the worse performance of the PSO-based attacks, when compared with the BSA-based attacks, in face of the same noise intensities.

To synthesize the error of each solution found, $|E_g|$ is computed as (5.11):

$$|E_g| = \sqrt{\sum_{i=1}^{6} (g_i - g_{ei})^2}, \tag{5.11}$$

wherein g_i and g_{ei} are the actual and estimated coefficients of the attacked system, respectively, and i is the index number of each of the six coefficients of the model being assessed. Note that $|E_g|$ is the module of a vector composed by the error of each coefficient found, which represents another metric to evaluate the performance of each attack. The histograms of $|E_g|$ are presented in Figure 26, considering the mentioned noise intensities. It graphically shows that higher values of $|E_g|$ tend to appear more frequently as the noise intensity grows, in both BSA-based and PSO-based attacks. However, based on these histograms it is possible to verify that the mode of $|E_g|$ is close to zero for all noise intensities, using both metaheuristics. This indicates that, even in the presence of noise, most solutions present low deviations from the actual coefficients. Note that, for all noise intensities, the BSA-based attacks provide more results in the modal class – where $|E_g|$ is close to zero – than the PSO-based attacks. Moreover, the worst results of the BSA-based attacks have an $|E_g|$ of about 4 when $I \geq 0.005$, while the worst results of the PSO-based attacks have an $|E_g| > 20$ when $I \geq 0.0025$. These results, together with the statistics shown in Figure 24, indicate that the performance of the Active System Identification attack is better when implemented with the BSA than with the PSO. It is worth mentioning that, to achieve these results, the BSA-based attacks consumed an average processing time $(6.68 \pm 0.47)\%$ higher than the PSO-based attacks.

In general, the outcomes indicate that, for the same amplitude of attack signal $a(k)$, the performance of the attack tends to decrease as the noise intensity increases (*i.e.* when the attack signal-to-noise ratio decreases). The minimum length of the attack signal in terms of number of manipulated samples (*i.e.* one single sample) improves the stealthiness of the attack in the k domain. On the other hand, a minimum attack signal-to-noise ratio required to guarantee the performance of this attack is a drawback with respect to its stealthiness, from the attacker's point of view. This issue makes more difficult for the attacker to approximate the amplitude of $a(k)$ to the noise amplitude or to noise values that have higher probability to occur, which should help to increase the stealthiness of the attack signal in terms of amplitude.



(a) BSA

(b) PSO

Figure 26 – Histograms of $|E_g|$ for different noise intensities – own figure published in (SA; CARMO; MACHADO, 2017b).

5.4.2 Data Injection Attack

The proposed Active System Identification attack is an useful tool – from the attacker point of view – for the design of other sophisticated and accurate attacks. To demonstrate this capability, this section presents a set of data injection attacks designed based on the models estimated in Section 5.4.1 by the Active System Identification attacks. These data injection attacks aim to cause an overshoot of 50% on the rotational speed of the DC motor during its transient response. As mentioned in Sections 2.1 and 3.2, this physically covert interference (SA; CARMO; MACHADO, 2017c) may cause stress and possibly damages to the plant, reducing its MTBF.

Aware of the estimated model of the NCS, the attacker – acting as an MitM – executes the attack function defined by (5.12):

$$y'(k) = \gamma_1 y(k-1) + \gamma_2 y'(k-1). \tag{5.12}$$

wherein γ_1 and γ_2 are adjusted through a root locus analysis, considering the estimated open-loop transfer functions. Note that in (5.12) the attacker is on the NCS's feedback stream, given that, according to Figure 23, $y(k)$ is the sensor's output and $y'(k)$ is the controller's input.

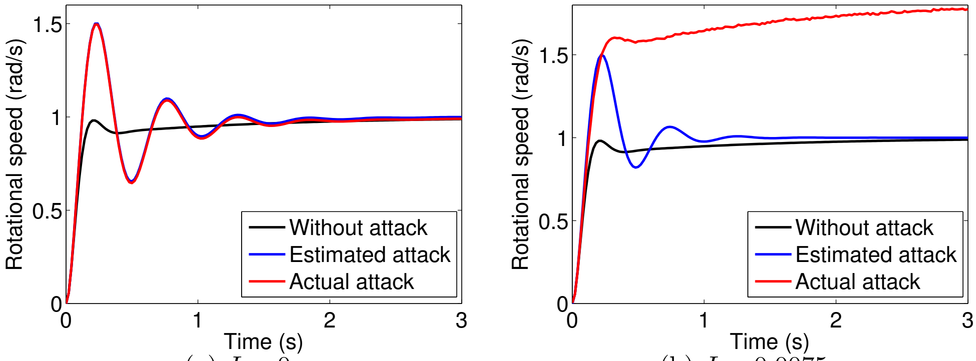
The models used to design these data injection attacks are built with the mean estimated coefficients shown in Table 5. Note that γ_1 and γ_2 have to be adjusted for each estimated model which, in turn, vary with the noise condition, the optimization algorithm (BSA or PSO) and the process for elimination of outliers, as shown in Table 5. The values of γ_1 and γ_2 used in each data injection attack are shown in Table 6, as well as the respective overshoots achieved with the attack. In Table 6, the row (I) contains the data injection attacks designed with the models estimated by the BSA-based attacks using the outliers elimination process of (SA; CARMO; MACHADO, 2017a). Row (II) contains the data injection attacks designed with the models estimated by the BSA-based attacks using the outliers elimination process proposed in (SA; CARMO; MACHADO, 2017b). As described in Section 5.4.1, the models estimated in (SA; CARMO; MACHADO, 2017b) and (SA; CARMO; MACHADO, 2017a) by the PSO-based attacks do not change with the different outliers elimination processes. Thus, in Table 6, the attacks designed with the models estimated by the PSO-based attacks – after either of the two outliers elimination processes – are contained in row (III).

Table 6 – Values of γ_1, γ_2 and the overshoot obtained with the data injection attacks – own table published in (SA; CARMO; MACHADO, 2017b).

		Noise (I) during the System Identification attack				
		0	0.0025	0.005	0.0075	0.01
(I)	γ_1	0.25316	0.25485	0.25523	0.58959	0.53297
	γ_2	0.74679	0.74515	0.74477	-0.07354	0.5911
	Overshoot	49.53 %	49.49 %	49.65 %	(*)	(*)
(II)	γ_1	0.25318	0.25286	0.2551	0.27652	0.31407
	γ_2	0.74682	0.74714	0.7449	0.72348	0.68593
	Overshoot	49.52 %	49.78 %	49.67 %	46.91 %	42.42 %
(III)	γ_1	0.26801	0.32328	0.32816	0.33074	0.33204
	γ_2	0.73199	0.67672	0.67184	0.66926	0.66796
	Overshoot	47.43 %	40.70 %	40.37 %	40.30 %	40.38 %

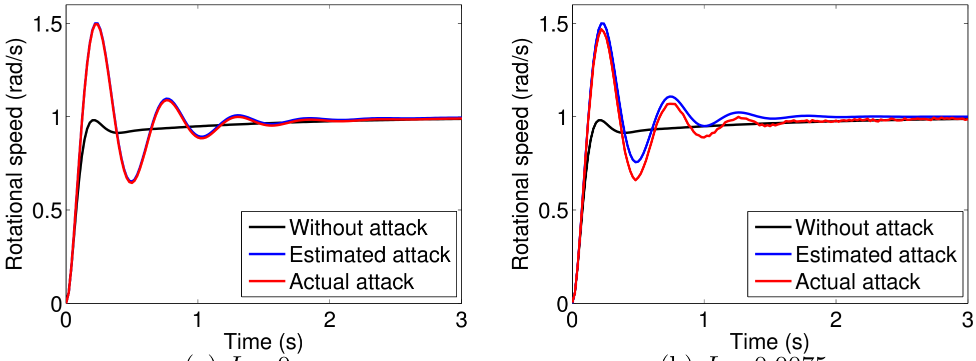
(*) The inaccuracy of the data injection attack caused a collateral effect: an expressive steady state error in the motor's rotational speed.

Examples of the data injection attacks shown in Table 6 are depicted, in the time domain, in Figures 27, 28 and 29. In these figures, the curves named as *estimated attack* represent the results predicted by the attacker when applying the designed attack function (5.12) on the estimated model – *i.e.* the model provided by the Active System Identification attack. On the other hand, the curves referred as *actual attack* represent the response of the actual system in face of the same attack function (5.12). In other words, the curve *estimated attack* is the result achieved in a first moment, during the design stage of the attack, and the curve *actual attack* is the result obtained in a second moment, when the designed attack is launched over the actual system.



(a) $I = 0$ (b) $I = 0.0075$

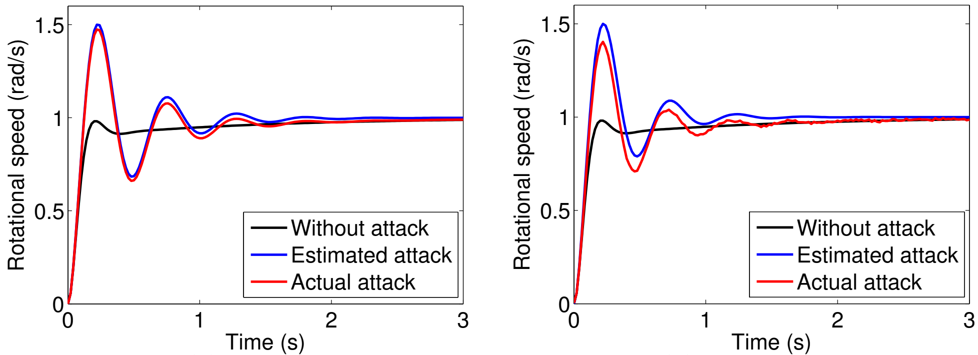
Figure 27 – Data injection attack using models estimated by a BSA-based attack with the outliers elimination process of (SA; CARMO; MACHADO, 2017a) – own figure published in (SA; CARMO; MACHADO, 2017b).



(a) $I = 0$ (b) $I = 0.0075$

Figure 28 – Data injection attack using models estimated by a BSA-based attack with the outliers elimination process of (SA; CARMO; MACHADO, 2017b) – own figure published in (SA; CARMO; MACHADO, 2017b).

In rows (I) and (II) of Table 6, it is possible to see that, when $0 \leq I \leq 0.005$, the data provided by the BSA-based Active System Identification attacks produce accurate data injection attacks, either with the outliers elimination process of (SA; CARMO; MACHADO, 2017a) or (SA; CARMO; MACHADO, 2017b). In these data injection attacks, all overshoots lie between 49.49% and 49.78% – *i.e.*, close to the goal



(a) $I = 0$ (b) $I = 0.0075$

Figure 29 – Data injection attack using models estimated by a PSO-based attack with the outliers elimination process – own figure published in (SA; CARMO; MACHADO, 2017b).

of 50%. However, for $0.0075 \leq I \leq 0.01$, the data injection attacks of row (I) – *i.e.*, using the models estimated by BSA-based attacks with the outliers elimination process of (SA; CARMO; MACHADO, 2017a) – produce a collateral behavior on the attacked system. They cause expressive steady state errors in the motor's rotational speed, as indicated, for instance, in Figure 27(b). On the other hand, for $0.0075 \leq I \leq 0.01$, when the outliers elimination process proposed in (SA; CARMO; MACHADO, 2017b) is applied to the BSA-based Active System Identification attacks, the estimated models eliminate the mentioned collateral effects on the data injection attacks. This can be seen in the example shown in Figure 28(b), for $I = 0.0075$, where the response of the actual attack is close to the response of the estimated attack, without a steady state error and with an overshoot of 46.91%. The reason for these different performances is explained by the impact of the outliers elimination process in the root locus analysis. When only an outlier coefficient $g_{i,u}$ is eliminated – as in (SA; CARMO; MACHADO, 2017a) –, instead of eliminating the whole solution S_u from where it belongs – as proposed in (SA; CARMO; MACHADO, 2017b) –, the roots of the open-loop transfer function suffer a distortion. For instance, in these simulations, when $0.0075 \leq I \leq 0.01$, the outliers elimination process of (SA; CARMO; MACHADO, 2017a) modifies a pole of $G(z)$ that should be 1. This pole exists due to the use of the PI controller – a premise known by the attacker – and, when modified, influences the adjustment of γ_1 and γ_2 of equation (5.12). On the other hand, by eliminating the whole solution S_u containing an outlier coefficient $g_{i,u}$, the mean estimated coefficients of $G(z)$ preserve the interdependencies necessary to produce less distorted roots. Note that, as shown in row (III) of Table 6 and in Figure 29, the PSO-based attacks produce less accurate data injection attacks than the BSA-based attacks with the outliers elimination process proposed in (SA; CARMO; MACHADO, 2017b). It is worth mentioning that the data injection attacks designed with the models estimated by the PSO-based attacks do not present any collateral effects, using any of the two outliers elimination processes. In both cases, as explained in Section 5.4.1, the whole solution S_u containing an outlier is eliminated from the set of solutions, producing less distortion in the roots of $G(z)$.

Moreover, with the exception of the attacks of row (I) for $0.0075 \leq I \leq 0.01$, all data injection attacks achieved satisfactory results. However, it is shown that the accuracy of the data injection attack, in general, decreases as the noise intensity increases during the Active System Identification attack.

5.5 SUMMARY

This chapter evaluates the System Identification attacks and model-based offensives described in chapter 3. The attacks are arranged in joint operations were the System Identification attacks are executed first, in order to provide the NCS models required to implement the subsequent model-based offensives.

The System Identification attacks, implemented with bioinspired metaheuristics (BSA and PSO), are evaluated in scenarios with and without data loss and noise. The results demonstrate that the proposed System Identification attacks are an effective tool to support the design of the model-based offensives addressed in this work, namely: SD-Controlled Data Injection attack; Covert Misappropriation attack; and SD-Controlled Data Loss attack. The outcomes indicate that the models learned through System Identification attacks are accurate enough build attacks considered to be physically covert (in case of SD-Controlled Data Injection attacks, or SD-Controlled Data Loss attacks), as well as cybernetically covert attacks (in case of Covert Misappropriation attacks). In general, the results indicate that the model-based attacks accurately meet what the attacker has planned, not evolving to an unwanted behavior that could be either extreme – which could cause the attack disclosure – or ineffective.

The results also demonstrate that the novel SD-Controlled Data Loss attack has a performance equivalent to the SD-Controlled Data Injection attack, however without the need to overcome possibly existing security mechanisms for data integrity and authenticity (which may be necessary in the latter). The SD-Controlled Data Loss attack demonstrates to be adjustable (obtaining accuracy for different overshoot levels) and does not cause indiscriminate loss of samples to achieve its goal.

6 EVALUATION ON THE COUNTERMEASURES

This Chapter evaluates the performance of the countermeasures proposed in Chapter 4. Sections 6.1 and 6.2 analyze the performance of the switching controller design proposed in Section 4.1 against the Active System Identification attack and the Passive System Identification attack, respectively. Section 6.3, in turn, analyzes the performance of the attack identification strategy proposed in section 4.2 when identifying the Controlled Data Injection attack described in Section 3.2.1.

6.1 MITIGATION OF THE ACTIVE SYSTEM IDENTIFICATION ATTACK

In this section, the performance of the countermeasure proposed in Chapter 4 is analyzed in face of the Active System Identification attack described in Section 3.1.2. In the simulations of this section, two NCSs are used for comparison: one with the proposed countermeasure – *i.e.* using a switching controller; and another without the proposed countermeasure – *i.e.* using a non-switching controller. The performance of the attack in both NCSs is evaluated through a set of simulations performed in MATLAB. The model of the two attacked NCSs, as well as the parameters of the Active System Identification attack, are specified in Section 6.1.1.

Recall that, as mentioned in Section 4.1.2, the design of the switching controller must meet simultaneously two objectives: comply with the plant control requirements; and hinder the identification process. In this sense, Section 6.1.2 presents the results of the switching controller as a countermeasure for the Active System Identification attack. Section 6.1.3, in turn, evaluates the performance of the proposed countermeasure from the control perspective, in order to identify possible trade-offs that may exist between the two mentioned objectives.

6.1.1 Attacked NCSs and Parameters of the Attack

The NCS without the proposed countermeasure – also referred in this section as a *system with vulnerable model* – is the same NCS attacked in Sections 5.1 and 5.4, endowed with a non-switching controller. It consists of a DC motor whose rotational speed is controlled by a Proportional-Integral (PI) controller. The DC motor's transfer function $P(z)$ and the PI control function $C_1(z)$ are represented by (6.1) and (6.2), respectively:

$$P(z) = \frac{0.3379z + 0.2793}{z^2 - 1.5462z + 0.5646}, \tag{6.1}$$

$$C_1(z) = \frac{0.1701z - 0.1673}{z - 1}. \tag{6.2}$$

Thereby, the open-loop transfer function of the *system with vulnerable model* $G_1(z)$ – to be identified – is defined as (6.3):

$$G_1(z) = C_1(z)P(z) = \frac{g_{1,1}z^2 + g_{2,1}z + g_{3,1}}{z^3 + g_{4,1}z^2 + g_{5,1}z + g_{6,1}}, \tag{6.3}$$

wherein $g_{1,1} = 0.0575$, $g_{2,1} = -0.0090$, $g_{3,1} = -0.0467$, $g_{4,1} = -2.5462$, $g_{5,1} = 2.1108$ and $g_{6,1} = -0.5646$.

The NCS endowed with the proposed countermeasure – *i.e.* the switching controller – also controls a DC motor defined by the transfer function (6.1). The switching controller switches among two control functions: $C_1(z)$, that is the same control function (6.2) of the *system with vulnerable model*; and $C_2(z)$ defined by (6.4).

$$C_2(z) = \frac{0.1208z - 0{,}1167}{z - 1}. \tag{6.4}$$

Therefore, the NCS with the switching controller is an SLS composed by two subsystems, each one having an open-loop transfer function. The two open-loop transfer functions are, respectively: $G_1(z)$, that is the same open-loop transfer function (6.3) of the *system with vulnerable model*; and $G_2(z)$ defined by (6.5),

$$G_2(z) = C_2(z)P(z) = \frac{g_{1,2}z^2 + g_{2,2}z + g_{3,2}}{z^3 + g_{4,2}z^2 + g_{5,2}z + g_{6,2}}, \tag{6.5}$$

wherein $g_{1,2} = 0.0408$, $g_{2,2} = -0.0057$, $g_{3,2} = -0.0326$, $g_{4,2} = -2.5462$, $g_{5,2} = 2.1108$ and $g_{6,2} = -0.5646$. Note that the denominators of $G_1(z)$ and $G_2(z)$ are equal, given that only the numerators of $C_1(z)$ and $C_2(z)$ are different. Thus, $g_{4,1} = g_{4,2}$, $g_{5,1} = g_{5,2}$ and $g_{6,1} = g_{6,2}$.

The control functions $C_1(z)$ and $C_2(z)$ are designed to make the two subsystems of this SLS stable. As described in Section 4.1.2, the control functions are randomly switched based on the Markov chain shown in Figure 8, under a restricted switching policy, whose restrictions are bounded by the PDF shown in Figure 9. The parameters a and b of the PDF were empirically adjusted to $a = 20$ and $b = 40$, in order to meet Objectives I and II, as discussed in Section 4.1.2. It is worth mentioning that, regarding the Objective I, the parameters a and b were empirically adjusted aiming, primarily, the overall stability of the system. However, the settling time and the overshoot of the system are also evaluated in these simulations.

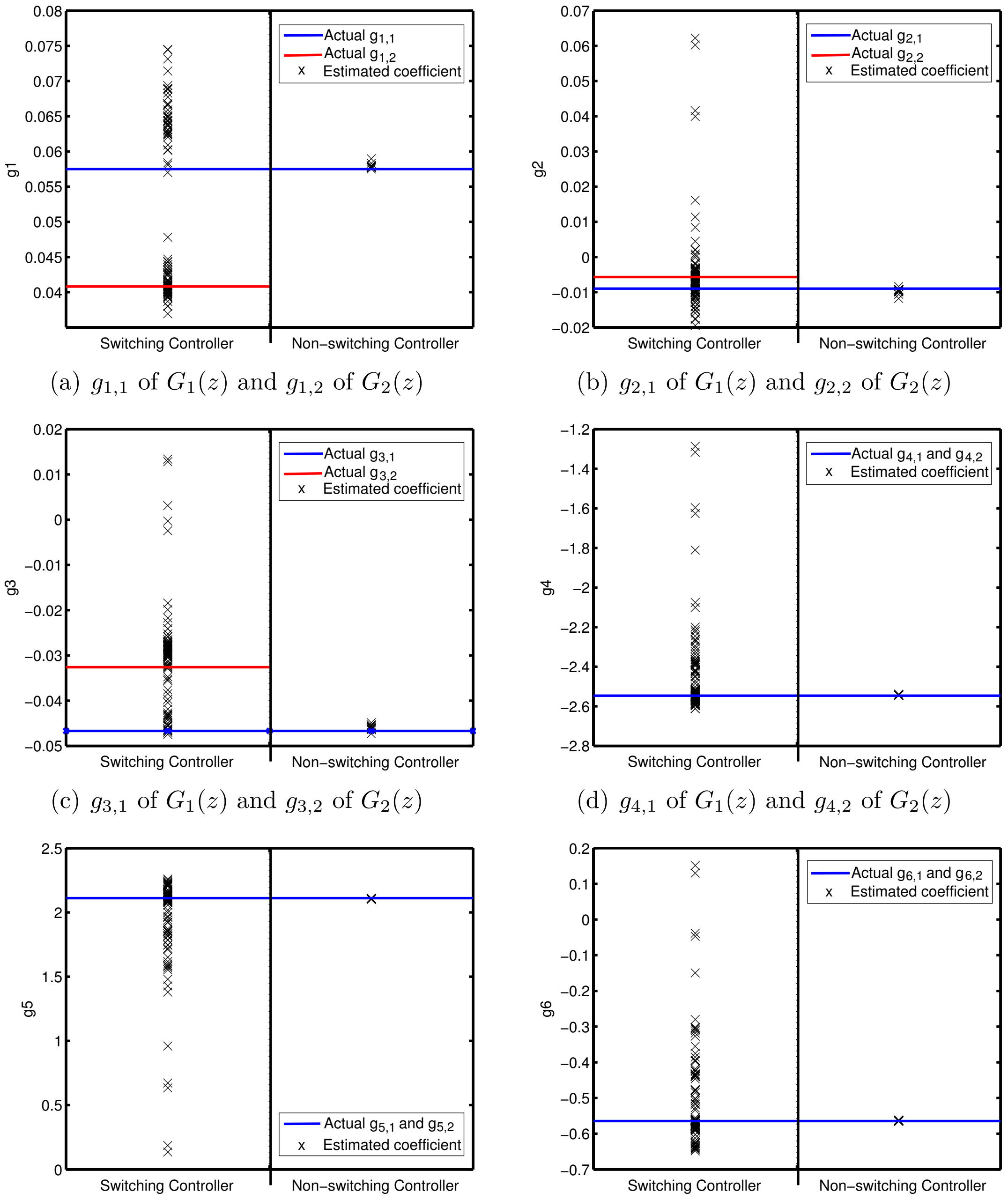
The attack is implemented using the BSA, given that this metaheuristic presented the best performance in the attack simulations shown in Section 5.4. Also, the parameters of the BSA are the same as in Section 5.4: the population is set to 100 individuals; the limits of each dimension of the search space are $[-10{,}10]$; and η – that establishes the amplitude of the movements of the individuals of the BSA – is set to 1. In each simulation, the BSA is executed for 4500 iterations.

As in Section 5.4, the attack signal $a(k)$ shown in Figure 4 is a discrete-time unitary impulse defined by (5.10). In each simulation, the feedback stream is captured by the attacker during a period $T = 2s$ (100 samples), starting at sample $k_a + 1$. In both NCSs, the sample rate is 50 samples/s and the set point $r(k)$ is an unitary step function. Network delay and packet loss are not taken into account in these simulations.

6.1.2 Performance as a Countermeasure

This section presents the results obtained by the Active System Identification attack when launched in the NCSs described in Section 6.1.1 – one NCS using the switching controller and the other using the non-switching controller. In each NCS, there were executed 100 attack simulations. All coefficients estimated by these 100 attack simulations in each NCS are presented in Figure 30. Recall that the NCS with the non-switching controller has only one open-loop transfer function $G_1(z)$, while the NCS with the switching controller has two open-loop transfer functions $G_1(z)$ and $G_2(z)$. Note that the actual values of the coefficients $[g_{1,1}, g_{2,1}, g_{3,1}, g_{4,1}, g_{5,1}, g_{6,1}]$ and $[g_{1,2}, g_{2,2}, g_{3,2}, g_{4,2}, g_{5,2}, g_{6,2}]$ of the two open-loop transfer functions $G_1(z)$ and $G_2(z)$, respectively, are also depicted in Figure 30. By observing Figures 30(a) to 30(f), it is possible to state that the coefficients estimated in the NCS with the non-switching controller are precise and accurate. In this NCS, with non-switching controller, the Active System Identification attack provides the information and the confidence that the attacker needs to design other covert/model-based attacks. On the other hand, in the NCS endowed with the proposed countermeasure, the use of the switching controller causes the dispersion of the estimated values, reducing the precision and the accuracy of the coefficients obtained by the attacker. As shown in Figure 30, the set of estimated values in this SLS are spread and does not accurately indicate any of the coefficients of $G_1(z)$ and $G_2(z)$.

The impact of the use of the switching controller in the attack performance can also be verified by comparing the global minimum values found for the fitness function (3.9). In the NCS endowed with the switching controller, the global minimum values of all attack simulations are within 1.81×10^{-06} and 1.96×10^{-04} (the mean is 2.50×10^{-05}, and the standard deviation is 3.97×10^{-05}). On the other hand, in the NCS with the non-switching controller, all global minimum values are within 7.82×10^{-09}



(a) $g_{1,1}$ of $G_1(z)$ and $g_{1,2}$ of $G_2(z)$

(b) $g_{2,1}$ of $G_1(z)$ and $g_{2,2}$ of $G_2(z)$

(c) $g_{3,1}$ of $G_1(z)$ and $g_{3,2}$ of $G_2(z)$

(d) $g_{4,1}$ of $G_1(z)$ and $g_{4,2}$ of $G_2(z)$

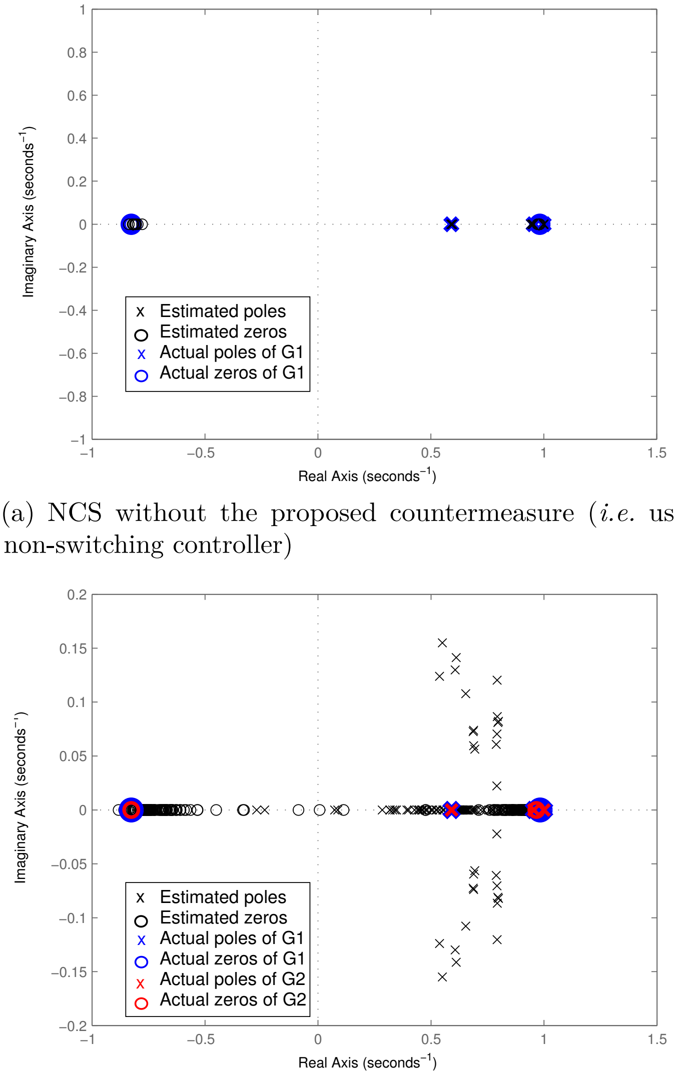
(e) $g_{5,1}$ of $G_1(z)$ and $g_{5,2}$ of $G_2(z)$

(f) $g_{6,1}$ of $G_1(z)$ and $g_{6,2}$ of $G_2(z)$

Figure 30 – Coefficients estimated by the Active System Identification Attack in NCSs using the proposed countermeasure (with a switching controller) and without the proposed countermeasure (using a non-switching controller) – own figure published in (SA; CARMO; MACHADO, 2017d).

and 4.46×10^{-08} (the mean is 8.75×10^{-09}, and the standard deviation is 4.80×10^{-09}). Recall that, as discussed in Section 3.1.2, without perturbation or noise, the minimum value of (3.9) is $\min f_j = 0$ when the attacked system is perfectly identified. So, the higher order of the global minimum values caused by the use of the switching controller also demonstrates the effectiveness of the proposed countermeasure. From the attacker point o view, these higher global minimum values may be an indicative that the Active

System Identification attack was not effective in obtaining the model of the attacked system. In this sense, the attacker must hesitate to launch covert/model-based attacks based on the information gathered by the Active System Identification attack.



(a) NCS without the proposed countermeasure (*i.e.* using a non-switching controller)

(b) NCS using the proposed countermeasure (*i.e.* with a switching controller)

Figure 31 – Zeros and poles estimated by the Active System Identification attack – own figure published in (SA; CARMO; MACHADO, 2017d).

The impact of the proposed countermeasure in the referred Active System Identification attack can also be verified in the pole-zero maps shown in Figure 31. Figure 31(a) shows the zeros and poles of the open-loop transfer functions estimated by the 100 simulations with the non-switching controller. Figure 31(b), in turn, shows the zeros and poles of the open-loop transfer functions estimated by the simulations using the switching controller. Note that, in the simulations with the non-switching controller, the estimated zeros and poles accurately meet the actual zeros and poles of the open-loop transfer function $G_1(z)$ of the NCS. On the other hand, Figure 31(b) shows that when the proposed countermeasure is used, the estimated zeros and poles are spread and do not concur for the actual zeros and poles of $G_1(z)$ and $G_2(z)$ – *i.e.* the open-loop transfer functions of the two subsystems of the SLS.

The spreading of the estimated poles and zeros in Figure 31(b), the inaccuracy of the estimated coefficients shown in Figure 30, and the higher global minimum values found by the BSA demonstrate the effectiveness of using switching controllers as a countermeasure for the Active System Identification attack described in Section 3.1.2. With the proposed countermeasure, it is possible to state that the model obtained by the attacker is imprecise/ambiguous in such a way that, with the obtained information, the attacker may hesitate in launching other covert/model-based attacks. So, Objective II defined in Section 4.1.2 is met.

6.1.3 Complying the Control Requirements

In this section, the performance of the proposed countermeasure is analyzed from the control perspective, in order to identify possible impacts that it may produce in the control of the plant. To do so, the following aspects are evaluated: stability; overshoot; and settling time. Considering these aspects, the performance of the switching controller is compared with the performance of the non-switching controller. Given the stochastic nature of the switching controller described in Section 6.1.1, which randomly switches among two control functions, the mentioned aspects are evaluated through a set of 100,000 simulations.

Figure 32 shows the responses of both NCSs in the time domain. The responses of the NCS endowed with the proposed countermeasure is represented by the highlighted area. The bounds of this area are drawn based on the maximum and minimum values of the output of the plant, considering all 100,000 simulations. In another words, when using the switching controller, all output signals provided by the simulations are within this highlighted area. The non-stochastic response of the NCS using the non-switching controller is represented in Figure 32 by the red line. Note that, up to $t = 0.4s$ the responses using the switching controller are the same as the response with the non-switching controller. This is caused by the minimum dwell time of $0.4s$, set by the minimum number of sampling intervals that the system have to remain in the same state, defined in Section 6.1.1 as $a = 20$ samples. Based on Figure 32, it is possible to verify that, considering all 100,000 simulations, the NCS with the proposed countermeasure is stable, the output of the plant converges to the set point ($1rad/s$) without stationary error, and it does not present overshoots. In these aspects, from the control perspective, the proposed countermeasure presents the same performance as the non-switching controller.

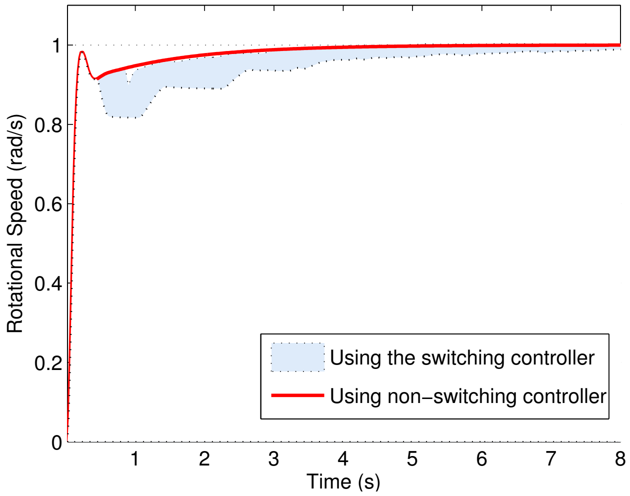


Figure 32 – Response of the systems in the time domain – own figure published in (SA; CARMO; MACHADO, 2017d).

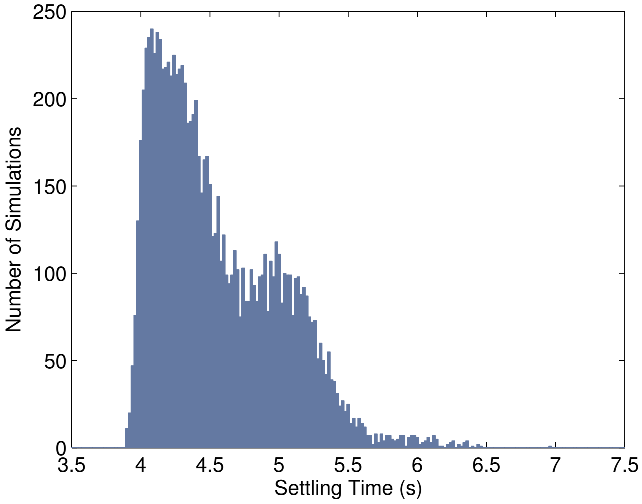


Figure 33 – Histogram of the settling time when using the proposed countermeasure – own figure published in (SA; CARMO; MACHADO, 2017d).

On the other hand, due to the successive switchings, it is possible to verify in Figure 32 that the settling time of the proposed countermeasure is higher than the settling time provided by the non-switching controller. The deterministic settling time of the NCS with the non-switching controller is $2.4s$. The settling time t_s provided by the switching controller is stochastic and depends on the sequence of dwell times occurred before achieving t_s, which is random. The settling times of all 100,000 simulations using the switching controller are represented in the histogram shown in Figure 33. The minimum and maximum settling times are $3.90s$ and $6.96s$, respectively, and the mean is $4.555 \pm 0.0088s$, with a confidence interval of 95%.

The performance of the proposed countermeasure, from the control perspective, is satisfactory and indicates the feasibility of meeting Objective I and Objective II, simultaneously. In these simulations, the control provided by the switching controller presents a performance similar to the performance of the non-switching controller. The primary requirement of Objective I – *i.e.* stability – is met, as well as the requirement of not causing overshoots on the plant. However, the simulations indicate an increase in the settling time of the system, which may not be an issue, but have to be analyzed depending on the specific process being controlled. In this sense, the tradeoff between hindering the identification attack and increasing the settling time must be taken into account when deciding for using this countermeasure.

6.2 MITIGATION OF THE PASSIVE SYSTEM IDENTIFICATION ATTACK

This section evaluates the performance of the switching controller when the Passive System Identification attack described in Section 3.1.1 is launched in an NCS. As in Section 6.1, two NCSs are used for comparison: one with the proposed countermeasure – *i.e.* using a switching controller; and another without the proposed countermeasure – *i.e.* using a non-switching controller. The specifications of these NCSs and the parameters of the attack are described in Section 6.2.1 Recall that, according to Section 4.1.2, the design of the switching controller must follow two objectives: hinder the identification process; and comply with the plant's control requirements. The results concerning these two objectives are presented in Sections 6.2.2 and 6.2.3, respectively, in order to demonstrate the feasibility of the solution from both perspectives. Additionally, Section 6.2.4 demonstrates the impact caused in the SD-Controlled Data Injection attack, described in Section 3.2.1, when the Passive System Identification Attack is mitigated by the proposed countermeasure.

6.2.1 Attacked NCSs and Parameters of the Attack

In Sections 6.2.2 and 6.2.3, the results obtained with the proposed countermeasure are compared with the results obtained in an NCS without the proposed countermeasure – *i.e.* endowed with a non-switching controller. As in Section 5.1, the NCS with the non-switching controller consists of a Proportional-Integral (PI) controller that controls the rotational speed of a DC motor. The PI control function $C_1(z)$ and the DC motor transfer function $G(z)$ are represented by (6.6) and (6.7), respectively:

$$C_1(z) = \frac{c_{1,1}z + c_{2,1}}{z - 1} \tag{6.6}$$

$$G(z) = \frac{g_1 z + g_2}{z^2 + g_3 z + g_4} \tag{6.7}$$

wherein $c_{1,1} = 0{,}1701$, $c_{2,1} = -0{,}1673$, $g_1 = 0{,}3379$, $g_2 = 0{,}2793$, $g_3 = -1{,}5462$ and $g_4 = 0{,}5646$. In both NCSs, the sample rate is 50 samples/s and the set point $r(k)$ is a unitary step function.

The NCS with the proposed countermeasure has the same architecture shown in Figure 7 and controls a DC motor whose transfer function is also defined by (6.7) – *i.e.* it controls the same plant that is controlled by the NCS with the non-switching controller. The switching controller has two control functions: $C_1(z)$, that is the same control function (6.6) of the non-switching controller; and $C_2(z)$ defined by (6.8),

$$C_2(z) = \frac{c_{1,2}z + c_{2,2}}{z - 1}. \tag{6.8}$$

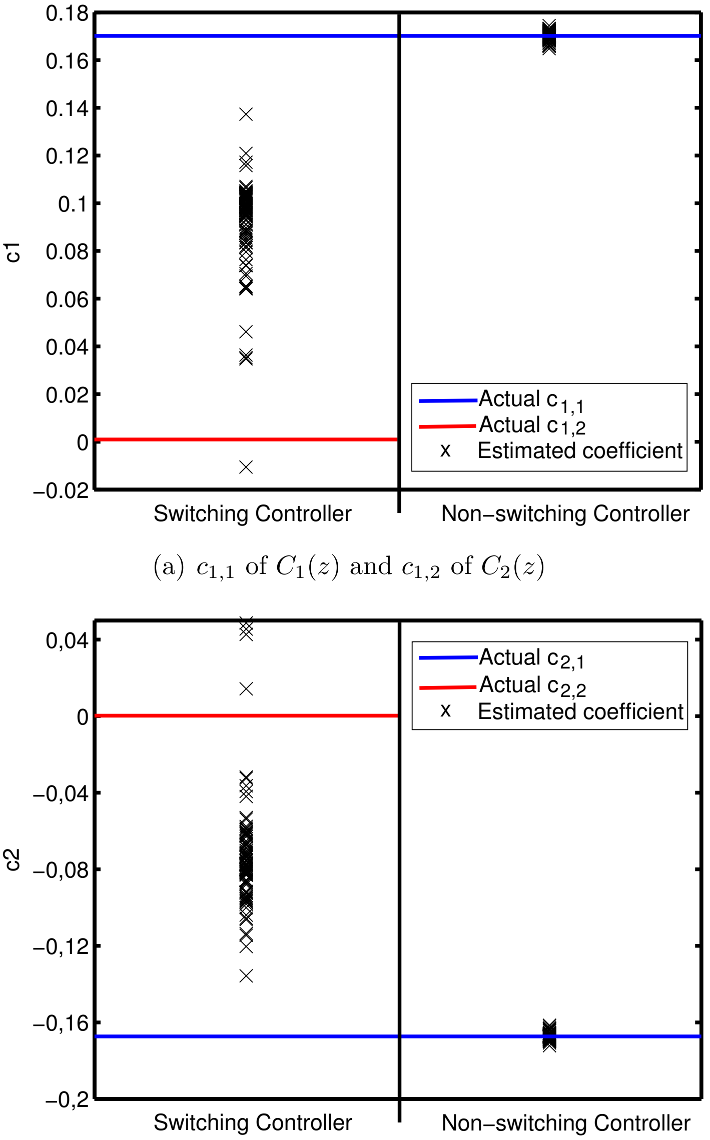
wherein $c_{1,2} = 0.001$ and $c_{2,2} = 0.0002$. So, the NCS with the switching controller is an SLS with two subsystems. The control functions $C_1(z)$ and $C_2(z)$ are designed to make each subsystem stable – when separately analyzed – and are randomly switched based on the switching rule defined by the Markov chain and the PDF shown in Figures 8 and 9, respectively. The parameters a and b of the PDF were empirically adjusted to $a = 40$ and $b = 60$, in order to meet Objectives I and II defined in Section 4.1.2. Regarding Objective I, it is worth mentioning that a and b were empirically adjusted aiming, primarily, the global stability of the SLS. However, the settling time and the overshoot of the plant are also evaluated in Section 6.2.3.

Regarding the Passive System Identification attack, the parameters of the BSA are the same as those defined in Section 5.1. Also, the forward and feedback streams are captured by the attacker during a period $T = 2s$ (100 samples).

6.2.2 Performance as a Countermeasure

This section presents the results obtained by the Passive System Identification attack, when attacking both switching and non-switching controllers. For each controller, 100 attack simulations were performed. To evaluate the proposed countermeasure, we considered the scenario where the attacker obtained the best performance in Section 5.1 – *i.e.* without packet loss.

The coefficients estimated by all attack simulations (100 for each controller) are presented in Figure 34. Recall that the non-switching controller just have one control function $C_1(z)$, while the switching controller has two control functions $C_1(z)$ and $C_2(z)$. Note that the actual values of the coefficients $[c_{1,1},c_{2,1}]$ and $[c_{1,2},c_{2,2}]$ of the two control functions $C_1(z)$ and $C_2(z)$, respectively, are also depicted in Figure 34. Analyzing Figures 34(a) and 34(b), it is possible to verify that the estimated coefficients of the non-switching controller are precise and accurate. In this case, the estimated coefficients



(a) $c_{1,1}$ of $C_1(z)$ and $c_{1,2}$ of $C_2(z)$

(b) $c_{2,1}$ of $C_1(z)$ and $c_{2,2}$ of $C_2(z)$

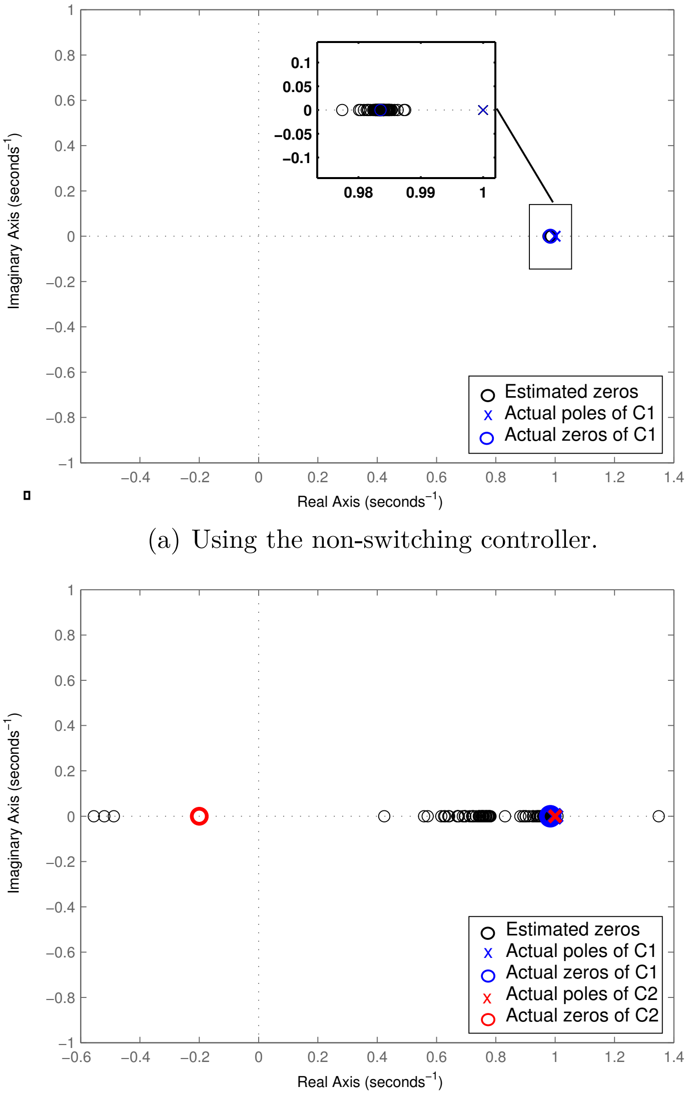
Figure 34 – Coefficients estimated by the Passive System Identification Attack – own figure published in (SA; CARMO; MACHADO, 2018).

are concentrated close to the actual values of $c_{1,1}$ and $c_{2,1}$. This concentration indicates that, with the non-switching controller, the Passive System Identification attack provides the information and the confidence that the attacker needs to design a covert/model-based attack – such as the SD-Controlled Data Injection attack described in Section 3.2.1. On the other hand, Figure 34 shows that the use of the switching controller causes the dispersion of the estimated coefficients, reducing the precision and the accuracy of the Passive System Identification attack. With the switchings, the set of estimated values are spread and does not accurately indicate any of the coefficients of $C_1(z)$ and $C_2(z)$. It is worth mentioning that this spreading has a dissuasive effect. It increases the uncertainty of the attacker regarding the model of the attacked controller, in such way that the attacker may hesitate to proceed with his intention of a covert/model-based attack that relies on an accurate knowledge about the controller.

The impact of the switching controller in the Passive System Identification attack can also be verified through the analysis of the global minimum values obtained for the fitness function (3.3). With the switching controller, the global minimum values of all attack simulations are between 2.64×10^{-04} and 8.53×10^{-04} (the mean is 7.42×10^{-04}, and the standard deviation is 1.70×10^{-04}). On the other hand, with the non-switching controller, all global minimum values are between 1.70×10^{-09} and 1.44×10^{-06} (the mean is 1.84×10^{-07}, and the standard deviation is 2.70×10^{-07}). Recall that, as discussed in Section 3.1.1, without sample loss, the minimum value of (3.9) is $\min f_j = 0$ when the attacked device is perfectly identified. Therefore, the higher order of the global minimum values obtained with the switching controller also demonstrates the effectiveness of the proposed countermeasure. From the attacker perspective, these higher global minimum values may indicate that the Passive System Identification attack was not effective in obtaining the model of the attacked device. In this sense, with this analysis, the attacker must hesitate to launch covert/model-based attacks based on the data provided by the Passive System Identification attack.

Another way to evaluate the impact of the proposed countermeasure in the Passive System Identification attack is through the zero-pole maps shown in Figure 35. Figure 35(a) shows the zeros estimated by the simulations using the non-switching controller. Figure 35(b), in turn, shows the zeros estimated by the simulations using the switching controller. Note that, when the non-switching controller is attacked, the estimated zeros accurately meet the actual zero of $C_1(z)$. On the other hand, according to Figure 35(b), when the proposed countermeasure is used the estimated zeros are spread and do not accurately meet the actual zeros of $C_1(z)$ and $C_2(z)$ – *i.e.* the control functions of the switching controller.

It must be considered the possibility that the attacker, after some time, detects that the controller is changing its behavior over the time like a switching controller. In this case, it is reasonable to think that the attacker would try to estimate the control functions based on smaller monitoring periods T, to avoid measurements containing switching events. Considering this hypothesis, the performance of the Passive System Identification attack is also evaluated using the following monitoring periods T: $0.2s$, $0.4s$, $0.6s$, $0.8s$, $1.0s$ and $1.2s$. Note that the maximum T in which the attacker can measure a signal without switchings is $T_b = 0.02b = 1.2s$. Therefore, to evaluate this tactic (of reducing T), the Passive System Identification attack is performed firstly during the



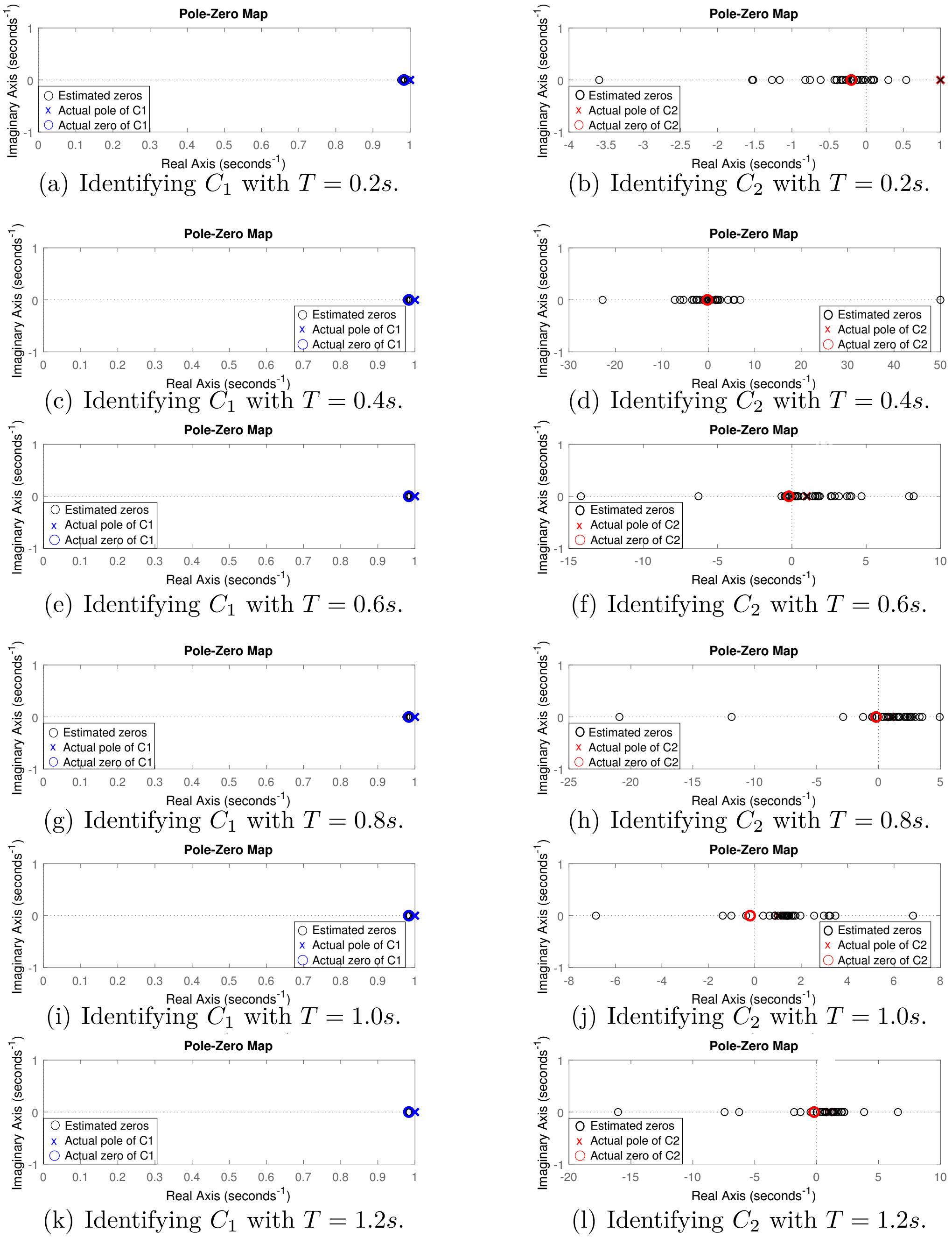
(a) Using the non-switching controller.

(b) Using the switching controller.

Figure 35 – Zeros and poles estimated by the Passive System Identification attack – own figure published in (SA; CARMO; MACHADO, 2018).

execution of $C_1(z)$ and, after that, during the execution of $C_2(z)$. For the identification of $C_1(z)$ all monitoring periods start at $t = 0s$. For the identification of $C_2(z)$ all monitoring periods start at the first switching event (when $C_2(z)$ starts to be executed).

For each control function and each monitoring period, 33 attack simulations were executed. Figure 36 shows the estimated zeros of $C_1(z)$ and $C_2(z)$ considering each of the mentioned monitoring periods T. It is possible to verify that, for these monitoring periods, the estimated zeros of $C_1(z)$ are quite close to the actual zero. However, although $C_1(z)$ was satisfactorily identified with small T, Figure 36 shows that, for all T, the estimated zeros of $C_2(z)$ are spread and do not accurately meet the actual zero of $C_2(z)$. These results indicate that small monitoring periods T may not be enough to identify some control functions, such as happened with $C_2(z)$. In this case, the switching controller



(a) Identifying C_1 with $T = 0.2s$.

(b) Identifying C_2 with $T = 0.2s$.

(c) Identifying C_1 with $T = 0.4s$.

(d) Identifying C_2 with $T = 0.4s$.

(e) Identifying C_1 with $T = 0.6s$.

(f) Identifying C_2 with $T = 0.6s$.

(g) Identifying C_1 with $T = 0.8s$.

(h) Identifying C_2 with $T = 0.8s$.

(i) Identifying C_1 with $T = 1.0s$.

(j) Identifying C_2 with $T = 1.0s$.

(k) Identifying C_1 with $T = 1.2s$.

(l) Identifying C_2 with $T = 1.2s$.

Figure 36 – Zeros and poles estimated by the Passive System Identification attack for smaller monitoring periods T (without a switching event during T). Identification of C_1 starting at $t = 0$. Identification of C_2 starting at the first switching event – own figure published in (SA; CARMO; MACHADO, 2018).

arises as a good strategy to limit the available monitoring period, which causes difficulties for this metaheuristic-based Passive System Identification attack. Additionally, it is worth mentioning that even if the attacker somehow identifies all control functions $C_i(z)$, the random switching rule still mitigates the launch of a subsequent covert/model-based attack. As discussed in Section 4.1.2, this follows from the fact that it is more difficult to synchronize the interference caused by a covert/model-based attack with the controller states, which are switched at random intervals. Moreover, it is not trivial to find a single $M(z)$ capable to produce the intended controlled behavior for all $C_i(z)$ – in case the attacker choose this tactic to overcome the need to synchronize the covert/model-based attack.

The inaccuracy of the estimated coefficients shown in Figure 34, the spreading of the estimated zeros shown in Figure 35(b), and the higher global minimum values found by the BSA demonstrate the effectiveness of using switching control functions to mitigate the Passive System Identification attack described in Section 3.1.1. With this countermeasure, it is possible to state that the model obtained by the attacker is imprecise/ambiguous such that the attacker may hesitate to launch a subsequent covert/model-based attack that depends on the knowledge about the controller. Therefore, Objective II established in Section 4.1.2 is met.

6.2.3 Complying the Control Requirements

This section analyzes the performance of the proposed countermeasure from the control perspective. The aim of the simulations herein presented is to identify the possible impacts that the countermeasure may produce in the behavior of the plant. As in Section 6.1.3, this analysis encompasses the following control aspects: stability; overshoot; and settling time. Based on these aspects, the present section compares the performance of the switching controller with the performance of the non-switching controller. Considering the stochastic nature of the proposed countermeasure, which randomly switches between two control functions, the referred aspects are evaluated through a set of 100,000 simulations.

Figure 37 shows the responses of the plant, in the time domain, with and without the proposed countermeasure. The responses obtained with the proposed countermeasure – *i.e.* using the switching controller – are represented by the highlighted area. The bounds of this area are drawn based on the maximum and minimum values of the output $y(t)$ of the plant, taking into account all 100,000 simulations. In other words, when using the proposed countermeasure, all output signals $y(t)$ provided by the simulations are inside this area. The deterministic response of the plant without this countermeasure – *i.e.* when using the non-switching controller – is represented by the red line depicted in Figure 37. Note that, for $0 \leq t \leq 0.8s$ the responses using the switching controller and the response using the non-switching controller are identical. This is caused by the minimum number of sampling intervals that the system has to remain in the same state, which is set to $a = 40$ samples (or $0.8s$, in the time domain).

Based on Figure 37, considering all 100,000 simulations, it is possible to verify that the NCS with the proposed countermeasure is stable and the output of the plant does not present a stationary error – it always converges to the set point of $1rad/s$. Considering these aspects, from the control perspective, the proposed countermeasure provides the same performance as the non-switching controller. Also, the highlighted area indicates that the overshoots obtained with the countermeasure are not expressive, not exceeding 2.93% of the set point.

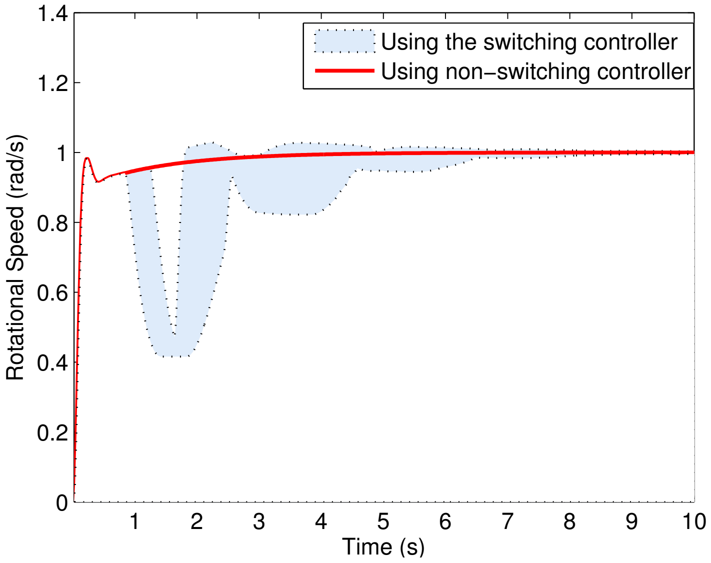


Figure 37 – Response of the plant in the time domain – own figure published in (SA; CARMO; MACHADO, 2018).

However, due to the successive switchings, it is possible to see in Figure 37 that the settling time obtained with the proposed countermeasure is higher than the settling time obtained with the non-switching controller. With the non-switching controller, the deterministic settling time of the plant is $2.4s$. On the other hand, with the switching controller, the settling time t_s of the plant is stochastic and depends on the random sequence of dwell times occurred before achieving t_s. Figure 38 shows a histogram of settling times that considers all 100,000 simulations using the switching controller. The minimum and maximum settling times are $2.88s$ and $6.42s$, respectively, and the mean is $4.2827s \pm 0.0146s$, with a confidence interval of 95%. It indicates that, regarding the settling time, the proposed countermeasure is less efficient than the non-switching controller.

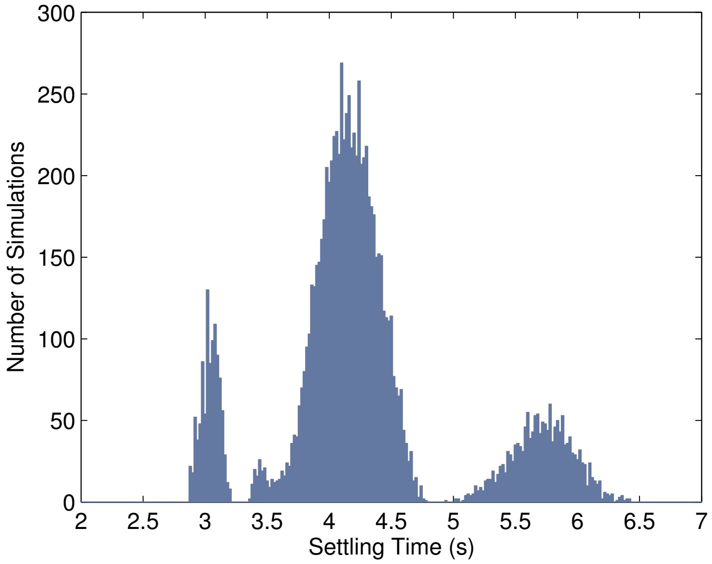


Figure 38 – Histogram of settling times when using the proposed countermeasure – own figure published in (SA; CARMO; MACHADO, 2018).

It is worth mentioning that Figure 37 exemplifies the behavior of the proposed countermeasure and compare its performance with the performance of an NCS with a non-switching controller. From this figure, it is possible to observe a behavioral profile that allows the evaluation of characteristics such as overshoot, settling time and stability. Regarding the latter, the stability of systems based on the average dwell time technique can be verified by the theory proposed in (ZHAI et al., 2002), which demonstrates the feasibility of the proposed countermeasure in terms of stability.

Note in Figure 37 that the random switching rule adds to the system a variable (however, controlled and stable) behavior, which could reduce the ability of a human observer to detect slight manipulations caused by a physically covert attack. However, it is noteworthy that when an attacker designs a physically covert attack, as a premise, he/she does not aim to explore or manipulate physical behaviors that are easy to be noticed by a human observer. Instead of this, the attacker would manipulate physical behaviors that are not accurately perceived by a human observer. In this case, it is reasonable to consider that the variations caused by the switching controller will not significantly contribute for the poor perception of malicious and covert interferences that would naturally not be perceived by a human observer (even when a non-switching controller is used).

From the control perspective, the performance of the proposed countermeasure is satisfactory and, with the results presented in Section 6.2.2, indicates the feasibility of meeting both Objectives I and II, simultaneously. According to the simulations of this section, the control provided by the switching controller presents a performance similar to the performance of the non-switching controller. The primary requirement of Objective I – *i.e.* stability – is met and the overshoots caused by the countermeasure, with the specified configurations, are not expressive. However, the simulations indicate an increase in the plant settling time, which may not be a drawback, but have to be analyzed in the face of the specific process being controlled. In this sense, the results denote the existence of a tradeoff between hindering the identification attack and increasing the settling time of the system, which must be taken into account when deciding for using this countermeasure.

6.2.4 Impact in the Controlled Data Injection Attack

Consider that the attacker was not dissuaded by the uncertainties caused by the proposed countermeasure in the identification of the controller. Doing so, the aim of this section is to evaluate the impact of the proposed countermeasure in the design of an SD-Controlled Data injection attack.

The SD-Controlled Data Injection attack simulated in this section aims to cause an overshoot of 50% in the rotational speed of the DC motor defined by (6.7), such as the attack evaluated in Section 5.1. According to Section 3.2.1, to perform an SD-Controlled Data Injection attack, the attack function $M(z)$ must be designed based on the models of the plant and its controller.

If an attacker, aiming to cause an overshoot of 50% in $y(k)$ (for instance), implements an attack function $M(z)$ in the forward stream of an NCS, as shown in Figure 5, then $y(k)$ is defined by (6.9):

$$y(k) = \mathcal{Z}^{-1}\left[\frac{C(z)M(z)G(z)}{1 + C(z)M(z)G(z)}R(z)\right]. \tag{6.9}$$

Similarly, if the attacker implements $M(z)$ in the feedback stream, then $y(k)$ is defined by (6.10):

$$y(k) = \mathcal{Z}^{-1}\left[\frac{C(z)G(z)}{1 + C(z)M(z)G(z)}R(z)\right]. \tag{6.10}$$

Note that in both cases, in the presence of an attack function $M(z)$, the dynamics of $y(k)$ rely on $C(z)$, $G(z)$ and $M(z)$. Therefore, if the attacker aims to cause an overshoot of 50% in $y(k)$, the design of $M(z)$ will require the knowledge of $C(z)$ and $G(z)$. Even if the attacker is still able to identify the plant model (which is not mitigated by this countermeasure), he/she will not be able to design $M(z)$ to cause the 50% overshoot based only on the model of the plant, regardless of whether $M(z)$ is implemented in the forward or the feedback stream.

The evaluation made in this section considers that $M(z)$ is implemented in the forward stream of the NCS. The identification of the plant's transfer function $G(z)$ is not impacted by the use of the switching controller, as discussed in Section 4.1.2. So, the same $G(z)$ estimated in Section 5.1 (with a non-switching controller) is used in this section to design $M(z)$. Specifically, the coefficients used for $G(z)$ are the mean estimated coefficients shown in Table 2 for 0% of sample loss (which is the most accurate estimated model of $G(z)$). Regarding the model of the controller, as done in Section 5.1, $M(z)$ is designed considering the mean of the coefficients estimated by the Passive System Identification attack when launched against the switching controller. Then, performing a root locus analysis, the attacker designs the attack function (6.11), to make the system underdamped with a peak of rotational speed 50% higher than its steady state speed.

$$M(z) = 1.2815 \tag{6.11}$$

In Figure 39, it is possible to compare the response that the attacker expects to obtain (referred as *expected response*) with the responses that (6.11) actually produces (referred as *actual responses*) when implemented in the real system. The *expected response* represents what the attacker would obtain by simulating (6.11) in the forward stream of an NCS built with the models provided by the Passive System Identification attack. The *actual responses* are represented by the highlighted area, whose bounds are drawn based on the maximum and minimum values of the output $y(t)$ of the plant, considering 100,000 simulations with (6.11) in the forward stream of the actual NCS. It means that, when (6.11) is implemented in the NCS all output signals $y(t)$ provided by the actual plant are inside this area.

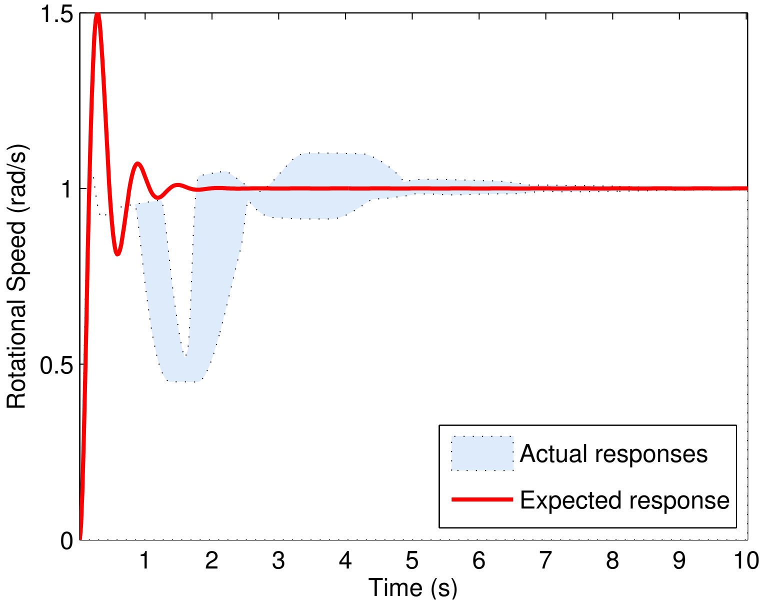


Figure 39 – Results of an SD-Controlled Data Injection attack in a system with the proposed countermeasure.

It is worth mentioning that the aim of Figure 39 is not to evaluate the stability of the proposed system after the execution of the SD-Controlled Data Injection attack (although in these simulations this system remained stable even after the execution of $M(z)$). The aim of Figure 39 is to demonstrate that, with the proposed countermeasure, the interference produced by the attacker is not what he/she intended with the mentioned Data Injection attack. Note that, the actual responses of the plant are significantly different from the response that the attacker expects to obtain with the SD-Controlled Data Injection attack. These results are in contrast to the results achieved in the NCS with the non-switching controller, where the attack was accurate and executed exactly what was planned by the attacker, as shown in Section 5.1. With the proposed countermeasure, the maximum overshoot achieved by the plant was 10.12% (instead of the desired 50%). Notwithstanding, the highlight of these simulations is the fact that, with the proposed countermeasure, the information provided by the Passive System Identification attack is not useful to support the design covert/model-based attacks. This inaccurate information may lead the attacker to cause unpredictable results in the system, which may either be ineffective (not causing the desired degradation on the plant) or extreme (reducing the physical or cybernetic covertness of the attack). This analysis is consistent with the reasoning provided in Section 6.2.2. It demonstrates that when the NCS is endowed with the proposed countermeasure, the attacker must hesitate to launch a covert/model-based attack due to the inaccuracy of the Passive System Identification attack.

Note that the countermeasure proposed in this work aims to mitigate the Passive System Identifications attacks when the attacker is trying to obtain information about the control functions of the NCS. Consequently, it prevents the use of accurate information about these control functions in the design of a covert/model-based attack (such as a data injection attack in the forward stream of an NCS aiming to cause an overshoot or a steady state error). For instance, in an SD-Controlled Data Injection attack performed in the forward stream of the NCS, the attacker cannot cause a steady state error by just adding a step signal to $u(k)$, because the PI control functions will adjust the control signal to bring $y(k)$ back to $1rad/s$. Adding a ramp signal to $u(k)$ can cause a steady error in $y(k)$ for a while. However, it may not be a good strategy for the attacker, because at some time the controller and $u(k)$ will saturate, leading the plant to extreme behaviors (which is not desired if the attacker aims a physically covert attack). The alternative to cause a steady state error through the manipulation of the forward stream is to implement the attack function $M(z)$ exemplified in Section 5.1 which, to be designed, requires the knowledge about the controller and plant. Without the knowledge about the coefficients of the numerator of the PI control function, for example, the gain of $M(z)$ cannot be adjusted to cause the exact steady deviation of $y(k)$ that the attacker intends to cause. This makes the attack described in Section 5.1 model-based and, in this case, the countermeasure herein proposed is useful to hinder the attacker from obtaining the knowledge about the control functions of the NCS. On the other hand, in a system with an unitary feedback, it is possible to manipulate the steady state error of the plant by injecting data in the feedback stream, even when the attacker does not know the models of the plant and the controller. In this case, the manipulation of $y(k)$ can be interpreted as the direct manipulation of set point $r(k)$, which determines the steady state of the system. This attack, performed in the feedback stream is an example of data injection attack that is not model-based and, thus, should be mitigated by an additional countermeasure (complementary to the countermeasure proposed in this work).

6.3 IDENTIFICATION OF CONTROLLED DATA INJECTION ATTACKS

This section analyses the performance of the attack identification strategy proposed in section 4.2 when identifying the Controlled Data Injection attack characterized in Section 3.2.1. The evaluation on the accuracy of the countermeasure is based on results obtained through simulations using MATLAB/SIMULINK. First, Section 6.3.1 describes the attacked NCS and the attack parameters. Then, Section 6.3.2 presents the results obtained by the proposed countermeasure in the scenario described in Section 6.3.1.

6.3.1 Attacked NCSs and Parameters of the Attack

In the simulations of this section, the attacked NCS has the same architecture of the NCS shown in Figure 10. The system consists of Proportional-Integral (PI) controller that controls the rotational speed of a DC motor. The control function $C(z)$ and the plant transfer function $P(z)$ are the same as in Section 5.1, which are represented by (6.12):

$$C(z) = \frac{0.1701z - 0.1673}{z - 1} \quad P(z) = \frac{0.3379z + 0.2793}{z^2 - 1.5462z + 0.5646} \tag{6.12}$$

The sample rate of the system is 50 samples/s and the set point $r(k)$ is a unitary step function.

As discussed in Section 3.2.1, one way to degrade the service of a plant is by causing overshoots during its transient response. Thus, an attack function $M(z)$ is designed to degrade the plant service by causing 50% of overshoot in the motor speed. To achieve this goal, a MitM located in the feedback link runs the attack function represented by (6.13), wherein $\alpha_0 = 0.25$ and $\beta_0 = -0.75$:

$$M(z) = \frac{\alpha_0}{z + \beta_0}. \tag{6.13}$$

6.3.2 Performance of the Attack Identification

Section 4.2 proposes an attack identification process where the NII technique is used to improve the accuracy of the estimation of LTI attack functions in NCSs. This section analyzes the performance of the proposed attack identification method when estimating the attack defined in Section 6.3.1. To statistically evaluate how the NII technique improves the accuracy of the identification process, two set of simulations are carried out:

1. 100 simulations using the identification process shown in Algorithm 1 – *i.e.* without the NII technique; and
2. 100 simulations using the identification process shown in Algorithm 3 – *i.e.* with the NII technique.

The noise $w(k) \sim N(\mu,\sigma)$ injected in the system by the identification scheme is configured with $\mu = 0$ and $\sigma = 0.005$, which makes 95% of the noise amplitudes within ±0.01 (these parameters are chosen to produce a small noise, considering the magnitude of the plant output signal transmitted through the feedback link). Each of the 100 simulations with Algorithms 1 and 3 uses a different (randomly generated) white gaussian noise signal.

Figure 40 shows examples of the system output (the motor speed) with and without the attack. Note that, when the attack is executed, the motor speed has an overshoot of 50% and a small noise is present in the plant output. However, in a normal condition – *i.e.*, without attack – the noise is cancelled and does not appear in the plant output (as expected, based on equation (4.1) when $M(z) = 1$).

As previously discussed, the attack identification scheme aims to estimate the coefficients of $M(z)$, which according to (6.13) are α_0 and β_0. The BSA settings in both Algorithms 1 and 3 are the same as those used in Section 5.1.2: the lower and upper limits of each search space dimension are -10 and 10, respectively; the BSA population has 100 individuals; and $\eta = 1$. The BSA is executed for 600 iterations.

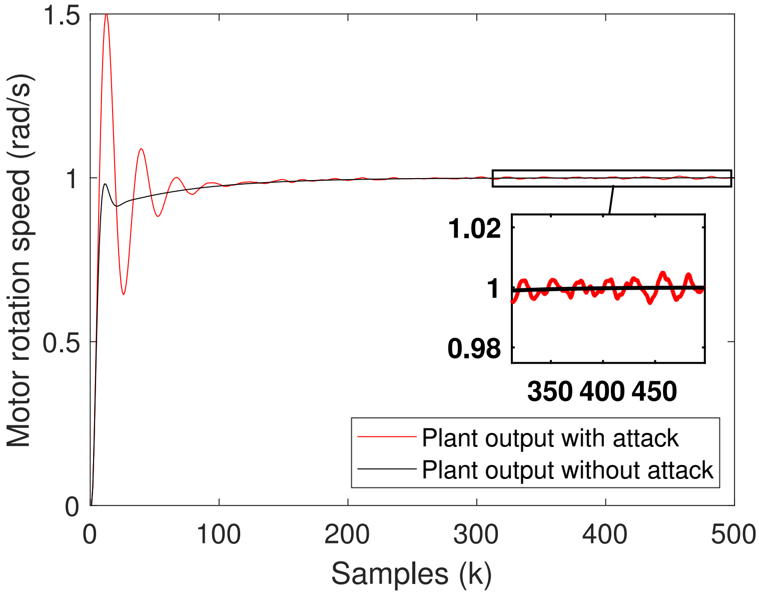
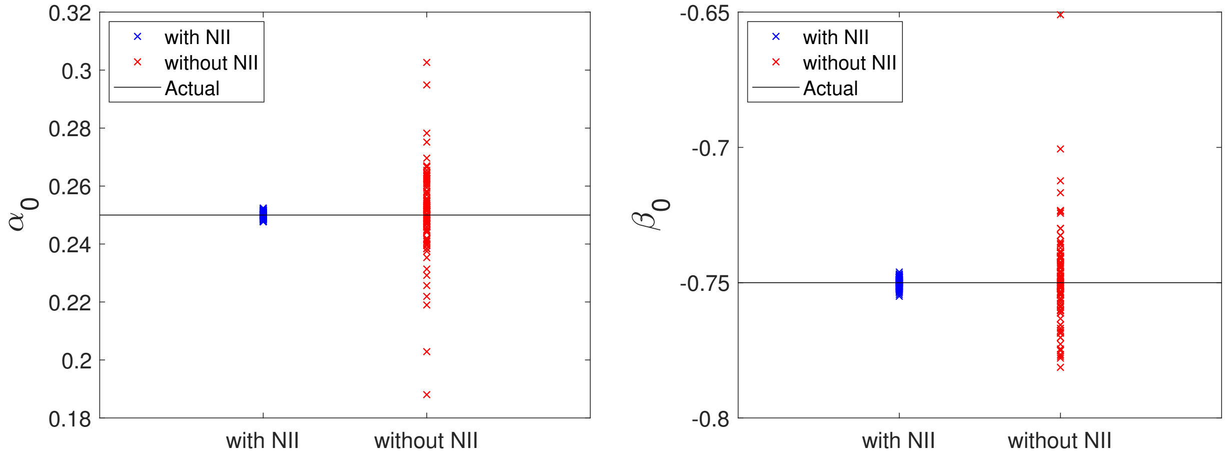


Figure 40 – Motor speed with and without attack.

For the execution of Algorithm 1 the signals $w(k)$ and $y''(k)$ are recorded during 100 samples, starting when the system achieves its steady state regarding to $r(k)$. Thus, the size of signals $w(k)$ and $y''_1(k)$ used by the BSA in (4.9) and (4.11), respectively, is $N = 100$ samples. For the execution of Algorithm 3 the signals $w(k)$ and $y''(k)$ are recorded during 0,5$Msamples$, also starting when the system achieves its steady state regarding to $r(k)$. Recall that in algorithm Algorithm 3, the recorded signals are not directly applied to the BSA process. They are processed through the NII stage to result in $\bar{\omega}_j(0)\delta(k)$ and $\Upsilon(k)$. The signals $\bar{\omega}_j(0)\delta(k)$ and $\Upsilon(k)$ used by the BSA in (4.28) and (4.29), respectively, are sized with $\mathcal{N} = 100$ samples. This way, the signals processed by the BSA have the same size in both algorithms 1 and 3 (*i.e.* $\mathcal{N} = N$). The amplitude threshold of the NII is $\Omega = 0.01$, which means that the condition defined in Algorithm 2 (*i.e.* $w(k) \geq \Omega$) is true in approximately 2.28% of the samples of $w(k)$.

Figure 41 shows the 100 values of α_0 and β_0 estimated by the identification processes with and without the NII stage (*i.e.*, with Algorithms 3 and 1, respectively). Additionally, Table 7 shows the statistics of the results presented in Figure 41. From Figure 41 and Table 7, it is possible to verify that the accuracy of the attack identification algorithm with the NII stage is better than the accuracy obtained without the proposed technique. Figure 41 demonstrates that, with the NII stage, the estimated values of α_0 and β_0 are closer to their actual values – *i.e.*, less spread – than without the NII stage. Note that, the statistics shown in Table 7 ratifies the better performance provided by the NII stage. In this case, the means of the estimated values are closer to the to the real values of α_0 and β_0, with lower standard deviation.



(a) Estimations of α_0

(b) Estimations of β_0

Figure 41 – Estimations of α_0 and β_0 with and without the NII stage.

Table 7 – Statistics of the attack identification proccess

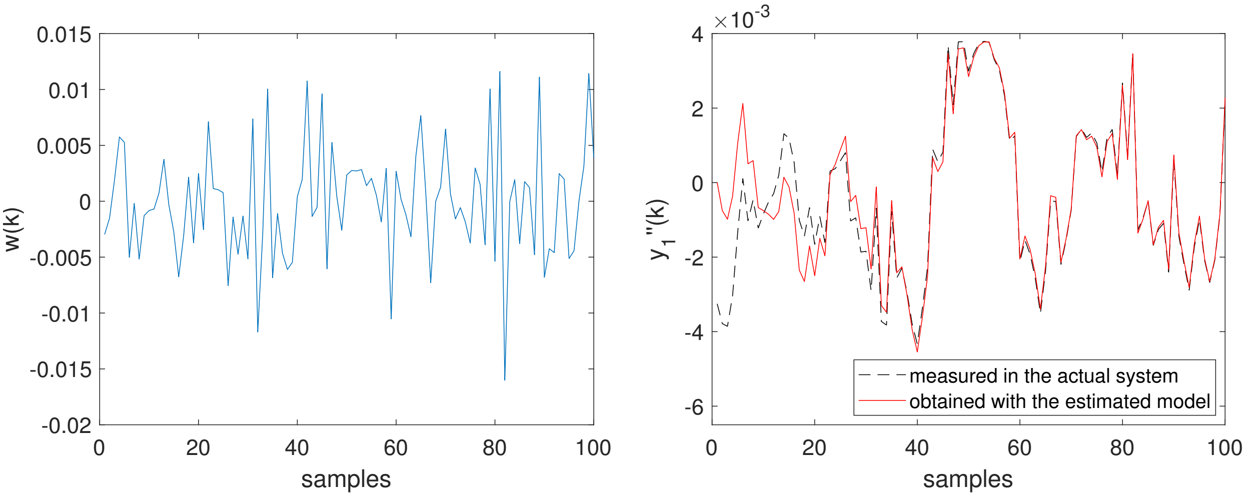
Coefficient	Algorithm	Mean	Standard Deviation
α_0	with NII	0.2500	0.0011
	without NII	0.2506	0.0147
β_0	with NII	-0.7502	0.0017
	without NII	-0.7485	0.0172

Figure 42 shows the input and output signals used by the BSA to estimate $M(z)$ in a simulation example performed with Algorithm 1 (without the NII stage). Figure 42(a) shows the noise $w(k)$ recorded in the actual system and used by the BSA as input for the model defined by (4.8). Figure 42(b) shows in black dashed line the signal $y_1''(k)$ measured in the actual system and used by the BSA as the reference output for the model defined by (4.8). Additionaly, Figure 42(b) shows in red line the signal $y_1''(k)$ produced by the estimated model – *i.e.* the model (4.8) containing the estimated attack function – when excited by the noise input shown in Figure 42(a). In Figure 42(b), it is possible to see that the output $y_1''(k)$ obtained with the estimated model does not completely match the output $y_1''(k)$ measured in the actual system. It exemplifies, as shown in Figure 41 and Table 7, the lower accuracy of Algorithm 1 when identifying $M(z)$.

Figure 43, in turn, shows the input and output signals used by the BSA to estimate $M(z)$ in a simulation example performed with Algorithm 3 (with the NII stage). Figure 43(a) shows the weighted impulse $\bar{\omega}_j(0)\delta(k)$ produced by the NII stage and used by the BSA as input for the model defined in (4.27). Figure 43(b) shows:

- In black dashed line: the integrated signal $\Upsilon(k)$ produced by the NII stage (based on measurements in the actual system) and used by the BSA as the reference output for the model defined in (4.27);

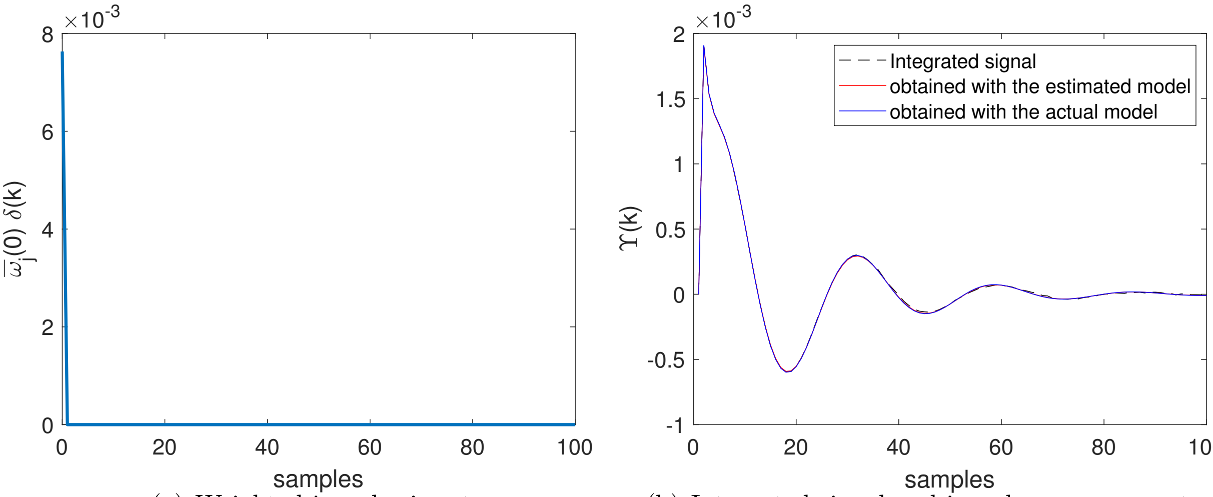
- In blue line: the impulse response produced when the weighted impulse $\bar{\omega}_j(0)\delta(k)$, shown Figure 43(a), is applied to the system defined in (4.27) containing the actual attack function;
- In red line: the impulse response produced when the weighted impulse $\bar{\omega}_j(0)\delta(k)$, shown Figure 43(a), is applied to the system defined in (4.27) containing the estimated attack function.



(a) Noise input signal

(b) Noisy output signals

Figure 42 – Input and output signals used by the BSA in Algorithm 1 to estimate $M(z)$ considering the model defined in (4.8).



(a) Weighted impulse input

(b) Integrated signal and impulse response outputs

Figure 43 – Input and output signals used by the BSA in Algorithm 3 to estimate $M(z)$ considering the model defined in (4.27).

From Figure 43(b), it is possible to see that the integrated signal (provided by the NII stage) accurately meets the impulse response of the actual system. It indicates that the NII technique is able to accurately reveal the impulse response of the system based on the signals produced by the white gaussian noise injected in the NCS. Additionally, Figure 43(b) shows that the impulse response obtained with the estimated model accurately meets the impulse response obtained with the actual system. It demonstrates that NII stage effectively contributes to enhance the accuracy of the identification process, as already shown in Figure 41 and Table 7.

The better performance obtained with the NII stage is mainly attributed to the cancelation of the initial conditions produced by the noise in the actual system. Note that, in Algorithm 1, the noise input was already present in the system since before $y_1''(k)$ was obtained, which makes $w(k)$ affect the initial conditions of the system. Thus, the lack of knowledge about the initial conditions of the system affects the estimation of the attack function in Algorithm 1. On the other hand, in Algorithm 3, the impact of $w(k)$ in the system's initial conditions is mitigated by the NII stage. This statement can be verified in equation (4.23), where $\Upsilon_1(k) \to 0$ when all $y_j(k)$ are integrated among all $j \in J$, as demonstrated in Section 4.2.2.2. Indeed, when the noise input $w(k)$ is transformed into a weighted impulse signal $\bar{\omega}_j(0)\delta(k)$, it is not expected to exist any initial conditions caused by $w(k)$ in the system defined in (4.27), given that $\bar{\omega}_j(0)\delta(k) = 0, \forall -\infty \leq k < 0$.

The results of this section indicates the effectiveness and accuracy of the proposed countermeasure when identifying SD-Controlled Data Injection attacks in NCSs, specially when the NII technique is used. In a normal conditions, when the system is not under attack, the injected noise is cancelled and does not affect the NCS. When the system is under attack, it is possible to see that noise is present in the plant output, but it is small due the parameters chosen for $w(k)$. It should be noted that such small noise is not necessarily a drawback for the system, however, the possible impacts of this noise in case of attack have to be evaluated for each specific system.

6.4 SUMMARY

This chapter evaluates the performance of the countermeasures presented in Chapter 4, namely:

- The switching controller design to mitigate the System Identification attacks presented in Section 3.1; and
- The link monitoring strategy to identify the SD-Controlled Data Injection attack described in Section 3.2.1.

The simulations demonstrate that the proposed switching controller design is able to mitigate both Passive and Active System Identification attacks, making the model obtained by the attacker imprecise and ambiguous. Also, the simulations demonstrate that the performance of the proposed countermeasure is satisfactory from the control perspective. Considering the control aspects, in general, the proposed countermeasure presents a performance similar to the performance of a non-switching controller, with an increase in the system's settling time caused by the successive switchings among control functions. Therefore, when deciding for using this countermeasure, it must be considered the existence of a tradeoff between mitigating the identification attack and increasing the system's settling time – which, depending on the plant, is not necessarily a drawback.

Regarding the countermeasure to identify SD-Controlled Data Injection attacks, the results indicate the effectiveness and accuracy of proposed identification scheme, specially when the NII technique is used. The outcomes show that the NII technique is able to accurately reveal the impulse response of the system based on the signals produced by the white gaussian noise injected in the NCS, ratifying the theoretical demonstration presented in Section 4.2.2.2. The better performance obtained with the NII is mainly attributed to the cancelation of the initial conditions produced by the noise injected in the system, which can also be verified in equation (4.23). Moreover, the simulation results show that when the system is not under attack, the injected noise is cancelled and does not affect the NCS (as expected based on the discussion presented in Section 4.2.1). When the system is under attack, it is possible to see that the noise is present in the plant output, which may not necessarily be drawback for the system. However, the possible impacts of such noise (in case of attack) have to be evaluated for each specific system.

7 CONCLUSIONS AND FUTURE WORKS

A brief summary of the contributions of this research and possible future works are discussed in this Chapter. Section 7.1 brings the conclusions regarding the results already obtained. Section 7.2 presents the publications and the award obtained to date with this work. Section 7.3 indicates possible future works that may derive from this research.

7.1 CONCLUSIONS

The present work proposes, in a first stage, two System Identification attacks in NCSs: the Passive System Identification attack; and the Active System Identification attack. These attacks are intended to estimate the LTI transfer functions of NCSs and are implemented based on bio-inspired metaheuristics – specifically the BSA and PSO. The referred System Identification attacks, which belong to the category of Cyber-physical Intelligence attacks, are developed to support the design of covert/model-based attacks in NCSs. The Passive System Identification attack does not interfere in the system to perform the identification task. However, it requires the occurrence of events that produce signals rich enough for the identification process. On the other hand, the Active System Identification attack injects an attack signal in the NCS to produce the signals necessary for the identification process.

In addition to the referred system identification attacks, this study covered three different model-based offensives:

- the SD-Controlled Data Injection attack;
- the Covert Misappropriation attack proposed in (SMITH, 2011; SMITH, 2015);
- the novel SD-Controlled Data Loss attack.

The simulation results show that the information provided by the proposed System Identification attacks allow the effective design of covert/model-based offensives against NCSs (even if the captured data is impaired by data loss or noise). Moreover, the results provide conclusive data on the effectiveness and potential impacts of the joint operation of the aforementioned System Identification attacks and model-based offensives against industrial devices – such as a DC motor – or critical infrastructure facilities – such as a large PHWR.

Regarding the SD-Controlled Data Loss attack, proposed in this work, the outcomes demonstrate that it is able to produce the same accurate and harmful behaviors of the SD-Controlled Data Injection attack, however, without the need to overcome eventual security mechanisms for data integrity and authenticity that may hinder an SD-Controlled Data Injection attack. Additionally, the results demonstrate that, based on the models learned through the System Identification attack, this model-based offensive is able to smartly decide which packets the NCS must lose – through malicious interferences – in order to degrade plant service, taking special care to avoid the indiscriminate loss of samples. This attack approach prevents the complete denial of communication, which makes the attack more difficult to be noticed than an arbitrary DoS attacks.

To support the discussion about the relationship between System Identification attacks and covert/model-based attacks in NCSs, the present work introduces a novel taxonomy, which is another contribution of this research to the literature on cybersecurity of NCSs. This taxonomy also sets the requirements for the attacks discussed in this work, which helps on the development of layered defense strategies against System Identification attacks and covert/model-based offensives.

In order to contribute to the security of NCSs, this thesis proposes two countermeasures for situations where the NCS suffers from eventual failure or lack of other conventional security mechanisms – such as encryption, authentication, and network segmentation. The first countermeasure aims to hinder system identification attacks. The second countermeasure aims to detect/identify SD-Controlled Data Injection attacks.

Towards the first countermeasure, the analysis of the system identification processes as feasible attacks led to the development of a switching controller design intended to hinder the identification task. This switching controller design take into account the need to meet simultaneously two objectives: comply with the plant control requirements; and hinder the identification process. The simulation results demonstrate that this countermeasure is able to mitigate the proposed System Identification attacks, making the model obtained by the attacker imprecise/ambiguous and, thus, discourage the implementation of covert/model-based attacks. At the same time, the simulations demonstrate that the performance of the proposed countermeasure is satisfactory from the control perspective. Considering the control aspects, in general, the simulations indicate that the proposed countermeasure presents a performance similar to the performance of a non-switching controller, with an increase in the system settling time. Small overshoots were also observed in simulations. Therefore, when deciding for using this countermeasure, it must be considered the existence of a tradeoff between mitigate the identification attack and increase the settling time of the system or even cause small overshoots –

which, depending on the plant, are not necessarily drawbacks. Besides the satisfactory performance obtained with this countermeasure, we encourage the development of a heuristic or an analytical method capable of providing control functions and switching rules that maximize the performance of the switching controller in both mentioned objectives.

The second countermeasure consists of a link monitoring strategy that uses white gaussian noise to detect/identify SD-Controlled Data Injection attacks. To increase its accuracy, the countermeasure is endowed with the Noise Impulse Integration (NII) technique, which was developed in this thesis using the radar pulse integration technique as inspiration. The results demonstrate the effectiveness and accuracy of the proposed countermeasure when identifying SD-Controlled Data Injection attacks in NCSs, specially when the NII technique is used. It is possible to see that the NII technique is able to accurately reveal the impulse response of the system with the attack, effectively contributing to enhance the accuracy of the estimated the attack function. The better performance obtained with the NII stage is mainly attributed to the mitigation of the initial conditions produced by the injected noise, which are cancelled according to the formulation of the NII technique. It is noteworthy that, when the system is not under attack, the injected noise does not affect the NCS. When the system is under attack, it is possible to see that a small noise is present in the plant output. Such small noise may not necessarily be a drawback, however, the possible impacts of this noise in case of attack have to be evaluated for each specific system.

7.2 PUBLICATIONS AND AWARD

To date, the contributions of this research resulted in the publication of the following papers, which are attached to this work in Appendices A to I:

I de Sá, A. O., Carmo, L. F. R. C., e Machado, R. C. S. (2016). Ataques Furtivos em Sistemas de Controle Físicos Cibernéticos. *Anais do XVI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2016)*, 128–141. Sociedade Brasileira de Computação.

II de Sá, A. O., Carmo, L. F. R. C., e Machado, R. C. S. (2017). Covert Attacks in Cyber-physical Control Systems. *IEEE Transactions on Industrial Informatics*, 13(4):1641–1651.[1]

[1] Paper II is an extended version of paper I, presented at SBSeg 2016.

III de Sá, A. O., Carmo, L. F. R. C., e Machado, R. C. S. (2017). Bio-inspired Active Attack for Adentification of Networked Control Systems. In *10th EAI Int. Conference on Bio-inspired Information and Communications Technologies (BICT 2017)*, 1–8. ACM.

IV de Sá, A. O., Carmo, L. F. R. C., e Machado, R. C. S. (2017). Bio-inspired Active System Identification: a Cyber-physical Intelligence Attack in Networked Control Systems. *Mobile Networks and Applications*, 1–14, Springer.[(2)]

V de Sá, A. O., Carmo, L. F. R. C., e Machado, R. C. S. (2017). Use of Switching Controllers for Mitigation of Active Identification Attacks in Networked Control Systems. In *Proceedings of the IEEE Cyber Science and Technology Congress*, 257–262. IEEE.

VI de Sá, A. O., Carmo, L. F. R. C., e Machado, R. C. S. (2018). A Controller Design for Mitigation of Passive System Identification Attacks in Networked Control Systems. *Journal of Internet Services and Applications*, 9(1):1–19, Springer.

VII de Sá, A. O., Carmo, L. F. R. C., e Machado, R. C. S. (2018). Evaluation on Passive System Identification and Covert Misappropriation attacks in Large Pressurized Heavy Water Reactors. In *2018 IEEE International Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0&IoT 2018)*, 203–208. IEEE.

VIII de Sá, A. O. et al. (2019) Bio-inspired system identification attacks in noisy networked control systems. In: *11th EAI International Conference on Bio-inspired Information and Communications Technologies (BICT 2019)*, 1–11. Springer.

IX de Sá, A. O., Carmo, L. F. R. C., e Machado, R. C. S. (2019) Countermeasure for Identification of Controlled Data Injection Attacks in Networked Control Systems. In: *2019 IEEE International Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0&IoT 2019)*, Accepted for publication.

The present research was also awarded with the first place in the Student Contest of the 2018 IEEE International Workshop on Metrology for Industry 4.0 and IoT (Brescia, Italy), with the poster: "Covert Attacks and Challenges for Metrology in Industrial Control Systems".

[2] Paper IV is an extended version of paper III, presented at BICT 2017.

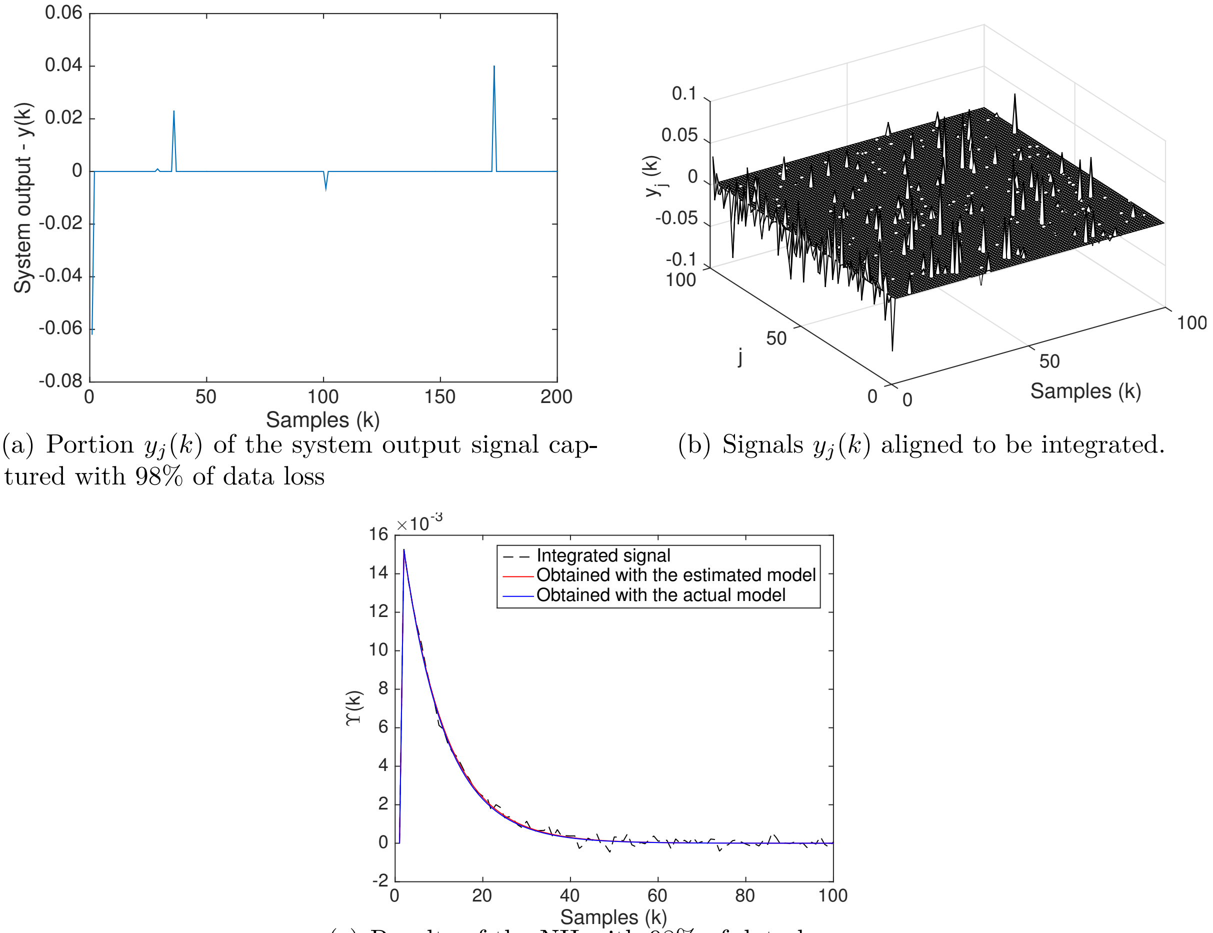
7.3 FUTURE WORKS

From the results obtained in this research, it is possible to identify some opportunities for future works and research directions:

- **Generalize the proposed System Identification attacks to LTI systems of unknown order**: In the System Identification attacks described in Section 3.1, the system identification technique considers that the attacker knows the order of the LTI function of the attacked devices – which is feasible when the attacker knows what kind of plant and controller the attacked NCS has. We estimate that, by analysing the magnitude of the residues of the estimated function, as well as the global minimum value found by the metaheuristic, it is possible to infer the system's order and extend the already developed algorithms to LTI systems of unknown order. This generalization would also benefit the countermeasure for identification of controlled data injection attacks, presented in Section 4.2, in identifying LTI attack functions of unknown order.

- **Optimize the switching control strategy as countermeasure for mitigating System Identification attacks:** as defined in Section 4.1, the strategy of using a switching controller as countermeasure for System Identification attacks take into account the need to meet simultaneously two objectives: comply with the plant control requirements; and hinder the identification process. Despite the satisfactory performance already obtained with this countermeasure, we believe that the use of a multi-objective optimization metaheuristic – such as the multi-objective particle swarm optimization (MOPSO) (COELLO; LECHUGA, 2002) – may be a path to obtain control functions and switching rules that maximize the performance of the switching controller in both mentioned (and potentially conflicting) objectives.

- **Optimize the SD-Controlled Data Loss attack:** the results obtained in the present work demonstrate that the SD-Controlled Data Loss attack is able to accurately produce harmful behaviors in a plant by causing the loss of specific packets transmitted in the NCS. In this work, the attack solution (*i.e.* attack sequence) is found by optimizing a fitness function that specifies the desired harmful behavior. Preliminary results indicate that it is possible add another objective to this problem, in order to reduce (even more) the number of packets to be dropped and still cause the same kind of harmful behavior to the plant. In this case a multi-objective optimization metaheuristic – such as the multi-objective particle swarm optimization (MOPSO) (COELLO; LECHUGA, 2002) – may be used to obtain such optimized attack sequences.



(a) Portion $y_j(k)$ of the system output signal captured with 98% of data loss

(b) Signals $y_j(k)$ aligned to be integrated.

(c) Results of the NII with 98% of data loss.

Figure 44 – NII with 98% of data loss.

- **Use of the NII technique as an attack tool in scenarios with high data loss**: this work demonstrates that the NII technique is effective in revealing the impulse response of attack functions executed by SD-Controlled Data Injection attacks. The technique is used here in a countermeasure, aiming a scenario where the monitored signals are not impaired by data loss. However, preliminary results indicate that the NII technique may be a useful tool to rebuild and reveal the impulse response functions of LTI systems in scenarios where the captured data is impaired by high percentage of loss. Such ability can be used, for instance, to enhance System Identifications attacks in scenarios with extreme data loss – as in the case of an attacker far from the WNCS transmitters, with poor connectivity, trying to identify the WNCS models. It is worth mentioning that, in this case, the system must naturally have a source of white gaussian noise exciting its devices. Figure 44 shows preliminary results were the NII technique is used in a simulated scenario with 98% of data loss. The actual system $H(z)$, to be identified, is represented by (7.1). The estimated system $H_e(z)$, after the identification process using the NII technique, is represented by (7.2). The monitoring time to obtain these results corresponds to 3×10^8 samples. It is possible to see from Figure 44(c) that the NII was able to rebuild and reveal the impulse response of the actual system, even with 98% of data loss, allowing an accurate identification of the attacked system. Although preliminary results indicate the feasibility of using the NII technique in scenarios with high data loss, it is still necessary to demonstrate this property from the theoretical basis presented in Section 4.2.2.2 of this thesis, which is planned to be done as future work.

$$H(z) = \frac{2}{z - 0.9} \tag{7.1}$$

$$H_e(z) = \frac{1.998}{z - 0.902} \tag{7.2}$$

REFERENCES

AHMED, S. Novel noncoherent radar pulse integration to combat noise jamming. **IEEE Transactions on Aerospace and Electronic systems**, IEEE, v. 51, n. 3, p. 2350–2359, 2015.

AKERBERG, J.; BJORKMAN, M. Exploring security in profinet io. In: IEEE. **2009 33rd Annual IEEE International Computer Software and Applications Conference**. [S.l.], 2009. v. 1, p. 406–412.

AKPINAR, K. O.; OZCELIK, I. Development of the ecat preprocessor with the trust communication approach. **Security and Communication Networks**, Hindawi, v. 2018, 2018.

AMIN, S. et al. Cyber security of water scada systems part i: analysis and experimentation of stealthy deception attacks. **IEEE Transactions on Control Systems Technology**, IEEE, v. 21, n. 5, p. 1963–1970, 2013.

AMIN, S. et al. Cyber security of water scada systems part ii: Attack detection using enhanced hydrodynamic models. **IEEE Transactions on Control Systems Technology**, IEEE, v. 21, n. 5, p. 1679–1693, 2013.

BANERJEE, S. et al. An interval approach for robust control of a large phwr with pid controllers. **IEEE Transactions on Nuclear Science**, IEEE, v. 62, n. 1, p. 281–292, 2015.

BOU-HARB, E.; DEBBABI, M.; ASSI, C. Cyber scanning: a comprehensive survey. **IEEE Communications Surveys & Tutorials**, IEEE, v. 16, n. 3, p. 1496–1519, 2014.

CHEN, X.; SONG, Y.; YU, J. Network-in-the-loop simulation platform for control system. In: **AsiaSim 2012**. [S.l.]: Springer, 2012. p. 54–62.

CHOW, M.-Y.; TIPSUWAN, Y. Network-based control systems: a tutorial. In: IEEE. **Industrial Electronics Society, 2001. IECON'01. The 27th Annual Conference of the IEEE**. [S.l.], 2001. v. 3, p. 1593–1602.

CIVICIOGLU, P. Backtracking search optimization algorithm for numerical optimization problems. **Applied Mathematics and Computation**, Elsevier, v. 219, n. 15, p. 8121–8144, 2013.

COELLO, C. C.; LECHUGA, M. S. Mopso: A proposal for multiple objective particle swarm optimization. In: IEEE. **Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)**. [S.l.], 2002. v. 2, p. 1051–1056.

COLLANTES, M. H.; PADILLA, A. L. **Protocols and Network Security in ICS Infrastructures**. [S.l.], 2015.

DAS, M. et al. Network control system applied to a large pressurized heavy water reactor. **IEEE Transactions on Nuclear Science**, IEEE, v. 53, n. 5, p. 2948–2956, 2006.

DASGUPTA, S. et al. Stability of networked control system (ncs) with discrete time-driven pid controllers. **Control Engineering Practice**, Elsevier, v. 42, p. 41–49, 2015.

DASGUPTA, S. et al. Networked control of a large pressurized heavy water reactor (phwr) with discrete proportional-integral-derivative (pid) controllers. **IEEE Transactions on Nuclear Science**, IEEE, v. 60, n. 5, p. 3879–3888, 2013.

DORF, R. C.; BISHOP, R. H. **Modern control systems**. [S.l.]: Pearson, 2011.

DRIAS, Z.; SERHROUCHNI, A.; VOGEL, O. Taxonomy of attacks on industrial control protocols. In: IEEE. **2015 International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS)**. [S.l.], 2015. p. 1–6.

EL-SHARKAWI, M.; HUANG, C. Variable structure tracking of dc motor for high performance applications. **Energy Conversion, IEEE Transactions on**, IEEE, v. 4, n. 4, p. 643–650, 1989.

FALLIERE, N.; MURCHU, L. O.; CHIEN, E. W32. stuxnet dossier. **White paper, Symantec Corp., Security Response**, v. 5, n. 6, p. 29, 2011.

FAROOQUI, A. A. et al. Cyber security backdrop: A scada testbed. In: IEEE. **Computing, Communications and IT Applications Conference (ComComAp), 2014 IEEE**. [S.l.], 2014. p. 98–103.

FERRARA, A.; SACONE, S.; SIRI, S. A switched ramp-metering controller for freeway traffic systems. **IFAC-PapersOnLine**, Elsevier, v. 48, n. 27, p. 105–110, 2015.

FERRARI, P. et al. Improving simulation of wireless networked control systems based on wirelesshart. **Computer Standards & Interfaces**, Elsevier, v. 35, n. 6, p. 605–615, 2013.

GRANAT, A.; HÖFKEN, H.; SCHUBA, M. Intrusion detection of the ics protocol ethercat. **DEStech Transactions on Computer Science and Engineering**, n. cnsce, 2017.

GUPTA, R. A.; CHOW, M.-Y. Overview of networked control systems. In: **Networked Control Systems**. [S.l.]: Springer, 2008. p. 1–23.

GUPTA, R. A.; CHOW, M.-Y. Networked control system: overview and research trends. **Industrial Electronics, IEEE Transactions on**, IEEE, v. 57, n. 7, p. 2527–2535, 2010.

HAHN, A. Operational technology and information technology in industrial control systems. In: **Cyber-security of SCADA and Other Industrial Control Systems**. [S.l.]: Springer, 2016. p. 51–68.

HESPANHA, J. P.; MORSE, A. S. Stability of switched systems with average dwell-time. In: IEEE. **Decision and Control, 1999. Proceedings of the 38th IEEE Conference on**. [S.l.], 1999. v. 3, p. 2655–2660.

HESPANHA, J. P.; NAGHSHTABRIZI, P.; XU, Y. A survey of recent results in networked control systems. **Proceedings of the IEEE**, IEEE, v. 95, n. 1, p. 138–162, 2007.

HUSSAIN, A.; HEIDEMANN, J.; PAPADOPOULOS, C. A framework for classifying denial of service attacks. In: ACM. **Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications**. [S.l.], 2003. p. 99–110.

HWANG, H. et al. A study on mitm (man in the middle) vulnerability in wireless network using 802.1 x and eap. In: IEEE. **Information Science and Security, 2008. ICISS. International Conference on**. [S.l.], 2008. p. 164–170.

JAZDI, N. Cyber physical systems in the context of industry 4.0. In: IEEE. **Automation, Quality and Testing, Robotics, 2014 IEEE International Conference on**. [S.l.], 2014. p. 1–4.

KENNEDY J. E EBERHART, R. Particle swarm optimization. In: **Proceedings of 1995 IEEE International Conference on Neural Networks**. [S.l.: s.n.], 1995. p. 1942–1948.

KHATRI, S. et al. A taxonomy of physical layer attacks in manet. **International Journal of Computer Applications**, Foundation of Computer Science, v. 117, n. 22, 2015.

KROMBHOLZ, K. et al. Advanced social engineering attacks. **Journal of Information Security and applications**, Elsevier, v. 22, p. 113–122, 2015.

LANGNER, R. Stuxnet: Dissecting a cyberwarfare weapon. **Security & Privacy, IEEE**, IEEE, v. 9, n. 3, p. 49–51, 2011.

LASI, H. et al. Industry 4.0. **Business & Information Systems Engineering**, Springer, v. 6, n. 4, p. 239–242, 2014.

LIBERZON, D.; MORSE, A. S. Basic problems in stability and design of switched systems. **IEEE Control systems**, IEEE, v. 19, n. 5, p. 59–70, 1999.

LIN, H.; ANTSAKLIS, P. J. Stability and stabilizability of switched linear systems: a survey of recent results. **IEEE Transactions on Automatic control**, IEEE, v. 54, n. 2, p. 308–322, 2009.

LONG, M.; WU, C.-H.; HUNG, J. Y. Denial of service attacks on network-based control systems: impact and mitigation. **Industrial Informatics, IEEE Transactions on**, IEEE, v. 1, n. 2, p. 85–96, 2005.

MATHUR, A. P.; TIPPENHAUER, N. O. Swat: a water treatment testbed for research and training on ics security. In: IEEE. **2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)**. [S.l.], 2016. p. 31–36.

MCLAUGHLIN, S. et al. The cybersecurity landscape in industrial control systems. **Proceedings of the IEEE**, IEEE, v. 104, n. 5, p. 1039–1057, 2016.

MO, Y. et al. Cyber–physical security of a smart grid infrastructure. **Proceedings of the IEEE**, IEEE, v. 100, n. 1, p. 195–209, 2012.

MO, Y.; SINOPOLI, B. Secure control against replay attacks. In: IEEE. **2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)**. [S.l.], 2009. p. 911–918.

MORSE, A. S. Supervisory control of families of linear set-point controllers-part i. exact matching. **IEEE Transactions on Automatic Control**, IEEE, v. 41, n. 10, p. 1413–1431, 1996.

MULLER, I.; NETTO, J. C.; PEREIRA, C. E. Wirelesshart field devices. **IEEE Instrumentation & Measurement Magazine**, IEEE, v. 14, n. 6, p. 20–25, 2011.

ÖNCÜ, S. et al. Cooperative adaptive cruise control: Network-aware analysis of string stability. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 15, n. 4, p. 1527–1537, 2014.

PANG, Z.-H.; LIU, G.-P. Design and implementation of secure networked predictive control systems under deception attacks. **IEEE Transactions on Control Systems Technology**, IEEE, v. 20, n. 5, p. 1334–1342, 2012.

PASQUALETTI, F.; DORFLER, F.; BULLO, F. Attack detection and identification in cyber-physical systems. **Automatic Control, IEEE Transactions on**, IEEE, v. 58, n. 11, p. 2715–2729, 2013.

PASQUALETTI, F.; DORFLER, F.; BULLO, F. Control-theoretic methods for cyberphysical security: Geometric principles for optimal cross-layer resilient control systems. **IEEE Control Systems Magazine**, IEEE, v. 35, n. 1, p. 110–127, 2015.

PESCHKE, J. et al. Security in industrial ethernet. In: **Proceedings of the 11th IEEE International Conference on Emerging Technologies and Factory Automation**. [S.l.: s.n.], 2006. p. 1214–1221.

PETERSEN, S.; CARLSEN, S. Wirelesshart versus isa100. 11a: The format war hits the factory floor. **IEEE Industrial Electronics Magazine**, v. 4, n. 5, p. 23–34, 2011.

PFRANG, S.; MEIER, D. On the detection of replay attacks in industrial automation networks operated with profinet io. In: **ICISSP**. [S.l.: s.n.], 2017. p. 683–693.

RAMOS, C.; VALE, Z.; FARIA, L. Cyber-physical intelligence in the context of power systems. In: **Future Generation Information Technology**. [S.l.]: Springer, 2011. p. 19–29.

SÁ, A. O. de; CARMO, L. F. R. d. C.; MACHADO, R. C. S. Evaluation on passive system identification and covert misappropriation attacks in large pressurized heavy water reactors. In: IEEE. **2018 Workshop on Metrology for Industry 4.0 and IoT**. [S.l.], 2018. p. 203–208.

SA, A. O. de; CARMO, L. F. R. da C.; MACHADO, R. C. S. Bio-inspired active attack for identification of networked control systems. In: **10th EAI International Conference on Bio-inspired Information and Communications Technologies (BICT)**. New Jersey, USA: ACM, 2017. p. 1–8.

SA, A. O. de; CARMO, L. F. R. da C.; MACHADO, R. C. S. Bio-inspired active system identification: a cyber-physical intelligence attack in networked control systems. **Mobile Networks and Applications**, Springer, p. 1–14, 2017.

SA, A. O. de; CARMO, L. F. R. da C.; MACHADO, R. C. S. Covert attacks in cyber-physical control systems. **IEEE Transactions on Industrial Informatics**, v. 13, n. 4, p. 1641–1651, Aug 2017. ISSN 1551-3203.

SA, A. O. de; CARMO, L. F. R. da C.; MACHADO, R. C. S. Use of switching controllers for mitigation of active identification attacks in networked control systems. In: **2017 IEEE Cyber Science and Technology Congress (CyberSciTech2017)**. Orlando, FL, USA: IEEE, 2017. p. 1–6.

SA, A. O. de; CARMO, L. F. R. da C.; MACHADO, R. C. S. A controller design for mitigation of passive system identification attacks in networked control systems. **Journal of Internet Services and Applications**, Springer, v. 9, n. 1, p. 1–19, Feb 2018.

SA, A. O. de et al. Bio-inspired system identification attacks in noisy networked control systems. In: **11th EAI International Conference on Bio-inspired Information and Communications Technologies (BICT)**. Pittsburgh, USA: Springer, 2019. p. 1–11.

SÁ, A. O. de; NEDJAH, N.; MOURELLE, L. de M. Distributed efficient localization in swarm robotic systems using swarm intelligence algorithms. **Neurocomputing**, Elsevier, v. 172, p. 322–336, 2016.

SABĂU, Ş. et al. Optimal distributed control for platooning via sparse coprime factorizations. **IEEE Transactions on Automatic Control**, IEEE, v. 62, n. 1, p. 305–320, 2017.

SACHS, L. **Applied statistics: a handbook of techniques**. [S.l.]: Springer Science & Business Media, 2012.

SADI, Y.; ERGEN, S. C.; PARK, P. Minimum energy data transmission for wireless networked control systems. **IEEE Transactions on Wireless Communications**, IEEE, v. 13, n. 4, p. 2163–2175, 2014.

SAFAEI, F. R. P. et al. Stability of an adaptive switched controller for power system oscillation damping using remote synchrophasor signals. In: IEEE. **Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on**. [S.l.], 2014. p. 1695–1700.

SCHWARTZ, M. Effects of signal fluctuation on the detection of pulse signals in noise. **IRE Transactions on Information Theory**, IEEE, v. 2, n. 2, p. 66–71, 1956.

SHI, Y.; HUANG, J.; YU, B. Robust tracking control of networked control systems: application to a networked dc motor. **IEEE Transactions on Industrial Electronics**, IEEE, v. 60, n. 12, p. 5864–5874, 2013.

SI, M. L. et al. Study on sample rate and performance of a networked control system by simulation. In: TRANS TECH PUBL. **Advanced Materials Research**. [S.l.], 2010. v. 139, p. 2225–2228.

SKAFIDAS, E. et al. Stability results for switched controller systems. **Automatica**, Elsevier, v. 35, n. 4, p. 553–564, 1999.

SKOLNIK, M. I. **Radar Handbook**. [S.l.]: McGraw-Hill, 1990. (Electronic engineering series). ISBN 9780070579132.

SMITH, R. A decoupled feedback structure for covertly appropriating networked control systems. In: **Proceedings of the 18th IFAC World Congress 2011**. Milano, Italy: IFAC-PapersOnLine, 2011. v. 18, n. 1.

SMITH, R. S. Covert misappropriation of networked control systems: Presenting a feedback structure. **Control Systems, IEEE**, IEEE, v. 35, n. 1, p. 82–92, 2015.

SNOEREN, A. C. et al. Single-packet ip traceback. **IEEE/ACM Transactions on Networking (ToN)**, IEEE Press, v. 10, n. 6, p. 721–734, 2002.

STALLINGS, W. **Cryptography and network security: principles and practices**. New Jersey, USA: Pearson Education India, 2006.

STOUFFER, K. et al. Nist special publication 800-82, revision 2: Guide to industrial control systems (ics) security. **Gaithersburg, MD, USA: National Institute of Standards and Technology**, 2015.

TEIXEIRA, A. **Toward Cyber-Secure and Resilient Networked Control Systems**. Tese (Doutorado) — KTH Royal Institute of Technology, 2014.

TEIXEIRA, A. et al. A secure control framework for resource-limited adversaries. **Automatica**, Elsevier, v. 51, p. 135–148, 2015.

TIPSUWAN, Y.; CHOW, M.-Y.; VANIJJIRATTIKHAN, R. An implementation of a networked pi controller over ip network. In: IEEE. **Industrial Electronics Society, 2003. IECON'03. The 29th Annual Conference of the IEEE**. [S.l.], 2003. v. 3, p. 2805–2810.

TRAN, T.; HA, Q. P.; NGUYEN, H. T. Robust non-overshoot time responses using cascade sliding mode-pid control. **Journal of Advanced Computational Intelligence and Intelligent Informatics**, Fuji Technology Press Ltd, 2007.

TULLEKEN, H. J. Generalized binary noise test-signal concept for improved identification-experiment design. **Automatica**, Elsevier, v. 26, n. 1, p. 37–49, 1990.

UMA, M.; PADMAVATHI, G. A survey on various cyber attacks and their classification. **IJ Network Security**, v. 15, n. 5, p. 390–396, 2013.

WANG, J. **Identification of Switched Linear Systems**. Tese (Doutorado) — University of Alberta, 2013.

WANG, W.; LU, Z. Cyber security in the smart grid: Survey and challenges. **Computer Networks**, Elsevier, v. 57, n. 5, p. 1344–1371, 2013.

YUNG, J.; DEBAR, H.; GRANBOULAN, L. Security issues and mitigation in ethernet powerlink. In: SPRINGER. **Conference on Security of Industrial-Control-and Cyber-Physical Systems**. [S.l.], 2016. p. 87–102.

ZETTER, K. **Countdown to zero day: Stuxnet and the launch of the world's first digital weapon**. [S.l.]: Crown, 2014.

ZHAI, G. et al. Qualitative analysis of discrete-time switched systems. In: IEEE. **American Control Conference, 2002. Proceedings of the 2002**. [S.l.], 2002. v. 3, p. 1880–1885.

ZHANG, C.; FAN, Y.; HAO, Y. Informative property of the data set in a single-input single-output (siso) closed-loop system with a switching controller. **Chinese Journal of Chemical Engineering**, Elsevier, v. 20, n. 6, p. 1128–1135, 2012.

ZHANG, L.; GAO, H.; KAYNAK, O. Network-induced constraints in networked control systems: a survey. **IEEE Transactions on Industrial Informatics**, IEEE, v. 9, n. 1, p. 403–416, 2013.

ZHANG, L. et al. Security solutions for networked control systems based on des algorithm and improved grey prediction model. **International Journal of Computer Network and Information Security**, Modern Education and Computer Science Press, v. 6, n. 1, p. 78, 2013.

ZOU, Y.; WANG, G. Intercept behavior analysis of industrial wireless sensor networks in the presence of eavesdropping attack. **IEEE Transactions on Industrial Informatics**, IEEE, v. 12, n. 2, p. 780–787, 2016.

APPENDIX A

IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 13, NO. 4, AUGUST 2017 1641

Covert Attacks in Cyber-Physical Control Systems

Alan Oliveira de Sá, Luiz F. Rust da Costa Carmo, and Raphael C. S. Machado

***Abstract*—The advantages of using communication networks to interconnect controllers and physical plants motivate the increasing number of networked control systems in industrial and critical infrastructure facilities. However, this integration also exposes such control systems to new threats, typical of the cyber domain. In this context, studies have been conducted, aiming to explore vulnerabilities and propose security solutions for cyber-physical systems. In this paper, a covert attack for service degradation is proposed, which is planned based on the intelligence gathered by another attack, herein proposed, referred as system identification attack. The simulation results demonstrate that the joint operation of the two attacks is capable to affect, in a covert and accurate way, the physical behavior of a system.**

***Index Terms*—Cyber-physical systems, networked control systems (NCSs), security.**

I. Introduction

THE integration of the systems used to control physical processes via communication networks aims to assign such systems better operational and management capabilities, as well as reduce its costs. Motivated by these advantages, there is a trend to have an increasing number of industrial process and critical infrastructure systems driven by networked control systems (NCS) [1]–[4], also referred to network-based control systems [5], [6]. As detailed in Fig. 1, an NCS consists of a controller, which runs a control function $C(z)$, a physical plant, described by its transfer function $G(z)$, and a communication network that interconnect both devices through a forward stream and a feedback stream. The forward stream connects the output of the controller to the plant's actuators. The feedback stream connects the output of the plant's sensors to the controller's input.

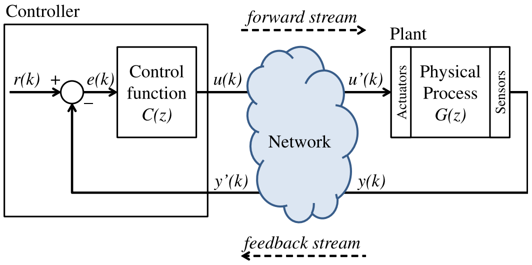


Fig. 1. Networked control system (NCS).

At the same time it brings several advantages, the integration of controllers and physical plants in a closed loop through a communication network also exposes such control systems to new threats, typical of the cyber domain. One possible way to attack an NCS, for example, is by hacking its software, i.e., changing the configuration or even the code executed by the controller, following a strategy similar to that used by the Stuxnet worm [7]. Another possible way for an attacker to negatively affect an NCS is by interfering on its communication process. Basically, an attacker may interfere in the forward and/or the feedback streams by three different means: inducing a jitter, causing data loss due to packet drop outs, or even injecting false data in the communication process.

In this context, studies have been conducted aiming to characterize vulnerabilities and to propose security solutions for the NCSs. In this work, a covert attack for service degradation (SD) is proposed, which consists of a novel joint operation of the following two attacks.

1) A system identification attack: executed to provide the attacker an accurate knowledge about the models of the targeted system, i.e., the plant's transfer function $G(z)$ and the controller's control function $C(z)$. This knowledge is obtained based on the signals that are collected from the input and output of the NCS's devices.
2) A data injection attack: where the attacker, as a Man-in-the-Middle (MitM), injects false data in the control loop of the NCS. The injected false data are computed based on the knowledge obtained by the attacker during the system identification attack, in order to covertly and accurately change the physical behavior of the plant.

Manuscript received July 27, 2016; revised October 28, 2016 and December 29, 2016; accepted February 18, 2017. Date of publication March 2, 2017; date of current version August 1, 2017. This work was supported in part by the Brazilian research agencies CNPq and FAPERJ. Paper no. TII-16-0756. *(Corresponding author: A. O. de Sá).*

A. O de Sá is with the Institute of Mathematics/NCE—Tércio Pacitti Institute, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, and also with the Admiral Wandenkolk Instruction Center, Brazilian Navy, Rio de Janeiro, Brazil (e-mail: alan.oliveira.sa@gmail.com).

L. F. R. da Costa Carmo is with the Institute of Mathematics/NCE—Tércio Pacitti Institute, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, and also with the National Institute of Metrology, Quality and Technology (INMETRO), Rio de Janeiro, Brazil (e-mail: lfrust@inmetro.gov.br).

R. C. S. Machado is with the National Institute of Metrology, Quality and Technology (INMETRO), Rio de Janeiro, Brazil, and also with the Federal Center for Technologic Education of Rio de Janeiro (CEFET/RJ) (e-mail: rcmachado@inmetro.gov.br).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2017.2676005

It is demonstrated that this joint operation is capable to degrade the service performed by a plant, through interventions that produce subtle changes on its physical behavior. Such interventions aim to reduce the efficiency of the plant or even cause damage in mid/long term. It is worth mentioning that an uncontrolled intervention in an NCS may lead the plant to an immediate breakdown, or even significantly change its behavior, which may cause the disclosure of the attack and the eventual failure of the operation. Thus, the changes driven by the attack herein proposed are dimensioned so that the modifications in the plant's behavior are physically difficult to be perceived. That is why the present attack is classified as physically covert.

To ensure that the attack to an NCS is physically covert, the attacker must plan his offensive based on an accurate knowledge about the system dynamics, otherwise the consequences of the attack may be unpredictable. One possible way to obtain such knowledge is through conventional intelligence operations, performed to collect information regarding to the design and the dynamics of the NCS. Another way to gather information about the targeted system is through what we refer in this paper as a *cyber-physical intelligence* attack. To this end, the mentioned system identification attack is proposed, which is based on the backtracking search optimization algorithm (BSA) [8]. As far as we know, there is no other system identification attack reported in the literature, which constitutes a novelty of this work. The BSA is specifically chosen to demonstrate the feasibility of the system identification attacks on NCSs. Although it is noteworthy that the use of the BSA to perform a system identification process was not reported earlier in the literature, which constitutes another novelty of this work. The attack herein proposed aims NCSs constituted by impulse-response systems, defined by linear time invariant (LTI) transfer functions, such as the NCSs presented in [1], [3], [4], and [9]–[13]. Examples of potential targets with this characteristic can range from noncritical industrial plants controlled by wireless networked control systems (WNCS) [14], [15] to large pressurized heavy water reactors (PHWR) [11], [16] or water canal systems [12], [13] controlled by wired NCSs. The well-known vulnerabilities of the cyber domain [17], [18], which may allow an attacker to have access to the control loop of an NCS, and the typical model of the aforementioned cyber-physical systems, which are consistent with the attack herein proposed, evidence why this attack may actually happen. Note that it includes targets with potentially significant impacts, such as the PHWR and the water canal systems.

This work motivated the formalization of a number of concepts related to covertness and intelligence in the context of the cyber-physical security. Thus, an additional contribution of this paper is the proposition of a terminology that encompasses a whole new class of attacks on cyber-physical systems. The proposed taxonomy establishes a new approach regarding to the covertness of attacks on cyber-physical systems, which must be analyzed from two aspects simultaneously: the physical and the cybernetic aspects.

It is worth mentioning that the objective of this work is not to facilitate covert attacks for SD in cyber-physical control systems. The purpose of this work is to demonstrate the degree of accuracy that may be achieved in this kind of attack, especially when supported by system identification attacks and, therefore, encourage the research for countermeasures to such attacks. The rest of this paper is organized as follows: first, in Section II, some related works are presented. Later, in Section III, a taxonomy regarding to the cyber-physical attacks that may happen in the control loop of an NCS is proposed. The attack herein proposed is then described in two parts, presented in Sections IV and V. In Section IV, the underlying details of the first part, consisting of a system identification attack are proposed. Then, in Section V, the second part is described, where data are injected into the NCS, based on the knowledge obtained through the system identification attack, to covertly degrade the plant's service. In Section VI, the results obtained through simulations of the attack, as well as a discussion for possible countermeasures are reported. Finally, in Section VII, some conclusions and possible future works are presented.

II. Related Works

The possibility of cyber-physical attacks became a reality after the launch of the Stuxnet worm [7], [19] and has been motivating research works concerning the security of NCSs. In this section, some works related to this subject are presented.

In [6], Long *et al.* propose two queueing models to evaluate the impact of delay jitter and packet loss in an NCS under attack. The attack is not designed taking into account a previous knowledge about the models of the controller and the physical plant. Thus, to affect the plant's behavior, the attacker arbitrarily floods the network with an additional traffic, causing jitter and packet loss. In this tactics, the excess of packets in the network can reduce the covertness of the attack, allowing the adoption of countermeasures, such as packet filtering [6] or blocking the malicious traffic on its origin [20]. Additionally, the arbitrary intervention in a system which the model is unknown may lead the plant to an extreme physical behavior, which is not desired if it is intended a covert attack.

In [4], Farooqui *et al.* present a supervisory control and data acquisition testbed using TrueTime, a MATLAB/Simulink based tool. They demonstrate an attack where a malicious agent sends false signals to the controller and to the actuator of an NCS. In that paper, the false signals are randomly generated aiming to make a DC motor lose its stability. This kind of attack does not require a previous knowledge about the plant and controller of the NCS. On the other hand, the desired physical effect and the covertness of the attack cannot be ensured due to the unpredictable consequences of the application of random false signals to a system which the model is unknown.

More recently, in [21], Teixeira *et al.* give a general framework for the analysis of a wide variety of methods of attack in NCSs. In their classification, it is stated that covert attacks in NCSs require high level of knowledge about the targeted system. Examples of covert attacks are provided in [9] and [13]. In these works the attacks are reformed by a MitM, where the attacker needs to know the model of the plant under attack and also inject false data into both the forward and the feedback streams. The covertness of the attacks described in [9] and [13], which depends on the difference between the actual model of

TABLE I
SYNTHESIS OF THE RELATED ATTACKS

Attack	Method	System knowledge	How the knowledge is obtained
Stuxnet *worm* [7], [19]	Modifications in the PLC code	Yes	Experiments in a real system
Long *et al.* [6]	Inducing *jitter* and packet loss	None	N/A
Farooqui *et al.* [4]	Data injection	None	N/A
Smith [9], [13]	Data injection	Yes	Not described
Teixeira [21]	Packet loss	None	N/A
	Data injection	Yes	Not described

the plant and the model used by the attacker, is analyzed from the perspective of the signals arriving to the controller, without addressing if the physical effects on the plant are noticeable, or if they are covert when faced by a human observer.

In [9], [21], and [13], where it is required a previous knowledge about the models of the NCS under attack, it is not described how the knowledge about the system is obtained by the attacker. It is just stated that a model is previously known to subsidize the design of the attack. The joint operation, herein proposed, of a covert attack for SD, supported by a system identification attack, aims to fill this hiatus, demonstrating how the data of an NCS can be obtained and how a covert attack can take advantage from it. Table I presents a synthesis of the characteristics of the attacks referred in this section.

III. TAXONOMY

In this section, it is presented a taxonomy concerning the possible attacks on cyber-physical control systems. In Section III-A, the attacks are briefly described and classified according to the way they act in the NCS. In Section III-B, a new approach for the analysis of the covertness of attacks in cyber-physical systems is proposed.

A. Classification of the Attacks

The attacks to cyber-physical control systems may take place on its devices—i.e., the controller, and the plant's sensors and actuators—and/or on its communication system, affecting the forward and the feedback streams. As a premise, we must consider that the *service* intended to be attacked/protected in such system is the work performed by the physical process, controlled by the NCS.

Considering the aforementioned definition of service in an NCS, the attacks may be classified within the following three different categories, as shown in Fig. 2.

1) Denial-of-Service (DoS) [22]: In an NCS, the DoS attacks comprehend all kind of cyber-physical attacks that deny the operation of the physical process, interrupting the execution of the work, or service, that the controlled plant is intended to do. The attack results, for example, in behaviors that may shut the plant down or even destroy it in a short term.

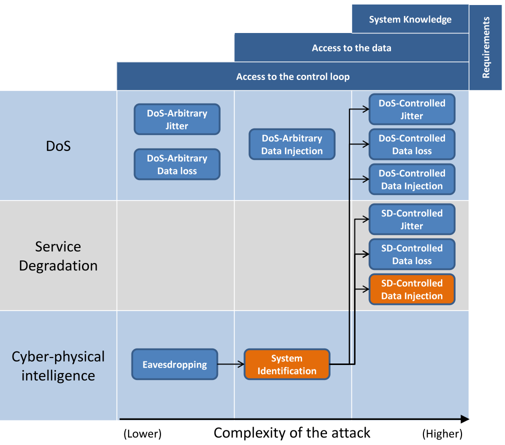


Fig. 2. Classification and requirements of the cyber-physical attacks that act in the control loop of an NCS.

2) Service degradation (SD): The SD attacks consist of malicious interventions that are done in the control loop in order to reduce the efficiency of the service, i.e., the efficiency of the physical process, or even reduce the mean time between failure (MTBF) of the plant in mid term or long term.
3) Cyber-physical intelligence (CPI): The concept of CPI, herein proposed, is different from the concept where cyber-physical systems are integrated with intelligent systems [23]. In the present taxonomy, the CPI attacks comprehend actions that are performed in the control loop of an NCS in order to gather information about the system's operation and/or its design. These attacks are intended to acquire the intelligence necessary to plan covert and controlled attacks, or even to subsidize data for replay attacks [7].

In Fig. 2, six kinds of DoS attacks are presented, with their respective requirements. From these six DoS attacks, the less complex are the three arbitrary ones.

1) DoS-arbitrary jitter: In this kind of attack, the delay of the forward and/or the feedback stream is arbitrarily changed, without a previous knowledge about the models of the NCS, in order to lead the system to an instability or to a condition that causes the interruption of the physical process. This attack only requires access to the control loop, once it may be performed by just consuming the resources of the system, such as the bandwidth of communication links, or the computational resources of the equipments that are in the control loop.
2) DoS-arbitrary data loss: In this kind of attack, the attacker prevents the data from reaching the actuator and/or the controller of the NCS. The communication channel is jammed arbitrarily, without a previous knowledge about the models of the NCS, leading the system to an instability or to a condition that causes the interruption

of the physical process. It is worth mentioning that some DoS-arbitrary jitter attack may result in a DoS-arbitrary data loss attack, if an eventual higher delay cause packet drop outs. As the DoS-arbitrary jitter attack, this attack only requires access to the control loop of the NCS.

3) DoS-arbitrary data injection: In such attacks, the attacker sends arbitrary false data to the controller, as it was sent by the sensors, and/or to the actuators, as it was sent by the controller. The false data is injected into the NCS closed loop without a previous knowledge about the models of the NCS. This attack is more complex than the DoS-arbitrary jitter and the DoS-arbitrary data loss attacks, given that it requires access to the data that flows in the control loop of the NCS.

The attacks classified as DoS-controlled—DoS-controlled jitter, DoS-controlled data loss, and DoS-controlled data injection—as shown in Fig. 2 interfere in the control loop of an NCS by the same means that their respective DoS-arbitrary attacks. The difference between a DoS-controlled attack and a DoS-arbitrary attack is that, in the former, the interference caused by the attacker is precisely planned and executed, in order to achieve the exact desired behavior that leads the physical service to an interruption, in a more efficient way. Thus, to achieve such efficiency, a DoS-controlled attack require an accurate knowledge about the NCS models, i.e., the plant and the controller transfer functions, which have to be analyzed to plan the attack.

Regarding to the SD attacks, we must consider the three different kinds of attack shown in Fig. 2: SD-controlled jitter, SD-controlled data loss, and SD-controlled data injection. The difference between an SD-controlled attack and a DoS-controlled attack is that the former is not intended to interrupt the physical process in a short term. It aims to keep the process running with reduced efficiency, sometimes also targeting a gradual physical deterioration of the controlled devices. To succeed, the SD-controlled attacks need to be planned based on an accurate knowledge about the dynamics and the design of the NCS. Otherwise, the attack can eventually interrupt the physical process, due to unpredicted reasons, evolving to a DoS attack.

The system knowledge required to both DoS-controlled and SD-controlled attacks can be gathered through CPI attacks, as shown in Fig. 2. The first, and simpler, CPI attack is the eavesdropping attack [24], [25], which consists of simply capturing the data transmitted through the forward and feedback streams of the NCS. The second CPI attack, herein proposed, is the system identification attack, which aims to obtain information about the control function of the controller and the transfer function of the plant, by analyzing the signals that flow in the network between the controller and the plant. The CPI attacks by themselves do not impact on the NCS, but they are an useful tool to plan efficient and accurate DoS-controlled and SD-controlled attacks.

B. Cybernetic Versus Physical Covertness

The covertness of an attack regards to its capacity to not be perceived or detected. In the case of cyber-physical attacks on NCSs, the covertness must be simultaneously analyzed in two different domains: the cyber domain and the physical domain. In this sense, it is presented in this section the definition of what is a *cybernetically covert* attack and what is a *physically covert* attack.

1) *Cybernetically covert attacks:* are the attacks that have low probability to be detected by algorithms that monitor the software, communications, and data of the NCS, or by systems that monitor the dynamics of the plant.

2) *Physically covert attacks:* are attacks that cause physical effects that cannot be easily noticed or identified by a human observer. The attack slightly modifies some behaviors of the system in a way that it physically affects the plant, but the effect is not easily perceptible or it can eventually be understood as a consequence of another root cause, other than an attack.

The taxonomy available in the literature does not clearly distinguish that an attack may have different degrees of covertness regarding to the cybernetic and physical domains. However, analyzing the cyber-physical attacks, it is possible to state that the measures taken to make an attack cybernetically covert do not necessarily guarantee a physically covert behavior, and vice versa. Thus, in order to provide a clear comprehension about these two aspects of the covertness of a cyber-physical attack, the two aforementioned classifications for covertness are introduced in this paper.

For instance, in [9] and [13], an attack architecture is proposed, where the attacker eliminates from the feedback signal the interference caused by him on a plant through data injection. That architecture hinders the system's ability to detect the attack through signal analysis, making the attack cybernetically covert. However, such architecture does not guarantee that the physical effects of the attack will not facilitate its disclosure. Indeed, depending on the plant's behavior, the attack can provide physical evidences that it is being manipulated, drawing the attention for the possibility of a cyber-physical attack. Thus, to be physically covert, the attacker's control function proposed in [9] and [13] has to be adjusted to meet the requirements of a physically covert attack, as herein defined, independently of the cybernetic covertness provided by the attack architecture.

IV. System Identification Attack

The system identification attack, herein proposed, is intended to assess the coefficients of the plant's transfer function $G(z)$ and the controller's control function $C(z)$. Both functions are LTI. The attack uses the BSA metaheuristic, proposed in [8] and briefly described in [26], to minimize the fitness function presented in this section.

The BSA is an evolutionary algorithm that uses the information obtained by past generations—or iterations—to perform the search for solutions for optimization problems. The algorithm has two parameters that are empirically adjusted: the size of its population P and η, described in [26], that establishes the amplitude of the movements of the individuals of P. The parameter η must be adjusted aiming to assign to the algorithm both good exploration and exploitation capabilities.

If the input $i(k)$ and the output $o(k)$ of a device of the NCS is known, the model of such device can be assessed by applying the known $i(k)$ in an estimated model, which must be adjusted until its estimated output $\hat{o}(k)$ converges to $o(k)$. In this sense, the BSA is used to iteratively adjust the estimated model, by minimizing a specific fitness function, until the estimated model converge to the actual model of the real device, that can be a controller or a plant of the NCS.

To establish the fitness function, first, it must be considered a generic LTI system, whose transfer function $Q(z)$ is represented by the following equation:

$$Q(z) = \frac{O(z)}{I(z)} = \frac{a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z^1 + a_0}{z^m + b_{m-1} z^{m-1} + \cdots + b_1 z^1 + b_0} \quad (1)$$

where $I(z)$ is the input of the system, $O(z)$ is the output of the system, n and m are the order of the numeratorand the denominator, respectively, and $[a_n, a_{n-1}, \ldots a_1, a_0]$ and $[b_{m-1}, b_{m-2}, \ldots b_1, b_0]$ are the coefficients of the numerator and the denominator, respectively, that are intended to be found by this system identification attack. Also, it must be considered that $i(k)$ and $o(k)$ represent the sampled input and output of the system, respectively, where $I(z) = \mathcal{Z}[i(k)]$, $O(z) = \mathcal{Z}[o(k)]$, k is the number of the sample, and $\mathcal{Z}$ represents the Z-transform operation.

In this system identification attack, $i(k)$ and $o(k)$ are first captured by an eavesdropping attack [24], [25], for example, during a monitoring period T. To deal with the eventual loss of samples, that may not be received by the attacker during T, the algorithm holds the value of the last received sample, according to (2), wherein $x(k)$ can either be $i(k)$ or $o(k)$

$$x(k) = \begin{cases} x(k-1), & \text{if the sample } k \text{ is lost} \\ x(k), & \text{otherwise.} \end{cases} \quad (2)$$

Then, after acquiring $i(k)$ and $o(k)$, the captured $i(k)$ is applied to the input of an estimated model, that is described by a transfer function whose coefficients $[a_{n,j}, a_{n-1,j}, \ldots a_{1,j}, a_{0,j}, b_{m-1,j}, b_{m-2,j}, \ldots b_{1,j}, b_{0,j}]$ are the coordinates of an individual j of the BSA. The application of $i(k)$ to the input of the estimated model results in an output signal $\hat{o}_j(k)$. After obtaining $\hat{o}_j(k)$, the fitness f_j of the individual j is computed comparing the output $o(k)$, captured from the attacked device, with the output $\hat{o}_j(k)$ of the estimated model, according to the following equation:

$$f_j = \frac{\sum_{k=0}^{N} (o(k) - \hat{o}_j(k))^2}{N} \quad (3)$$

where N is the number of samples that exist during the monitoring period T. Note that, if the attacker does not lose any sample of $i(k)$ and $o(k)$ during T, then $\min f_j = 0$ when $[a_{n,j}, a_{n-1,j}, \ldots a_{1,j}, a_{0,j}, b_{m-1,j}, b_{m-2,j}, \ldots .. b_{1,j}, b_{0,j}] = [a_n, a_{n-1}, \ldots a_1, a_0, b_{m-1}, b_{m-2}, \ldots b_1, b_0]$, i.e., when the estimated model converges to the actual model of the attacked device.

It is possible to establish an analogy between this system identification attack and the known plaintext cryptanalytic attack [27], where $i(k)$ and $o(k)$ correspond to the plaintext and ciphertext, respectively, the form of the generic transfer function $Q(z)$ corresponds to the encryption algorithm, and the actual coefficients of $Q(z)$ correspond to the secret key.

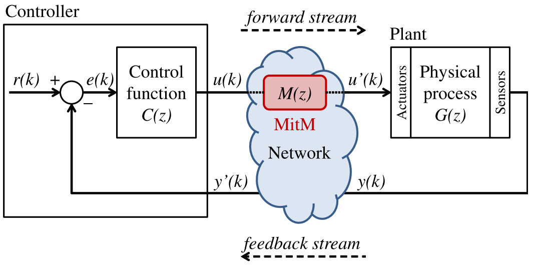


Fig. 3. MitM attack.

V. Covert Attack for SD

Based on the taxonomy presented in Section III-A, the attack described in this section is classified as an SD-controlled data injection attack. Its purpose is to reduce the MTBF of the plant and/or to reduce the efficiency of the physical process that the plant performs, by inserting false data into the control loop. At the same time, the attacker desires that the attack meets the requirement of being physically covert, as the definition presented in Section III-B.

One way to degrade a physical service is through the induction of an overshoot during the transient response of a plant. The overshoots, or peaks occurred when the system exceeds the targeted value during the transient response, can cause stress and possibly damage physical systems, such as mechanical, chemical, and electromechanical systems [28], [29]. Additionally, once they occur in a short period of time, the overshoots are difficult to be noticed by a human observer. Another way to degrade the service of a plant is causing a constant steady-state error on it, i.e., producing a constant error when $t \to \infty$. A low proportion steady-state error, besides being difficult to be perceived by a human observer, may reduce the efficiency of the physical process or, occasionally, stress and damage the system in mid/long term.

In the present attack, to achieve either of the two mentioned effects, i.e., an overshoot or a constant steady-state error, the attacker interfere in the NCS's communication process by injecting false data into the system in a controlled way. To do so, the attacker act as a MitM that executes an attack function $M(z)$, as presented in Fig. 3, where $U'(z) = M(z)U(z)$, $U(z) = \mathcal{Z}[u(k)]$, and $U'(z) = \mathcal{Z}[u'(k)]$. The function $M(z)$ is designed based on the models of the plant and the controller, both obtained through the system identification attack, described in Section IV. The effectiveness of the attack, therefore, depends on the design of $M(z)$, which, in turn, depends on the accuracy of the system identification attack. It is worth mentioning that, in Fig. 3, although the MitM is placed in the forward stream, it is possible to perform an attack by interfering in the feedback stream of the NCS. The MitM may act in wired or wireless networks, such as in [30].

VI. Results

In this section, the results obtained through simulations that combine the system identification attack with physically covert SD-controlled attacks are presented. First, in Section VI-A, the model of the attacked system is described. Then, in Section VI-B, the results obtained by the system identification attack are presented. After that, in Section VI-C, the results achieved by the simulations of physically covert SD-controlled data injection attacks, planned based on the data gathered by the system identification attack, are presented. Finally, in Section VI-D, possible countermeasures for the complete attack, in face of each requirement established in Fig. 2, are discussed.

A. Model of the Attacked System

The attacked NCS has the same architecture as of the NCS shown in Fig. 1, and consists of a proportional-integral (PI) controller that controls the rotational speed of a DC motor. The PI control function $C(z)$ and the DC motor transfer function $G(z)$ are the same as in [6]. The equations of this NCS are represented as follows:

$$C(z) = \frac{c_1 z - c_2}{z - 1} \qquad G(z) = \frac{g_1 z + g_2}{z^2 - g_3 z + g_4} \tag{4}$$

where $c_1 = 0,1701$, $c_2 = -0,1673$, $g_1 = 0,3379$, $g_2 = 0,2793$, $g_3 = -1,5462$, and $g_4 = 0,5646$. The sample rate of the system is 50 samples/s and the set point $r(k)$ is an unitary step function. The network delay is not taken into account in the simulations of this paper.

B. Results of the System Identification Attack

In this section, the performance of the system identification attack is evaluated through a set of simulations performed in MATLAB. The SIMULINK tool is used to compute the output $\hat{o}_j$ of the estimated models, whose coefficients are the coordinates of an individual j of the BSA.

The structure of the equations represented in (4) are previously known by the attacker once that, as a premise, it is known that the target is an NCS that controls a DC motor using a PI controller. In these simulations, the goal of the system identification attack is to discover g_1, g_2, g_3, g_4, c_1, and c_2, also taking into account scenarios in which the attacker occasionally loses samples of the forward and feedback streams.

Each time that the DC motor is turned ON, the forward and the feedback streams are captured by the attacker during a period $T = 2$ s. All initial conditions are considered 0, by the time that the motor is turned ON. The coefficients of $G(z)$, $[g_1, g_2, g_3, g_4]$, and the coefficients of $C(z)$, $[c_1, c_2]$, are computed separately considering that, albeit the closed loop, $G(z)$ and $C(z)$ are independent transfer functions. To assess $[g_1, g_2, g_3, g_4]$, the attacker considers the forward stream as the input and the feedback stream as the output of the estimated plant. In the opposite way, to assess $[c_1, c_2]$, the attacker considers the feedback stream as the input and the forward stream as the output of the estimated controller.

To simulate the loss of samples, four different rates l of sample loss: 0, 0.05, 0.1 and 0.2 are considered. Thus, a sample is lost by the attacker every time that $l < \mathcal{P}$, where $\mathcal{P} \sim U(0, 1)$ and U is the uniform distribution. A total of 100 different simulations for each rate of sample loss are executed.

In the BSA, the population is set to 100 individuals and η, empirically adjusted, is 1. To assess the coefficients of the controller $[c_1, c_2]$, the algorithm is executed for 600 iterations. To assess the coefficients of the plant $[g_1, g_2, g_3, g_4]$, the number of iterations is increased to 800, due to the higher number of dimensions of the search space in this case. The limits of each dimension of the search space are $[-10, 10]$.

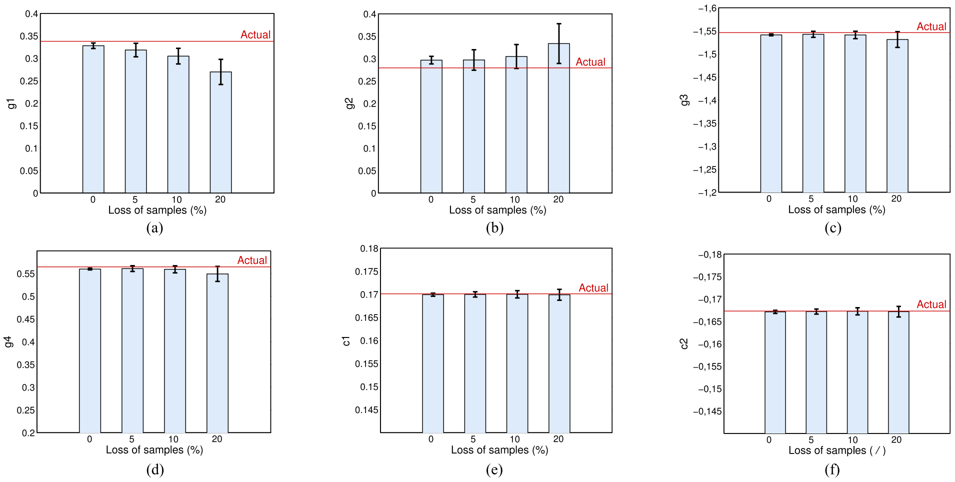
Fig. 4 shows the means of 100 estimated values of g_1, g_2, g_3, g_4, c_1, and c_2, with a confidence interval (CI) of 95%, considering different rates of sample loss. The actual values of the coefficients of $C(z)$ and $G(z)$ are also depicted in Fig. 4. Note that the scales of Fig. 4(a)–(d) are different from the scales of Fig. 4(e) and (f), due to the smaller amplitude of the CI of c_1 and c_2. In addition, some statistics of the obtained results are presented in Table II.

According to Table II the distributions of g_1, g_2, g_3, and g_4 have a high skewness, while the distributions of c_1 and c_2 have a moderate skewness. Table II also provides the kurtosis of all coefficients of $G(z)$ and $C(z)$. The kurtosis, computed in accordance with [31], is a statistical information used to evaluate whether the distribution is tall and thin (leptokurtic) or flat (platykurtic) when compared with the normal distribution. Based on the criteria defined in [31], the distributions of all coefficients of $G(z)$ and $C(z)$ are leptokurtic, which means that these distributions have more results closer to the mean than the normal distribution. However, analyzing Table II, it is not possible to state a clear general pattern of an increasing/decreasing skewness or kurtosis, in face of the growth of sample loss.

In Fig. 4, it is possible to verify that, in all cases, the CIs tend to grow with the increase of the sample loss. The same thing occurs with the standard deviations shown in Table II. Regarding to the coefficients of $G(z)$, Fig. 4 shows that the difference between the mean and the actual value of g_1, g_2, g_3, and g_4 also tends to raise with the increase of sample loss. It is worth mentioning that the performance of the algorithm when computing g_3 and g_4 is better than when computing g_1 and g_2, regarding the means and their CIs. This behavior results from the higher sensitivity that the output of $G(z)$ has to the variation of its poles than to the variations of its zeros. It means that, in this problem, f_j grows faster for errors in g_3 and g_4 than for errors in g_1 and g_2, making the BSA population converge more accurately in dimensions g_3 and g_4.

In Fig. 4, it is also possible to note that the accuracy obtained with the coefficients of $C(z)$ is better than the accuracy of the coefficients of $G(z)$, for all rates of sample loss. The means of c_1 and c_2 are closer to their actual values, with a smaller CI. In fact, the optimization process is more effective when computing the coefficients of $C(z)$ due to its smaller search space, that has only two dimensions instead of the four dimensions of the $G(z)$ problem.

As an additional metric to evaluate the performance of the algorithm, it is computed $|E_g| = |\mathcal{G}_r - \mathcal{G}_e|$ and $|E_c| = |\mathcal{C}_r - \mathcal{C}_e|$, that synthesize the error of the estimated coefficients of $G(z)$ and $C(z)$, respectively, for each solution found. $\mathcal{G}_r$ and



(a) (b) (c)

(d) (e) (f)

Fig. 4. Mean, with a CI of 95%, of the estimated coefficients of $G(z)$ and $C(z)$, in face of different rates of sample loss. (a) g_1 of $G(z)$, (b) g_2 of $G(z)$, (c) g_3 of $G(z)$, (d) g_4 of $G(z)$, (e) c_1 of $C(z)$, and (f) c_2 of $C(z)$.

TABLE II
STATISTICS OF THE RESULTS OBTAINED WITH DIFFERENT RATES OF SAMPLE LOSS

Loss of samples	Mean						Standard deviation					
	g_1	g_2	g_3	g_4	c_1	c_2	g_1	g_2	g_3	g_4	c_1	c_2
0%	0.32793	0.29652	−1.54121	0.55983	0.16991	−0.16712	0.03097	0.04288	0.00986	0.00944	0.00167	0.00178
5%	0.31835	0.29689	−1.54251	0.56085	0.16997	−0.16719	0.07572	0.11523	0.03322	0.03194	0.00287	0.00287
10%	0.30473	0.30461	−1.54110	0.55925	0.16999	−0.16724	0.08781	0.13483	0.04076	0.03922	0.00397	0.00399
20%	0.26963	0.33352	−1.53119	0.54916	0.16989	−0.16716	0.14120	0.22378	0.08596	0.08313	0.00596	0.00598
Loss of samples	Skewness(*)						Kurtosis(**)					
	g_1	g_2	g_3	g_4	c_1	c_2	g_1	g_2	g_3	g_4	c_1	c_2
0%	−1.21214	1.23278	1.75298	−1.73202	−0.64331	0.79458	0.18846	0.19433	0.21259	0.21218	0.15119	0.16472
5%	−2.34607	1.64875	1.35284	−1.41346	−0.42288	0.36037	0.08094	0.10527	0.09412	0.09802	0.02540	0.03118
10%	−2.52938	1.97711	1.18018	−1.26045	−0.23379	0.13377	0.16833	0.17123	0.25041	0.24811	0.24361	0.23429
20%	−3.24122	1.75186	1.68335	−1.71055	−0.40055	0.37927	0.21292	0.21127	0.25054	0.24932	0.23883	0.24441

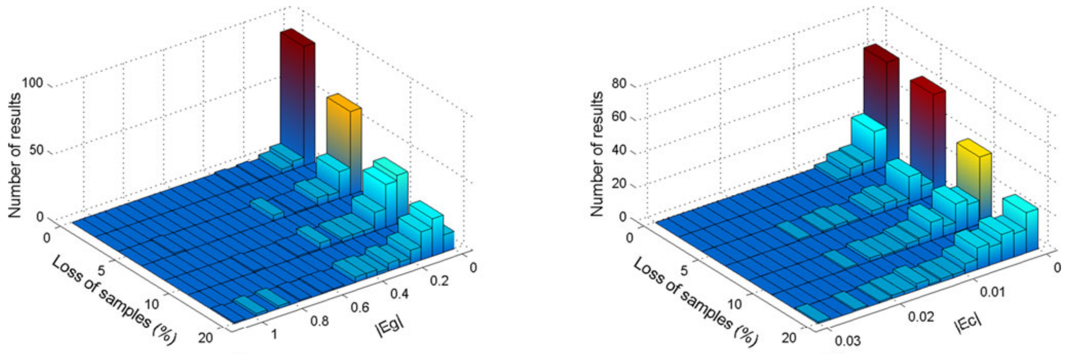
(*) Computed in accordance with Pearson's second coefficient of skewness. (**) Computed in accordance with [31].

$\mathcal{G}_e$ are vectors that contain the actual and the estimated coefficients of $G(z)$, respectively. Similarly, $\mathcal{C}_r$ and $\mathcal{C}_e$ are vectors that contain the actual and the estimated coefficients of $C(z)$, respectively. The histograms of $|E_g|$ and $|E_c|$ are presented in Fig. 5, considering the mentioned rates of sample loss. The histograms graphically show that $|E_g|$ and $|E_c|$, which correspond to the modulus of the error of the estimated coefficients of $G(z)$ and $C(z)$, respectively, tend to present higher values as the loss of samples grows. It can also be confirmed by the increase of the standard deviation of the coefficients of $G(z)$ and $C(z)$ presented in Table II. However, according to Fig. 5, the mode of this errors remains close to zero for all considered rates of sample loss.

C. Results of the SD Attacks

In this section, the results obtained through simulations of SD-controlled data injection attacks are presented, performed by a MitM acting in the control link of the NCS, as shown in Fig. 3. The attacks were simulated in MATLAB, aiming to evaluate their accuracy when planned based on the results provided in Section VI-B, obtained by the system identification attack. The two sets of attack were performed. The first one, aims to cause an *overshoot* of 50% in the rotational speed of the motor. The second one, aims to cause a stationary error of -10% in the rotational speed of the motor when it is on the steady state.

In the attack aiming the overshoot, the function executed by the attacker is $M(z) = \mathcal{K}_o$. Performing a root locus analysis considering the obtained models, the attacker adjusts $\mathcal{K}_o$ to make the system underdamped, with a peak of rotational speed 50% higher than its steady-state speed. The values of $\mathcal{K}_o$ are adjusted considering the average of the coefficients estimated in Section III-B. Table III shows the values of $\mathcal{K}_o$, estimated considering different rates of sample loss during the system identification attack, as well as the overshoots obtained with the respective $\mathcal{K}_o$ in the real model. In Fig. 6, it is possible to compare the



(a)

(b)

Fig. 5. Histograms of $|E_g|$ and $|E_c|$ in face of different rates of sample loss. (a) Distribution of $|E_g|$. (b) Distribution of $|E_c|$.

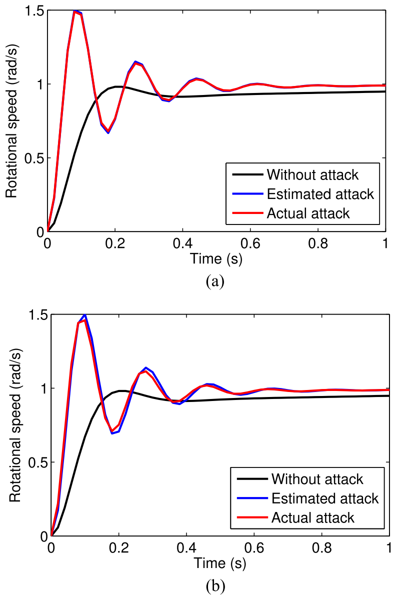
TABLE III
VALUES OF $\mathcal{K}_o$, $\mathcal{K}_{\mathrm{Ess}}$, AND THE RESULTS OBTAINED WITH THE ATTACKS

	Sample loss in the system identification attack			
	0%	5%	10%	20%
$\mathcal{K}_o$	4.0451	4.0745	4.0828	3.796
Overshoot in the real model	48.90%	49.43%	49.57%	45.94%
$\mathcal{K}_{\mathrm{Ess}}$	5.7471	5.7803	5.8140	5.8823
Stationary error in the real model	-10%	-10%	-9.9%	-9.8%

response of the system without attack with the response of the system with an attack aiming the overshoot of 50%. The curves referred to *estimated attack*, represent the results predicted by the attacker when applying the designed attack function $M(z)$ on the estimated model—i.e., the model discovered by the attacker through to the system identification attack. On the other hand, the curves referred to *actual attack* represent the response of the actual system in face of the same attack function $M(z)$. In another words, the curve *estimated attack* is the result achieved in a first moment, during the design stage of the attack, and the curve *actual attack* is the result obtained in a second moment, when the designed attack is launched over the actual system. It is noteworthy that the attack to the actual model—represented by the *actual attack* curve—presents, in the time domain, a response quite similar to the attack estimated with the model obtained by the system identification attack—represented by the *estimated attack* curve. This can be verified not only in the case where the system is identified with 0% of sample loss, but also in the worst considered case, i.e., with 20% of sample loss. It is worth mentioning that all responses presented in Fig. 6 converge to 1 rad/s.

In the attack where objective is to cause a stationary error of -10% on the rotational speed of the motor, the attacker executes the following equation:

$$M(z) = \frac{\mathcal{K}_{\mathrm{Ess}}(z-1)}{z-0.94} \tag{5}$$



(a)

(b)

Fig. 6. Response of the system to SD-controlled data injection attacks planned to cause an overshoot of 50% in the rotational speed of the motor. (a) Attack based on the data obtained without loss of samples. (b) Attack based on the data obtained with 20% of sample loss.

where $\mathcal{K}_{\mathrm{Ess}}$ is adjusted based on the data obtained with the system identification attack. The pole of $M(z)$ is added aiming to allow a stationary error in the system. The zero of $M(z)$ is intended to format the root locus in order to guarantee the existence of a stable $\mathcal{K}_{\mathrm{Ess}}$ that leads the system to a stationary error of -10%. Table III shows the $\mathcal{K}_{\mathrm{Ess}}$ resultant from different rates of sample loss during the system identification attack, as well as the stationary errors obtained with the respective $\mathcal{K}_{Ess}$ in the real model.

According to the data presented in Table III, it is possible to state that the SD-controlled data injection attack, designed based on the data gathered by the system identification attack, is capable to modify, in an accurate way, the response of the physical system, considering all the evaluated rates of sample loss. In the worst case, i.e., with 20% of sample loss, it is obtained an overshoot of 45.94% and a stationary error of -9.8%, quite close to the desired values of 50% and -10%, respectively. Such accuracy allows the attacker to keep his offensive under control, leading the system to a behavior that is predefined as physically covert and capable to degrade the service performed by the plant under attack.

These simulations provide conclusive data regarding to the effectiveness and potential impacts of the joint operation of system identification and SD-controlled data injection attacks on cyber-physical systems. However, the following issues, not explored in this paper, should be considered in the case of actual experiments or real attacks: the presence of noise, coming from the physical process, actuator, and sensors, as well as possible jitter on the network [32], which might influence both the system identification and SD-controlled data injection attacks; the delay unwittingly introduced by the MitM in the control loop during the SD-controlled data injection, which, depending on the magnitude, may influence the system dynamics; and last, but not least, the existing techniques/systems for communication security that must be overcome to allow the attacker get access to the NCS's control loop and data.

D. Discussion for Countermeasures

An NCS owner might think being safe from covert and accurate attacks, supposing that an eventual attacker does not know the plant's design and, thus, its models. Notwithstanding, this work demonstrates how a physically covert, and accurate, attack may be built starting from few information about the NCS—here, the only starting information is the structure of the transfer functions of both the plant and controller. Thus, the security of the system must not be relaxed, and countermeasures have to be adopted.

As shown in Fig. 2 the complete attack, herein proposed, is composed of a sequence of three individual attacks—or stages—namely eavesdropping, system identification, and SD-controlled data injection. Note that the requirements specified in Fig. 2 help on the development of layered defense strategies [33] for the proposed attack, where both information technology and operational technology countermeasures may be involved. Thus, a set of preventive countermeasures can be systematically thought based on the requirements drawn in Fig. 2.

1) The first, and straightforward preventive countermeasure, is to increase the difficulties for an attacker to have access to the control loop which, according to Fig. 2, may prevent the execution of the three mentioned stages of the attack. According to [34] the most effective architectural concept to protect an NCS is to segregate the control network from other networks. However, sometimes, it is not feasible or even wanted. Then, the possibility of an undesirable access to the control loop can be reduced by applying network segmentation, demilitarized zones, firewall policies, and using specific network architectures, such as established at the guidelines described in [34]. In the case of WNCSs, techniques are designed to minimize the transmitting power of the network devices [15] that should be used in order to reduce the probability of an attacker getting access to the control loop. Note that, minimizing the transmitting power of the WNCS's devices also minimizes the area from where the control loop can be accessed, which preventively reduce the probability to have the proposed attack launched on the WNCS.
2) In addition to the countermeasures aimed to prevent access to the control loop, other countermeasures are recommended to deny the access to the data that flows through the NCS, in case the former fails. In [10], a countermeasure that integrates a symmetric-key encryption algorithm, a hash algorithm, and a timestamp strategy is proposed to form a secure transmission mechanism between the controller side and the plant side, which is responsible for enforcing the data confidentiality and checking the data integrity and authenticity. The use of such countermeasure should hinder the access to the NCS data, which, according to Fig. 2, is required for the system identification attack and for the SD-controlled data injection attack.
3) Another way to avoid the attack herein proposed is preventing the attacker to obtain the required knowledge about the system. If the attacker eventually get access to the NCS's control loop and data, then it is necessary to make the system identification process harder and/or less accurate. Thus, the third preventive countermeasure lies on the use of control functions harder to be accurately identified, such as switching controllers [35], for instance. In this particular case, the accuracy/feasibility of the identification process may be influenced by the switching manner between the controller states [35], as well as by its dwell time [36]—i.e., the time between two consecutive switches. The pros and cons of this kind of countermeasure still need to be investigated. However, the present work suggests that the level of difficulty and accuracy to identify a control function should be taken into account during the design of the NCS.

Practical experimental results [10], [12], [37], [38] related to the cyber-security of cyber-physical systems evince the feasibility of launching actual attacks on such systems, as well as demonstrate the efforts to propose effective countermeasures for them. In [12] and [37], field-operational test attacks, performed at the Gignac canal system—in Southern France—where the attacker pilfers water from the canal, without being noticed, by manipulating the data transmitted by a sensor are reported. The authors indicate that, among all sensors of the attacked canal, there is a set of sensors that are more critical and should receive more investments on cyber-security mechanisms aiming more resilience to tampering. Examples of such cyber-security mechanisms are experimentally assessed in [10], where Pang and Liu propose a recursive networked predictive control technique, combined with a symmetric-key encryption algorithm, a timestamp, and a hash algorithm. The authors demonstrate, using a real Internet-based control system controlling a DC motor, that the solution is capable to make the NCS immune to attacks where 20% of the data is affected, and still effective if this percentage is raised to 80%. In [38], the use of cyber-security mechanisms in devices endowed with limited computational resources—such as the actuators and sensors of an NCS—is quantitatively evaluated through experiments using the communication module TS7250 (200-MHz ARM9 CPU and 32-MB SD-RAM). The results given in [38] indicate that a DES-CBC encryption requires 183.81 ms of processing time, while a RSA encryption requires 228.18 ms to encrypt the same amount of data from a solid-state transformer, using a 1024-bit key. If a 2048-bit key is used, the processing time of the RSA, for example, grows to 1457.14 ms. This may be an issue if it is considered an NCS sensitive to delay. Such processing times exemplify the tradeoff between security and performance, possibly faced when dealing with NCSs, which must be taken into account while deciding for a countermeasure.

VII. Conclusion

This work proposes a physically covert attack for SD, in which the performance depends on the knowledge about the model of the plant under attack and its controller. To obtain such knowledge, a system identification attack, based on the BSA algorithm, is proposed. The effectiveness of the system identification attack is demonstrated and its performance is statistically analyzed in face of different rates of sample loss. The results achieved by the physically covert attacks for SD, designed based on the data gathered by the system identification attack, demonstrate the high degree of accuracy that may be achieved with the joint operation of the two attacks. In the worst case, i.e., with 20% of sample loss during the system identification attack, the attacker attained an overshoot of 45.94% and a stationary error of -9.8%, quite close to the desired values of 50% and -10%, respectively. In both physically covert interventions, the accuracy of the attacks ensures that they will not evolve to unwanted behaviors, physically perceivable.

As future work, the research of techniques capable to avoid, or complicate, physically convert attacks planned with the data obtained by system identification attacks is encouraged. In this sense, we plan to further investigate countermeasures capable to make it difficult to obtain information about cyber-physical control systems, which is essential for planning covert and controlled attacks.

References

[1] Y. Tipsuwan, M.-Y. Chow, and R. Vanijjirattikhan, "An implementation of a networked PI controller over IP network," in *Proc. IEEE 29th Annu. Conf. Ind. Electron. Soc.*, 2003, vol. 3, pp. 2805–2810.

[2] R. A. Gupta and M.-Y. Chow, "Networked control system: Overview and research trends," *IEEE Trans. Ind. Electron.*, vol. 57, no. 7, pp. 2527–2535, Jul. 2010.

[3] L. Zhang, L. Xie, W. Li, and Z. Wang, "Security solutions for networked control systems based on des algorithm and improved grey prediction model," *Int. J. Comput. Netw. Inf. Security*, vol. 6, no. 1, pp. 78–85, 2013.

[4] A. A. Farooqui, S. S. H. Zaidi, A. Y. Memon, and S. Qazi, "Cyber security backdrop: A SCADA testbed," in *Proc. IEEE Comput., Commun. IT Appl. Conf.*, 2014, pp. 98–103.

[5] M.-Y. Chow and Y. Tipsuwan, "Network-based control systems: A tutorial," in *Proc. 27th Annu. Conf. IEEE Ind. Electron. Soc.*, 2001, vol. 3, pp. 1593–1602.

[6] M. Long, C.-H. Wu, and J. Y. Hung, "Denial of service attacks on network-based control systems: Impact and mitigation," *IEEE Trans. Ind. Informat.*, vol. 1, no. 2, pp. 85–96, May 2005.

[7] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security Privacy*, vol. 9, no. 3, pp. 49–51, May/Jun. 2011.

[8] P. Civicioglu, "Backtracking search optimization algorithm for numerical optimization problems," *Appl. Math. Comput.*, vol. 219, no. 15, pp. 8121–8144, 2013.

[9] R. Smith, "A decoupled feedback structure for covertly appropriating networked control systems," in *Proc. 18th IFAC World Congr.*, 2011, vol. 18, no. 1, pp. 90–95.

[10] Z.-H. Pang and G.-P. Liu, "Design and implementation of secure networked predictive control systems under deception attacks," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 5, pp. 1334–1342, Sep. 2012.

[11] S. Dasgupta *et al.*, "Networked control of a large pressurized heavy water reactor (PHWR) with discrete proportional-integral-derivative (PID) controllers," *IEEE Trans. Nucl. Sci.*, vol. 60, no. 5, pp. 3879–3888, Oct. 2013.

[12] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen, "Cyber security of water SCADA systems—Part I: Analysis and experimentation of stealthy deception attacks," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 5, pp. 1963–1970, Sep. 2013.

[13] R. S. Smith, "Covert misappropriation of networked control systems: Presenting a feedback structure," *IEEE Control Systems*, vol. 35, no. 1, pp. 82–92, Feb. 2015.

[14] P. Ferrari, A. Flammini, M. Rizzi, and E. Sisinni, "Improving simulation of wireless networked control systems based on wirelesshart," *Comput. Stand. Interfaces*, vol. 35, no. 6, pp. 605–615, 2013.

[15] Y. Sadi, S. C. Ergen, and P. Park, "Minimum energy data transmission for wireless networked control systems," *IEEE Trans. Wireless Commun.*, vol. 13, no. 4, pp. 2163–2175, Apr. 2014.

[16] M. Das *et al.*, "Network control system applied to a large pressurized heavy water reactor," *IEEE Trans. Nucl. Sci.*, vol. 53, no. 5, pp. 2948–2956, Oct. 2006.

[17] M. Uma and G. Padmavathi, "A survey on various cyber attacks and their classification," *Int. J. Netw. Security*, vol. 15, no. 5, pp. 390–396, 2013.

[18] Z. Drias, A. Serhrouchni, and O. Vogel, "Taxonomy of attacks on industrial control protocols," in *Proc. Int. Conf. Protocol Eng., Int. Conf. New Technol. Distrib. Syst.*, 2015, pp. 1–6.

[19] N. Falliere, L. O. Murchu, and E. Chien, "W32. stuxnet dossier," White Paper, Symantec Corp., Mountain View, CA, USA, Secur. Response, vol. 5, p. 6, 2011.

[20] A. C. Snoeren *et al.*, "Single-packet ip traceback," *IEEE/ACM Trans. Netw.*, vol. 10, no. 6, pp. 721–734, Dec. 2002.

[21] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "A secure control framework for resource-limited adversaries," *Automatica*, vol. 51, pp. 135–148, 2015.

[22] A. Hussain, J. Heidemann, and C. Papadopoulos, "A framework for classifying denial of service attacks," in *Proc. Conf. Appl., Technol., Archit. Protocols Comput. Commun.*, 2003, pp. 99–110.

[23] C. Ramos, Z. Vale, and L. Faria, "Cyber-physical intelligence in the context of power systems," in *Future Generation Information Technology*. Berlin, Germany: Springer, 2011, pp. 19–29.

[24] S. Khatri, P. Sharma, P. Chaudhary, and A. Bijalwan, "A taxonomy of physical layer attacks in MANET," *Int. J. Comput. Appl.*, vol. 117, no. 22, pp. 6–11, 2015.

[25] Y. Zou and G. Wang, "Intercept behavior analysis of industrial wireless sensor networks in the presence of eavesdropping attack," *IEEE Trans. Ind. Informat.*, vol. 12, no. 2, pp. 780–787, Apr. 2016.

[26] A. O. de Sá, N. Nedjah, and L. de Macedo Mourelle, "Distributed efficient localization in swarm robotic systems using swarm intelligence algorithms," *Neurocomputing*, vol. 172, pp. 322–336, 2016.

[27] W. Stallings, *Cryptography and Network Security: Principles and Practices*. Upper Saddle River, NJ, USA: Pearson, 2006.

[28] M. El-Sharkawi and C. Huang, "Variable structure tracking of DC motor for high performance applications," *IEEE Trans. Energy Convers.*, vol. 4, no. 4, pp. 643–650, Dec. 1989.

[29] T. Tran, Q. P. Ha, and H. T. Nguyen, "Robust non-overshoot time responses using cascade sliding mode-PID control," *J. Adv. Comput. Intell. Intell. Informat.*, vol. 10, pp. 1224–1231, 2007.

[30] H. Hwang, G. Jung, K. Sohn, and S. Park, "A study on MITM (man in the middle) vulnerability in wireless network using 802.1 x and eap," in *Proc. IEEE Int. Conf. Inf. Sci. Security*, 2008, pp. 164–170.

[31] L. Sachs, *Applied Statistics: A Handbook of Techniques*. Berlin, Germany: Springer, 2012.

[32] L. Zhang, H. Gao, and O. Kaynak, "Network-induced constraints in networked control systems: A survey," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 403–416, Feb. 2013.

[33] A. Hahn, "Operational technology and information technology in industrial control systems," in *Cyber-Security of SCADA and Other Industrial Control Systems*. Berlin, Germany: Springer, 2016, pp. 51–68.

[34] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn, "Nist special publication 800-82, revision 2: Guide to industrial control systems (ICS) security," National Inst. Stand. Technol., Gaithersburg, MD, USA, 2015.

[35] C. Zhang, Y. Fan, and Y. Hao, "Informative property of the data set in a single-input single-output (SISO) closed-loop system with a switching controller," *Chin. J. Chem. Eng.*, vol. 20, no. 6, pp. 1128–1135, 2012.

[36] M. Baştuğ, "Recursive modeling of switched linear systems: A behavioral approach," Master's thesis, Istanbul Technical University, Istanbul, Turkey, 2012.

[37] S. Amin, X. Litrico, S. S. Sastry, and A. M. Bayen, "Cyber security of water SCADA systems—Part II: Attack detection using enhanced hydrodynamic models," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 5, pp. 1679–1693, Sep. 2013.

[38] W. Wang and Z. Lu, "Cyber security in the smart grid: Survey and challenges," *Comput. Netw.*, vol. 57, no. 5, pp. 1344–1371, 2013.

Alan Oliveira de Sá graduated in electronic engineering from Rio de Janeiro Federal Center for Technological Education, Rio de Janeiro, Brazil, in 2006, and received the M.Sc. degree in electronic engineering from the State University of Rio de Janeiro, Rio de Janeiro, in 2015. He is currently working toward the Ph.D. degree in Cyber Security for Control and Automation Systems at the Federal University of Rio de Janeiro.

He is currently a Professor with the Brazilian Navy. His research interests include cybersecurity, control systems, and intelligent systems.

Luiz F. Rust da Costa Carmo received the Ph.D. degree in computer science from the LAAS/CNRS, Toulouse III, France, in 1994.

He is currently a Senior Specialist of computer sciences at the Brazilian Institute of Metrology and Quality (INMETRO), Rio de Janeiro, Brazil, responsible for developing new information security assessment programs. He is an active Lecturer of the doctoral program in computer sciences of Federal University of Rio de Janeiro, Rio de Janeiro, and the Head of the master of sciences program in metrology of INMETRO. His research interests include information security and embedded systems.

Raphael C. S. Machado received the Ph.D. degree in computers and systems engineering from the Federal University of Rio de Janeiro, Rio de Janeiro, Brazil, in 2010.

He is currently a Researcher in the National Institute of Metrology, Quality and Technology, Rio de Janeiro, Brazil. He is also a Professor of the graduate program in computer science at the Rio de Janeiro Federal Center for Technological Education, Rio de Janeiro. He has several publications in international journals and conferences, in themes, including cryptography, software analysis and protection, combinatorics, algorithms and applications to smart grids, computational geometry, genomics, and metrology.

APPENDIX B

Mobile Netw Appl
https://doi.org/10.1007/s11036-017-0943-5

Bio-inspired Active System Identification: a Cyber-Physical Intelligence Attack in Networked Control Systems

Alan Oliveira de Sá[1,2] · Luiz F. R. da C. Carmo[2,3] · Raphael C. S. Machado[3,4]

Abstract From the point of view of the control theory, the literature indicates that stealthy and accurate cyber-physical attacks on Networked Control System (NCS) must be planned based on an accurate knowledge about the model of the attacked system. However, most literature about these attacks does not indicate how such knowledge is obtained by the attacker. So, to fill this hiatus, an Active System Identification attack is proposed in this paper, where the attacker injects data on the NCS to learn about its model. The attack is implemented based on two bio-inspired metaheuristics: Backtracking Search Optimization Algorithm (BSA) and Particle Swarm Optimization (PSO). To improve the accuracy of the estimated models, a statistical refinement is proposed for the outcomes of the two optimization algorithms. Additionally, a set of data injection attacks are shown in order to demonstrate the capability of the proposed attack in supporting the design of other sophisticated attacks. The results indicate a better performance of the BSA-based attacks, especially when the captured signals contain white Gaussian noise. The goal of this paper is to demonstrate the degree of accuracy that this System Identification attack may achieve, highlighting the potential impacts and encouraging the research of possible countermeasures.

Keywords Security · Cyber-physical systems · Networked control systems · System identification · Backtracking search algorithm · Particle swarm optimization

1 Introduction

System identification, i.e. the action of building mathematical models of dynamic systems, is often used to obtain the model of physical processes aiming to support the design of their respective control systems. However, it can also be considered a key step for the execution of accurate and stealth – or covert, as mentioned in [7, 20, 21, 24] – attacks against Networked Control Systems (NCS). Indeed, to reduce the probability to be detected by algorithms that monitor the dynamics of the controlled plant, the attacker must have an accurate model of the targeted system, such as demonstrated in [21, 24].

A possible strategy to obtain information about the model of the targeted system is through passive System Identification attacks, as reported in [7]. In that technique, the attacker eavesdrops the communications between the controller, actuators and sensors of the NCS until enough information is collected to determine the parameters of the plant and its control system. Such passive approach can

This research was partially supported by the Brazilian research agencies CNPq and FAPERJ.

✉ Alan Oliveira de Sá
alan.oliveira.sa@gmail.com

Luiz F. R. da C. Carmo
lfrust@inmetro.gov.br

Raphael C. S. Machado
rcmachado@inmetro.gov.br

[1] Admiral Wandenkolk Instruction Center, Brazilian Navy, Rio de Janeiro, Brazil

[2] Institute of Mathematics/NCE, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil

[3] National Institute of Metrology, Quality and Technology, Rio de Janeiro, Brazil

[4] Rio de Janeiro Federal Center for Technological Education, Rio de Janeiro, Brazil

Published online: 05 October 2017

make the system identification more time consuming, until meaningful information transits at the eavesdropped communication links. The situation is even worse if the system is in steady state because no meaningful information may transit through the NCS's communication links for a long time. The information content of the signals measured under steady operating conditions is often insufficient for identification purposes [26]. This attacker's constraint may be overcome by the Active System Identification Attack herein proposed, which, as far as we know, is not reported in the literature.[1]

Our approach was inspired by the classic active cryptanalytic attacks (chosen plaintext and chosen cyphertext), where the attacker inserts messages in the crypto-engine to deduce the secret key. Note that this is the opposite of the passive attacks (cyphertext only and known plaintext), where the attacker simply listens the communication channels and passively collects information to recover the secret key [23].

In the attack presented in this work, a specially tailored signal is inserted by the attacker in an NCS communication channel. After that, by observing the behavior of the system in closed-loop, the attacker determines the parameters of its open-loop transfer function. To do so, the attacker only needs to intercept one communication channel of the NCS, where the attacker both inserts the attack signal and listens the consequent system response.

If an attack signal $a(k)$ and the consequent response $y_a(k)$ of an NCS are known, the open-loop transfer function can be assessed by applying $a(k)$ in an estimated model, which is adjusted until its estimated output $\hat{y}_a(k)$ matches $y_a(k)$. The Backtracking Search Optimization algorithm (BSA) [4] and the Particle Swarm Optimization (PSO) [12] are herein used to iteratively adjust the parameters of the estimated model, by minimizing a specific fitness function until the estimated model converges to the actual model of the NCS. The BSA and the PSO are chosen to perform this task due to their capability to converge to good solutions, such as demonstrated in [10, 11, 16, 27] specifically for control system problems. Given the stochastic nature of the used algorithms (BSA and PSO), the results need to be statistically analyzed in order to perform a refinement of the estimated model. In this work, the statistical refinement used in [6] is improved, leading to a more accurate estimated model than the results obtained in [6].

The knowledge of the NCS's open-loop transfer function obtained by the Active System Identification attack is useful for the design of other sophisticated attacks [7, 21]. To demonstrate this usefulness from the attacker point of view, this paper includes the simulation of a set of data injection attacks designed based on the data gathered by the Active System Identification attack. In these simulations, not presented in [6], the attacker accurately induces an overshoot on the attacked plant, which may cause stress and possible damages [8, 25], reducing its mean time between failure (MTBF).

It is worth mentioning that the Active System Identification attack herein proposed is different from the active attacks performed to identify vulnerabilities of protocols and applications within the layers of the OSI model, such as the active scanning process used to identify network services [2]. The attack herein proposed aims to identify the physical model of a plant that, in an NCS, lies above the application layer of the OSI model.

Note that the applications of NCSs can range from cooperative control of vehicles using mobile networks [15, 17] to large Pressurized Heavy Water Reactors (PHWR) [5] or water canal systems [1, 21] controlled by wired NCSs. This include a vast number of potential – sometimes critical – targets for the attack herein proposed. In this sense, the goal of this paper is to demonstrate the degree of accuracy that the present attack may achieve, highlighting its potential impacts and encouraging the research of countermeasures capable to prevent or detect its execution.

In summary, the main contributions of this paper, with regard to the preliminary version of this work [6], are:

- The review on the taxonomy presented in [7], in order to encompass the Active System Identification attack in the context of the Cyber-Physical Intelligence (CPI) attacks. Also, it sets the role that the proposed attack – as a CPI attack – plays in building model-dependent attacks.
- The proposal of a new statistical refinement method for the outcomes provided by the bio-inspired metaheuristics. The results demonstrate that this refinement improves the quality of the information produced by the identification attack.
- The novel joint operation of the Active System Identification Attack and a Controlled Data Injection Attack, which allows the evaluation on how a model-dependent attack can benefit from the intelligence obtained by the Active System Identification Attack. The results indicate that the referred model-dependent attack can achieve high accuracy when supported by the Active System Identification Attack, specially when the latter is statistically refined by the method introduced in this paper.

The remainder of this paper is organized as follows. In Section 2, we review the literature of NCS attacks, with focus on the intelligence gathered to support their design.

[1] A preliminary version of this work was presented in the 10th EAI International Conference on Bio-inspired Information and Communications Technologies (BICT 2017) and published in the proceedings of the event [6]. The present paper proposes a refinement for the system identification method described in [6] and simulates a data injection attack using the data obtained after this refinement.

In Section 3, we discuss and review the taxonomy presented in [7] in order to encompass the attack herein proposed. In Sections 4 and 5, we provide brief descriptions of the BSA and PSO, respectively. In Section 6, the Active System Identification attack, herein proposed, is described. Section 7 presents the results achieved by the proposed attack, comparing both metaheuristics in simulations where the NCS is constituted by a DC motor and a proportional-integral (PI) controller. Also, Section 7 quantitatively demonstrates the accuracy that a data injection attack may achieve, when supported by the proposed Active System Identification attack. Section 8 contains our final considerations.

2 Related works

The possibility of large impact cyber-physical attacks became unprecedentedly concrete after the launch of the Stuxnet worm [13] and has been motivating researches concerning the security of NCSs. In this section, a review of the literature related to this subject is presented.

In [14] the authors propose two queuing models that are used to evaluate the impact of delay jitter and packet loss in an NCS under attack. The attack is not designed taking into account the models of the controller and the physical plant. Such models are unknown by the attacker. Thus, to affect the plant's behavior, the attacker arbitrarily floods the network with traffic, causing jitter and packet loss. In this method of attack, the excess of packets in the network can reduce the stealthiness of the attack, allowing the adoption of countermeasures, such as packet filtering [14] or blocking the malicious traffic on its origin [22]. Moreover, the arbitrary intervention in a system which the models are unknown may lead the plant to an extreme physical behavior, which is not desired if a stealth attack is intended [7].

In [9], a testbed for Supervisory Control and Data Acquisition (SCADA) using TrueTime (a MATLAB/ Simulink based tool) is presented. The authors demonstrate an attack where a malicious agent transmits false signals to the controller and actuator of an NCS. The false signals are randomly generated, aiming to make a DC motor lose its stability. This kind of attack does not require a previous knowledge about the plant and controller of the NCS. The drawback is that the desired physical effect and the stealthiness of the attack cannot be ensured due to unpredictable consequences from the application of random false signals to a system which the model is not known.

A general framework for the analysis of a wide variety of attacks over NCSs is provided in [24]. The authors classify and establish the requirements for the attacks in terms of model knowledge, disclosure and disruption resources. In their work, it is stated that covert attacks require high level of knowledge about the model of the targeted system. Examples of covert attacks that agree with this statement are provided in [20, 21]. In these works, the attacks are performed by a man-in-the-middle (MitM), where the attacker needs to know the model of the plant under attack and also inject false data in both forward and feedback streams. The stealthiness of the attacks described in [20, 21] is analyzed from the perspective of the signals arriving to the controller and depends on the difference between the actual model of the plant and the model known by the attacker. In [1], another stealth attack is demonstrated. The attacker, aware of the system's model, injects an attack signal in the NCS to steal water from the Gignac canal system located in Southern France.

In [1, 20, 21, 24], where a previous knowledge about the models of the NCS under attack is required, it is not described how this knowledge is obtained by the attacker. It is just stated that a model is previously known to support the design of the attack. More recently, in [7], the authors propose a System Identification attack to fill this hiatus. They demonstrate how the data required for the design of Denial-of-Service (DoS) or Service Degradation (SD) attacks may be obtained through a passive System Identification attack. The attack proposed in [7] does not need to inject signals on the NCS to estimate its models. However, it depends on the occurrence of events, that are not controlled by the attacker, to produce signals that carry meaningful information for the system identification algorithm. The Active System Identification attack, herein proposed, constitutes an alternative to the passive System Identification attacks in situations where the attacker may not wait so long for the occurrence of such meaningful signals. A synthesis of the characteristics of the attacks referred in this section is presented in Table 1.

3 Taxonomy

In [7], the authors propose a taxonomy that encompasses three main classes of attack – Denial-of-Service (DoS), Service Degradation (SD), and Cyber-Physical Intelligence (CPI) – in which the service to be attacked/ protected is the work performed by the physical process controlled by an NCS. According to that taxonomy, the DoS attacks are intended to interrupt the execution of the work performed by the controlled plant, or even destroy the plant in a short term. On the other hand, the SD attacks aims to reduce the efficiency of the physical process, or even reduce the mean time between failure (MTBF) of the plant in mid/long term. Yet, according to that taxonomy, the CPI attacks are intended to gather information of the NCS basically through two kinds of attack – eavesdropping, and System Identification attacks –, in order to provide the information necessary for planning and designing DoS and SD controlled attacks. The referred taxonomy establishes the requirements for each attack of

Table 1 Synthesis of the related attacks

Attack	Method	System knowledge	How the knowledge is obtained
Stuxnet *worm* [13]	Modifications in the PLC code	Yes	Experiments in a real system
Long, et al. [14]	Inducing *jitter* and packet loss	None	N/A
Farooqui, et al. [9]	Data injection	None	N/A
Smith [20, 21]	Data injection	Yes	Not described
Teixeira [24]	Packet loss	None	N/A
	Data injection	Yes	Not described
Amin [1]	Data injection	Yes	Not described
SD-Controlled [7]	Data injection	Yes	Passive system identification

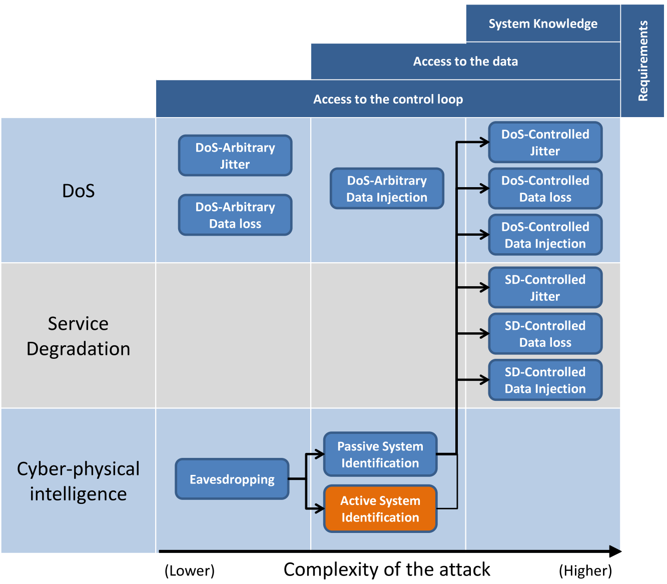
these three main classes and, above all, explains how model-dependent attacks, such as the DoS and SD controlled attacks can benefit from the information provided by CPI attacks.

The attacks belonging to the first two classes, i.e. DoS and SD, are premised active, once they act through the induction of jitter, data loss or data injection on the NCS. On the other hand, according to that taxonomy, the attacks belonging to the CPI class of attacks do not impact or interfere on the NCS, once they only need to listen the control signals that flow through the NCS. The eavesdropping attack simply capture the control signals that flow through the network. The System Identification attack, according to [7], collects the data that flows through the input and output of the NCS devices, i.e. controllers and plants, and uses the collected information to passively estimate the model of such devices.

However, the results achieved by the present work lead us to review the taxonomy proposed in [7], specifically with regard to the System Identification attacks. Different from the System Identification attack defined in [7], the attack proposed in this paper requires the injection of an attack signal in the NCS, in order to estimate its model through the analysis of its consequent response. Thus, it is necessary to expand the taxonomy related to System Identification attacks, that are now divided within two kinds, as shown in Fig. 1:

– The Passive System Identification attacks: this kind of attack estimates the model of an NCS based on the analysis of the signals collected from the input and output of the system's devices. This kind of attack analyzes signals that typically flow through the NCS, as a result of its normal operation. In this case, both input and

Fig. 1 Classification and requirements of the cyber-physical attacks that act in the control loop of an NCS



output signals must carry meaningful information – i.e. information enough to estimate the transfer function of the attacked system/device –, and it is not necessary to inject signals into the attacked system.

- The Active System Identification attack: in this kind of attack, aim of this work, the attacker injects a signal into the system and estimates its model based on the system's response in face of the attack signal. From the attacker point of view, this attack is useful, for example, when the system is in steady state and the attacker cannot wait for a signal carrying meaningful information for the identification process.

It is noteworthy that an Active System Identification attack is less stealthy than a Passive System Identification attack, given that the former needs to interfere in the system and the latter just needs to listen its signals. In this sense, when performing an Active System Identification attack, the attacker must choose signals that, when injected on the NCS, are more difficult to be perceived by a defense system. From the defender perspective, it is important to be aware of this kind of attack and also learn about the stealthiness of Active System Identification attacks, in order to develop techniques to identify and avoid them.

4 Backtracking search algorithm

In this section, the basic concepts of the BSA are described in order to provide a clear comprehension regarding to the parameters of the algorithm that are adjusted for the attack. The BSA is a bio-inspired metaheuristic that searches for solutions of optimization problems using the information obtained by past generations – or iterations. According to [4], its search process is metaphorically analogous to the behavior of a social group of animals that, at random intervals, returns to hunting areas previously visited for food foraging. The general evolutionary structure of the BSA is shown in Algorithm 1.

Algorithm 1 BSA

```
begin
    Initialization;
    repeat
        Selection-I;
        Generate new population
            Mutation;
            Crossover;
        end
        Selection-II;
    until Stopping Condition;
end
```

At the Initialization stage, the algorithm generates and evaluates the initial population $\mathcal{P}_0$ and sets the historical population $\mathcal{P}_{hist}$. The latter constitutes the BSA's memory that, in Selection-I stage, is updated with historical coordinates visited by the individuals.

During the first selection stage (Selection-I), the algorithm randomly determines, based on an uniform distribution U, whether the current population $\mathcal{P}$ should be kept as the new historical population and, therefore, replace $\mathcal{P}_{hist}$ (i.e. if $a < b|a, b \sim U(0, 1)$, then $P_{hist} = P$). Subsequently, at every iteration, it shuffles the individuals of $\mathcal{P}_{hist}$ – having $\mathcal{P}_{hist}$ been replaced or not.

The mutation operator creates $\mathcal{P}_{mod}$, which is the preliminary version of the new population $\mathcal{P}_{new}$). It does so according to Eq. 1:

$$\mathcal{P}_{mod} = \mathcal{P} + \eta \cdot \Gamma(\mathcal{P}_{hist} - \mathcal{P}), \qquad (1)$$

wherein η is empirically adjusted through simulations and $\Gamma \sim N(0, 1)$, with N being a normal standard distribution. Therefore, $\mathcal{P}_{mod}$ is the result of the movement of $\mathcal{P}$'s individuals in the directions established by vector $(\mathcal{P}_{hist} - \mathcal{P})$ and η controls the displacements' amplitude.

In order to create the final version of $\mathcal{P}_{new}$, the crossover operator randomly combines, also following a uniform distribution, individuals from $\mathcal{P}_{mod}$ and others from $\mathcal{P}$.

At the second selection stage (Selection-II), the algorithm firstly evaluates the individuals of $\mathcal{P}_{new}$ using a fitness function f. After that, individuals of $\mathcal{P}$ (i.e. individuals before applying the mutation and crossover operators) are replaced by individuals of $\mathcal{P}_{new}$ (i.e. individuals obtained after mutation and crossover) with better fitness. Therefore, $\mathcal{P}$ includes only new individuals that evolved. While the stopping condition has not yet been reached, the algorithm iterates. Otherwise, it returns the best solution found.

Note that the algorithm has two parameters that are empirically adjusted: the size $|\mathcal{P}|$ of its population $\mathcal{P}$; and η, that establishes the amplitude of the movements of the individuals of $\mathcal{P}$. The parameter η must be adjusted to assign good exploration and exploitation capabilities to the algorithm. With these parameters set, the BSA is used to search for the global minimum of the fitness function described in Section 6.

5 Particle swarm optimization

PSO has roots in the collective behavior of social models such as bird flocking and fish schooling. A particle, i.e. the basic element of the algorithm, represents a possible solution of a problem. Therefore, the swarm represents a set of possible solutions. At each iterative cycle, the position of each particle is updated according to Eq. 2, where x_j and v_j are the position and velocity of particle j, respectively.

$$x_j(t+1) = x_j(t) + v_j(t+1) \tag{2}$$

The computation of v_j considers three terms: the particle's inertia; the particle's cognition, which is based on the best solution found by the particle so far; and social term, which is based on global best solution found by the swarm. The velocity of particle j, at each dimension d, is defined in Eq. 3:

$$\begin{aligned} v_{jd}(t+1) &= \omega v_{jd}(t) + \varphi_1 r_{1d}(t)(m_{jd} - x_{jd}(t)) \\ &\quad + \varphi_2 r_{2d}(t)(m_{gd} - x_{jd}(t)), \end{aligned} \tag{3}$$

wherein ω is a parameter that weighs the inertia of the particle, φ_1 and φ_2 are parameters that weigh the cognitive and social terms, respectively, r_1 and r_2 are random numbers in [0,1], m_j is the best position visited by particle j so far, and m_g is the best position discovered by the swarm considering the experience of all the particles. To obtain m_j and m_g the algorithm evaluates, at each iteration, the position x_j of each particle j using a fitness function $f(x)$.

In order to better explore multi-dimensional search spaces, a velocity limit is imposed for each dimension d, as in Eq. 4:

$$0 \leq v_{jd} \leq \delta(max_d - min_d), \tag{4}$$

wherein max_d and min_d are the maximum and minimum limits of the search space at each dimension d and $\delta \in [0, 1]$. The overall computation that the PSO performs to minimize a fitness function $f(x)$ is given in Algorithm 2, where x is the particle position and $\mathcal{S}$ is the swarm size.

Algorithm 2 PSO algorithm

```
begin
    for each particle $j$, $1 \leq j \leq \mathcal{S}$ do
        Set randomly position $x_j$ and velocity $v_j$;
        $m_j \leftarrow x_j$;
    end
    $m_g \leftarrow smallest\ m_j$, $1 \leq j \leq \mathcal{S}$;
    repeat
        for each particle $j$, $1 \leq j \leq \mathcal{S}$ do
            Update velocity $v_j$, as in Eqs. 3 and 4;
            Update position $x_j$, as in Eq. 2;
            $fitness \leftarrow f(x_j)$;
            $m_k \leftarrow x_j$, whenever $fitness < f(m_j)$;
            $m_g \leftarrow x_j$, whenever $fitness < f(m_g)$;
        end
    until Stopping condition;
    return $m_g$;
end
```

6 The active system identification attack

The Active System Identification attack, herein proposed, is intended to assess the coefficients of a transfer function $G(z) = C(z)P(z)$ of an NCS, wherein $C(z)$ is the controller's control function and $P(z)$ is the plant's transfer function, as shown in Fig. 2. The transfer functions are all linear time-invariant (LTI). This attack is performed by a MitM that may be located either in the forward or in the feedback link. For the sake of clarity of the analysis presentation, but without loss of generality, we focus on the case where the MitM is in the feedback link, i.e. between the plant's sensors and the controller's input. To estimate the model of the attacked NCS, the attacker injects an attack signal $a(k)$ and measures the response of the system to such signal.

The complete response of the generic NCS shown in Fig. 2, considering only the inputs $R(z) = \mathcal{Z}[r(k)]$ and $A(z) = \mathcal{Z}[a(k)]$, is expressed in the z domain by Eq. 5:

$$Y(z) = \frac{G(z)}{1+G(z)}R(z) - \frac{G(z)}{1+G(z)}A(z), \tag{5}$$

wherein $Y(z) = \mathcal{Z}[y(k)]$. $\mathcal{Z}$ represents the Z-transform operation. As a premise, in a normal condition, it is considered that $a(k) = 0$ and the system is designed to make $y(k) \to q$, in such way that $y(k) \approx q \forall k > k_s$, i.e. the output $y(k)$ of the NCS converges and stabilizes at a constant value q after a certain amount of samples k_s. Indeed, it is usually one of the main aims of a control system. Now, considering $a(k) \neq 0$, the output $y(k)$, $\forall k > k_s$, may be defined approximately as Eq. 6:

$$y(k) = q - \mathcal{Z}^{-1}\left[\frac{G(z)}{1+G(z)}A(z)\right], \forall k > k_s. \tag{6}$$

Thus, after k_s, the portion of $y(k)$ caused by $r(k)$ can be eliminated by just subtracting q from Eq. 6, which leads to Eq. 7:

$$y_a(k) = y(k) - q = -\mathcal{Z}^{-1}\left[\frac{G(z)}{1+G(z)}A(z)\right], \forall k > k_s. \tag{7}$$

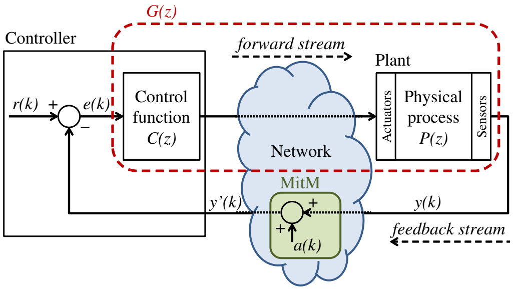


Fig. 2 Active System Identification attack with a MitM in the feedback link

wherein $y_a(k)$ represents the portion of $y(k)$ caused by the attack signal $a(k)$. The value of q can be assessed by the attacker through an eavesdropping attack in the feedback stream, by just capturing $y(k)$ after the stabilization of the NCS. The subtraction of q after k_s makes the system identification attack independent of $r(k)\ \forall k > k_s$. The Active System Identification attack now just relies on the attack signal $a(k)$, which can be chosen, and the response of the system to the attack $y_a(k)$ can be obtained in accordance with Eq. 7. The signal $y_a(k)$ starts with the injection of $a(k)$ and has the size of a monitoring period T.

If the attack input $a(k)$ and its consequent output $y_a(k)$ are known, the model of $G(z)$ can be assessed by applying the known $a(k)$ in an estimated system, defined by Eq. 8:

$$\hat{y}_a(k) = -\mathcal{Z}^{-1}\left[\frac{G_e(z)}{1+G_e(z)}\right] * a(k), \tag{8}$$

wherein $G_e(z)$ is the estimation of $G(z)$ and $\hat{y}_a(k)$ is the output of the estimated system in face of $G_e(z)$. By comparing $\hat{y}_a(k)$ with $y_a(k)$, the attacker is capable to evaluate whether $G_e(z)$ is equal/approximately $G(z)$. Note that $G_e(z)$ is a generic transfer function represented by Eq. 9:

$$G_e(z) = \frac{\alpha_n z^n + \alpha_{n-1}z^{n-1} + \ldots + \alpha_1 z^1 + \alpha_0}{z^m + \beta_{m-1}z^{m-1} + \ldots + \beta_1 z^1 + \beta_0}, \tag{9}$$

wherein n and m are the order of the numerator and the denominator, respectively, while $[\alpha_n, \alpha_{n-1}, \ldots \alpha_1, \alpha_0]$ and $[\beta_{m-1}, \beta_{m-2}, \ldots \beta_1, \beta_0]$ are the coefficients of the numerator and the denominator, respectively, that are intended to be found by this Active System Identification attack. Therefore, to find $G(z)$, the coefficients of $G_e(z)$ are adjusted until the estimated output $\hat{y}_a(k)$ converges to the known $y_a(k)$.

In this sense, the BSA and the PSO are used to iteratively adjust the estimated model, by minimizing a specific fitness function presented in this section until the estimated model $G_e(z)$ converges to the actual $G(z)$ of the real NCS. To compute the fitness of the individuals of the optimization algorithm (i.e. the BSA or PSO), the same attack signal $a(k)$ that caused $y_a(k)$ is applied on the estimated system defined by Eqs. 8 and 9, where the coefficients of $G_e(z)$ are the coordinates $x_j = [\alpha_{n,j}, \alpha_{n-1,j}, \ldots \alpha_{1,j},\ \alpha_{0,j}, \beta_{m-1,j}, \beta_{m-2,j}, \ldots \beta_{1,j}, \beta_{0,j}]$ of an individual j of the BSA/PSO. The output $\hat{y}_{aj}(k)$ is the response of the estimated model (8, 9) in face of $a(k)$, when the coefficients of $G_e(z)$ are x_j. Then, the fitness f_j of each individual j is obtained comparing $\hat{y}_{aj}(k)$ with $y_a(k)$, according to Eq. 10:

$$f_j = \frac{\sum\limits_{k=0}^{N}(y_a(k) - \hat{y}_{aj}(k))^2}{N}, \tag{10}$$

wherein N is the number of samples that exist during the monitoring period T of $y_a(k)$. Note that, if no other inputs – perturbation or noise – occur in the NCS during T, then $\min f_j = 0$ when $[\alpha_{n,j}, \alpha_{n-1,j}, \ldots \alpha_{1,j}, \alpha_{0,j}, \beta_{m-1,j}, \beta_{m-2,j}, \ldots \beta_{1,j}, \beta_{0,j}] = [\alpha_n, \alpha_{n-1}, \ldots \alpha_1, \alpha_0, \beta_{m-1}, \beta_{m-2}, \ldots \beta_1, \beta_0]$, i.e. when the estimated $G_e(z)$ converges to $G(z)$.

An analogy may be established between this Active System Identification attack and the Chosen Plaintext cryptanalytic attack [23], wherein $a(k)$ corresponds to the chosen plaintext, $y_a(k)$ represents the ciphertext, the Eqs. 8 and 9 together correspond to the encryption algorithm, and the actual coefficients $[\alpha_n, \alpha_{n-1}, \ldots \alpha_1, \alpha_0]$ and $[\beta_{m-1}, \beta_{m-2}, \ldots \beta_1, \beta_0]$ of $G_e(z)$ correspond to the secret key.

It is worth mentioning that this attack requires the previous knowledge about the order of the numerator and denominator of Eq. 9 (n and m, respectively). Using the analogy with the Chosen Plaintext cryptanalytic attack, it is equivalent to require the knowledge about the size of the secret key of the encryption algorithm. In this Active System Identification attack, the information of n and m is necessary to define the number of dimensions of the search space of the BSA – or the number of unknown coefficients of $G(z)$ – which must be set to $n + m - 1$. Although this is a constraint of the attack, this information may be inferred if the attacker, at least, knows what the attacked plant is and what type of controller is being used.

7 Results

In Section 7.1, the results obtained by both BSA-based and PSO-based Active System Identification attacks are analyzed and statistically refined in order to provide a demonstration of the degree of accuracy that the attacker may obtain with the proposed attack. Additionally, Section 7.2 presents a set of data injection attacks designed based on the models estimated by the Active System Identification process. The purpose of the simulations of these data injection attacks is to demonstrate how an Active System Identification attack may contribute for the accuracy of other sophisticated attacks.

7.1 Active system identification attack

The attacked system, shown in Fig. 3, consists of a DC motor whose rotational speed is controlled by a Proportional-Integral (PI) controller. This example is chosen due to the use of DC motors in a vast number of real world control systems. Moreover, DC motors has been widely used in previous works about NCS [3, 7, 14, 18, 19]. It is noteworthy that the model herein chosen as an example does not exhaust the potential targets for this attack. NCSs composed by another kinds of LTI devices may also be a target.

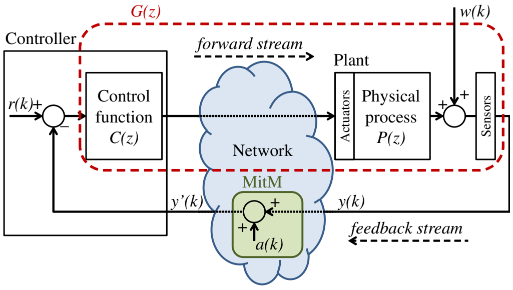


Fig. 3 Attack on a noisy NCS

However, it must be taken into account that the computational cost of the attack, when launched over different LTI systems, may vary with the number of their unknown coefficients – i.e. the number of dimensions of the search space explored by the optimization algorithms (BSA or PSO, in this paper).

The PI control function $C(z)$ and the DC motor transfer function $P(z)$, obtained from [14], are represented by Eqs. 11 and 12, respectively:

$$C(z) = \frac{0.1701z - 0.1673}{z - 1}, \tag{11}$$

$$P(z) = \frac{0.3379z + 0.2793}{z^2 - 1.5462z + 0.5646}. \tag{12}$$

Thereby, the transfer function to be identified $G(z)$ – which is also the open-loop transfer function of the NCS – is defined by Eq. 13:

$$G(z) = C(z)P(z) = \frac{g_1 z^2 + g_2 z + g_3}{z^3 + g_4 z^2 + g_5 z + g_6}, \tag{13}$$

wherein $g_1 = 0.0575$, $g_2 = -0.0090$, $g_3 = -0.0467$, $g_4 = -2.5462$, $g_5 = 2.1108$ and $g_6 = -0.5646$. The sample rate of the system is 50 samples/s and the set point $r(k)$ is an unitary step function. Network delay and packet loss are not taken into account in the simulations of this paper.

The structure of the Eqs. 11 and 12, and so the structure of Eq. 13, are previously known by the attacker once that, as a premise, it is known that the target is an NCS that controls a DC motor using a PI controller. Thus, in these simulations, the goal of the Active System Identification attack is to discover g_1, g_2, g_3, g_4, g_5 and g_6.

The chosen attack signal $a(k)$ is a discrete-time unit impulse (14):

$$a(k) = \begin{cases} 1 \text{ if } k = k_a; \\ 0 \text{ otherwise,} \end{cases} \tag{14}$$

wherein k_a is the single sample in which the attacker interfere in the system by adding 1 to the feedback stream. Note that the discrete-time unit impulse is chosen to excite the NCS due to its short active time – i.e. one sample –, which increases the stealthiness of the attack in the time domain. Moreover, the Fourier transform of an impulse function has an uniform – flat – density in the frequency domain , which is easily masked by the frequency distribution of a white Gaussian noise. This fact also increases the stealthiness of the attack signal in the frequency domain.

The effectiveness of the Active System Identification attack is evaluated with and without noise. To simulate the noise, $w(k) \sim N(\mu, \sigma)$ is inserted in the NCS as indicated in Fig. 3. Note that $w(k)$ is a white Gaussian noise wherein N is a normal distribution, μ is its mean and σ is its standard deviation. In all simulations, the mean is $\mu = 0$ rad/s. The standard deviation is adjusted in such manner that 95% of the amplitudes of $w(k)$ are within $\pm I$ ($I = 2\sigma$). The simulations consider four different noise intensities I: 0 (no noise), 0.0025 rad/s, 0.005 rad/s and 0.01 rad/s. For each noise intensity I, 100 different simulations are executed using each of the mentioned metaheuristics. In each simulation, the feedback stream is captured by the attacker during a period $T = 2s$ (100 samples), starting at sample $k_a + 1$.

The attack model was implemented in MATLAB, where the simulations were carried out. The SIMULINK tool was used to compute $y_a(k)$ and $\hat{y}_{aj}(k)$ – the latter, for each individual j of the optimization algorithms. The parameters of the BSA and PSO described in Sections 4 and 5, respectively, were empirically adjusted through a set of simulations without noise ($I = 0$). These parameters are then used for all noise conditions. In the BSA-based attacks, the parameter η is set to 1. In the PSO-based attacks, the following parameters configuration is used: $\omega = 0.4$, $\varphi_1 = \varphi_2 = 1.5$ and $\delta = 0.1$. In both algorithms, the population is set to 100 individuals and the limits of each dimension of the search space are $[-10, 10]$. In each simulation, the BSA and the PSO are executed for 4500 iterations.

Let S_u be the solution of an attack simulation u, and $g_{i,u}$ the value estimated for the i^{th} coefficient of $G(z)$ in the u^{th} attack simulation. Each attack simulation provides a solution $S_u = [g_{1,u}, g_{2,u}, g_{3,u}, g_{4,u}, g_{5,u}, g_{6,u}]$ containing estimated values for the six coefficients of $G(z)$. In [6], for a given coefficient g_i of $G(z)$, if an estimated value $g_{i,u}$ is beyond two standard deviation from the mean, then $g_{i,u}$ is considered an outlier and eliminated from the set of values found for g_i. After that, the estimated value of each g_i is assumed to be the mean of the remaining $g_{i,u}$. However, in the present work, to improve the accuracy of the estimated model, this statistical refinement is modified. In this paper, if an estimated value $g_{i,u}$ is beyond two standard deviation from the mean, the whole solution S_u (to which $g_{i,u}$ belongs) is considered as an outlier and eliminated from the set of solutions. Doing so, the estimated value of each g_i is assumed to be mean of all $g_{i,u}$ contained in the set of

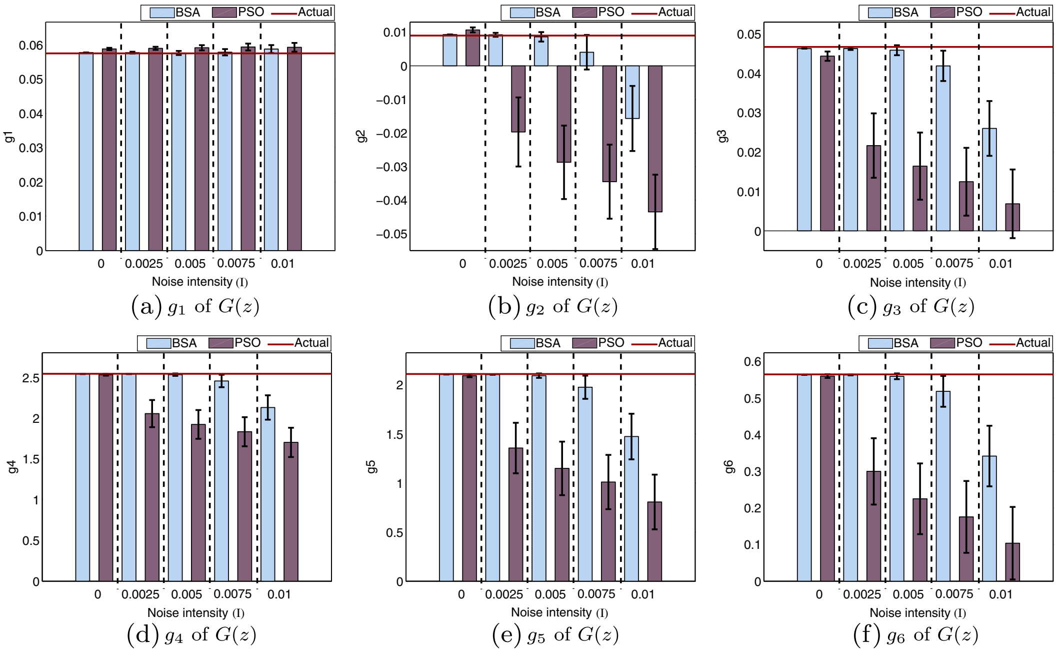
Table 2 Mean estimated coefficients of $G(z)$ after the statistical refinement

		Mean of the coefficients statistically refined in [6]						Mean of the coefficients statistically refined in the present work					
	Noise (I)	$g_1(\times 10^{-2})$	$g_2(\times 10^{-3})$	$g_3(\times 10^{-2})$	g_4	g_5	$g_6(\times 10^{-3})$	$g_1(\times 10^{-2})$	$g_2(\times 10^{-3})$	$g_3(\times 10^{-2})$	g_4	g_5	$g_6(\times 10^{-1})$
BSA	0	5.7756	−9.3337	−4.6261	−2.5431	2.1063	−5.6319	5.7750	−9.3128	−4.6268	−2.5431	2.1063	−5.6319
	0.0025	5.7736	−9.2001	−4.6301	−2.5428	2.1058	−5.6305	5.7714	−9.2299	−4.6294	−2.5428	2.1059	−5.6306
	0.005	5.7826	−9.0411	−4.5528	−2.5345	2.0937	−5.5924	5.7628	−8.6145	−4.5870	−2.5350	2.0944	−5.5931
	0.0075	5.8215	−0.7908	−3.4930	−2.4023	1.8911	−4.7857	5.7843	−4.0346	−4.1886	−2.4578	1.9761	−5.1824
	0.01	5.8561	20.7982	−2.5371	−2.0906	1.3852	−3.1095	5.8763	15.6817	−2.6009	−2.1322	1.4738	−3.4164
PSO	0	5.8799	−10.6784	−4.4361	−2.5341	2.0940	−5.5989	5.8799	−10.6784	−4.4361	−2.5341	2.0940	−5.5989
	0.0025	5.8987	19.7038	−2.1653	−2.0568	1.3567	−2.9982	5.8987	19.7038	−2.1653	−2.0568	1.3567	−2.9982
	0.005	5.9148	28.7309	−1.6431	−1.9242	1.1493	−2.2507	5.9148	28.7309	−1.6431	−1.9242	1.1493	−2.2507
	0.0075	5.9357	34.5026	−1.2472	−1.8347	1.0102	−1.7552	5.9357	34.5026	−1.2472	−1.8347	1.0102	−1.7552
	0.01	5.9288	43.4950	−0.6878	−1.7036	0.8073	−1.0370	5.9288	43.4950	−0.6878	−1.7036	0.8073	−1.0370

remaining S_u. Table 2 presents a summary that compares the results achieved in this work with the results obtained in [6], in both BSA-based and PSO-based attacks. The most accurate results are highlighted. Note that in all cases the most accurate results were achieved by the BSA-based attacks. According to Table 2, the statistical refinement used in the present work in general improves the accuracy of the results obtained by the BSA-based attacks. This improvement is more evident in Section 7.2, where the performance of other attacks designed with the data presented in Table 2 is analyzed. Note that the results shown in Table 2 for the PSO-based attacks are the same as the results of [6]. This occurs because in PSO-based attacks all outlier coefficients belong to solutions wherein all other coefficients are also outliers – i.e. beyond two standard deviations from their means. Thus, in the PSO-based attacks, the whole solution S_u which contains an outlier is eliminated from the set of solutions even when the statistical refinement of [6] is applied.

The mean estimated values of g_1, g_2, g_3, g_4, g_5 and g_6, statistically refined as proposed in this work, are shown in Fig. 4 with a Confidence Interval (CI) of 95%, for different values of noise intensity I. Note that the actual values of these coefficients are also depicted in Fig. 4. In this Figure, it is possible to compare the results achieved by the BSA-based and the PSO-based attacks. According with Fig. 4, it is possible to verify that, for all coefficients of $G(z)$, both the BSA-based and PSO-based attacks present good accuracy when $I = 0$ (i.e. without noise, the mean values of the estimated coefficients are close to their actual values). Despite the similar and accurate performance of the two metaheuristics without noise, it is possible to state that the BSA presented a slightly better performance than the PSO in this noise condition ($I = 0$), specially with regard to the coefficients g_1, g_2 and g_3. Note that the performance of the PSO-based attack is degraded when noise is added to the system. This performance degradation of the PSO occurs for $I \geq 0.0025$ and tends to be more expressive with the increase of I. On the other hand, it is possible to verify in Fig. 4 that the BSA-based attack still present good accuracy for noise intensities up to 0.005. When $I \leq 0.005$, all coefficients estimated by the BSA-based attack present a mean close to their actual values and with a small CI. When $I \geq 0.0075$, the performance of the BSA-based attack decreases with the raise of noise in a more expressive way, being at its worst when $I = 0.01$. In general, among the six coefficients of $G(z)$, the estimation of g_2 presents the lowest accuracy for both BSA-based and PSO-based attacks. This behavior is attributed to a lower sensitivity that the output $\hat{y}_a(k)$ of the estimated system has to the variation of g_2. This means that, in this problem, f_j grows faster for errors in g_1, g_3, g_4, g_5 and g_6 than for errors in g_2, making the BSA population converge less accurately in dimension g_2.

The performance of the attacks can also be evaluated in the k domain through the examples provided in Fig. 5, considering two different intensities of noise: without noise, in Fig. 5a; and with $I = 0.005$, in Fig. 5b and c. Figure 5b shows that, without noise, the response of the system estimated by both BSA-based and PSO-based attacks matches the response of the actual system with high accuracy. In Fig. 5b, even with a noise intensity of $I = 0.005$, the response of the system estimated by the BSA-based attack still matches the response of the actual system, indicating the convergence of $G_e(z)$ to $G(z)$ and ratifying the statistics shown in Fig. 4 for the BSA with such noise intensity. On the other hand, when applying the PSO-based attack with the same noise, as exemplified in Fig. 5c, there is a slight difference between the response of the estimated system and the response of the actual system, produced by the mismatch of the estimated coefficients in the presence of such noise intensity. This exemplifies the worse performance of the PSO-based attacks, when compared with the BSA-based attacks, in face of the same noise intensities.



(a) g_1 of $G(z)$ (b) g_2 of $G(z)$ (c) g_3 of $G(z)$

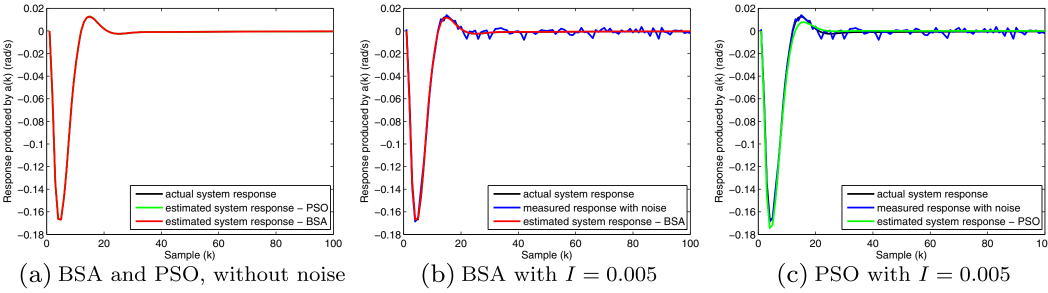
(d) g_4 of $G(z)$ (e) g_5 of $G(z)$ (f) g_6 of $G(z)$

Fig. 4 Mean of the estimated coefficients of $G(z)$, with CI of 95%, in face of different noise intensities I

To synthesize the error of each solution found, $|E_g|$ is computed according to Eq. 15:

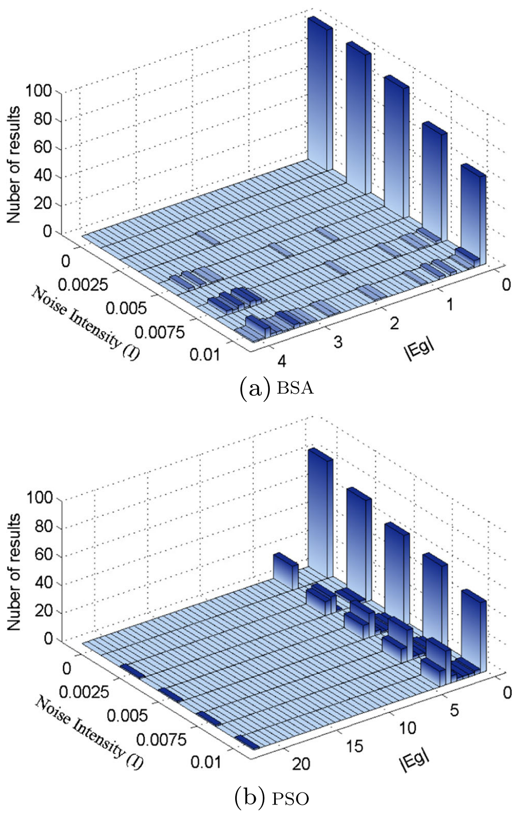
$$|E_g| = \sqrt{\sum_{i=1}^{6} (g_i - g_{ei})^2}, \tag{15}$$

wherein g_i and g_{ei} are the actual and estimated coefficients of the attacked system, respectively, and i is the index number of each of the six coefficients of the model being assessed. Note that $|E_g|$ is the module of a vector composed by the error of each coefficient found, which represents another metric to evaluate the performance of each attack. The histograms of $|E_g|$ are presented in Fig. 6, considering the mentioned noise intensities. It graphically shows that higher values of $|E_g|$ tend to appear more frequently as the noise intensity grows, in both BSA-based and PSO-based attacks. However, based on these histograms it is possible to verify that the mode of $|E_g|$ is close to zero for all noise intensities, using both metaheuristics. This indicates that, even in the presence of noise, most solutions present low deviations from the actual coefficients. Note that, for all noise intensities, the BSA-based attacks provide more results in the modal class – where $|E_g|$ is close to zero – than the PSO-based attacks. Moreover, the worst



(a) BSA and PSO, without noise (b) BSA with $I = 0.005$ (c) PSO with $I = 0.005$

Fig. 5 Response of actual and estimated systems produced by $a(k)$, in face of different noise intensities



(a) BSA

(b) PSO

Fig. 6 Histograms of $|E_g|$ for different noise intensities

results of the BSA-based attacks have an $|E_g|$ of about 4 when $I \geq 0.005$, while the worst results of the PSO-based attacks have an $|E_g| > 20$ when $I \geq 0.0025$. These results, together with the statistics shown in Fig. 4, indicate that the performance of the Active System Identification attack is better when implemented with the BSA than with the PSO. It is worth mentioning that, to achieve these results, the BSA-based attacks consumed an average processing time $(6.68 \pm 0.47)\%$ higher than the PSO-based attacks.

In general, the outcomes indicate that, for the same amplitude of attack signal $a(k)$, the performance of the attack tends to decrease as the noise intensity increases (i.e. when the attack signal-to-noise ratio decreases). The minimum length of the attack signal in terms of number of manipulated samples (i.e. one single sample) improves the stealthiness of the attack in the k domain. On the other hand, a minimum attack signal-to-noise ratio required to guarantee the performance of this attack is a drawback with respect to its stealthiness, from the attacker's point of view. This issue makes more difficult for the attacker to approximate the amplitude of $a(k)$ to the noise amplitude or to noise values that have higher probability to occur, which should help to increase the stealthiness of the attack signal in terms of amplitude.

7.2 Data injection attack

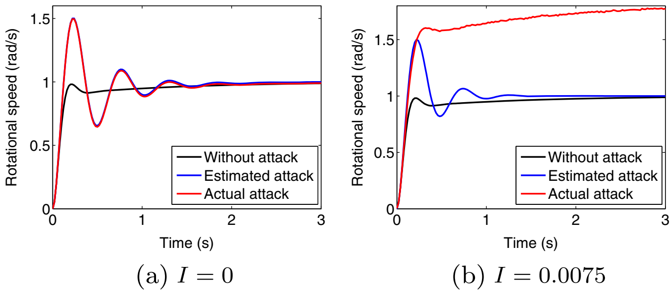
The proposed Active System Identification attack is an useful tool – from the attacker point of view – for the design of other sophisticated and accurate attacks. To demonstrate this capability, this section presents a set of data injection attacks, all designed based on the models estimated in Section 7.1 by the Active System Identification attacks. These data injection attacks aim to cause an overshoot of 50% on the rotational speed of the DC motor during its transient response. As mentioned in Section 1, this physically covert interference [7] may cause stress and possibly damages to the plant, reducing its MTBF.

Table 3 Values of a, b and the overshoot obtained with the data injection attacks

		Noise (I) during the system identification attack				
		0	0.0025	0.005	0.0075	0.01
(I)	a	0.25316	0.25485	0.25523	0.58959	0.53297
	b	0.74679	0.74515	0.74477	−0.07354	0.5911
	Overshoot	49.53%	49.49%	49.65%	(*)	(*)
(II)	a	0.25318	0.25286	0.2551	0.27652	0.31407
	b	0.74682	0.74714	0.7449	0.72348	0.68593
	Overshoot	49.52%	49.78%	49.67%	46.91%	42.42%
(III)	a	0.26801	0.32328	0.32816	0.33074	0.33204
	b	0.73199	0.67672	0.67184	0.66926	0.66796
	Overshoot	47.43%	40.70%	40.37%	40.30%	40.38%

(*) The inaccuracy of the data injection attack caused a collateral effect: an expressive steady state error in the motor's rotational speed

Fig. 7 Data injection attack using models estimated by a BSA-based attack and refined as in [6]



(a) $I = 0$

(b) $I = 0.0075$

Aware of the estimated model of the NCS, the attacker – acting as an MitM – executes the attack function defined by Eq. 16:

$$y'(k) = ay(k-1) + by'(k-1). \tag{16}$$

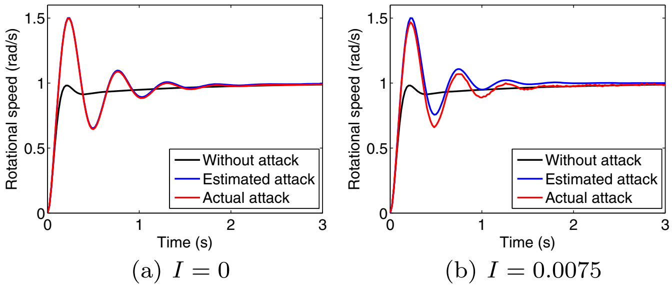
wherein a and b are adjusted through a root locus analysis, considering an estimated open-loop transfer function. Note that the attacker is still on the NCS's feedback stream once that, according with Fig. 3, $y(k)$ is the sensor's output and $y'(k)$ is the controller's input.

The models used to design these data injection attacks are built with the mean estimated coefficients shown in Table 2. Note that a and b have to be adjusted for each estimated model which, in turn, vary with the noise condition, the used optimization algorithm and the applied statistical refinement, as shown in Table 2. The values of a and b used in each data injection attack are shown in Table 3, as well as the respective overshoots achieved with the attack. In Table 3, the row (I) contains the data injection attacks designed with the models estimated by the BSA-based attacks using the statistical refinement of [6]. Row (II) contains the data injection attacks designed with the models estimated by the BSA-based attacks using the statistical refinement proposed in this work. As described in Section 7.1, the models estimated in this work and in [6] by the PSO-based attacks do not change due to the statistical refinement method. Thus, in Table 3, the attacks designed with the models estimated by the PSO-based attacks – statistically refined by either of the two methods – are contained in row (III).

Examples of the data injection attacks shown in Table 3 are depicted, in the time domain, in Figs. 7, 8 and 9. In these figures, the curves named as *estimated attack* represent the results predicted by the attacker when applying the designed attack function (16) on the estimated model – i.e. the model provided by the Active System Identification attack. On the other hand, the curves referred as *actual attack* represent the response of the actual system in face of the same attack function (16). In other words, the curve *estimated attack* is the result achieved in a first moment, during the design stage of the attack, and the curve *actual attack* is the result obtained in a second moment, when the designed attack is launched over the actual system.

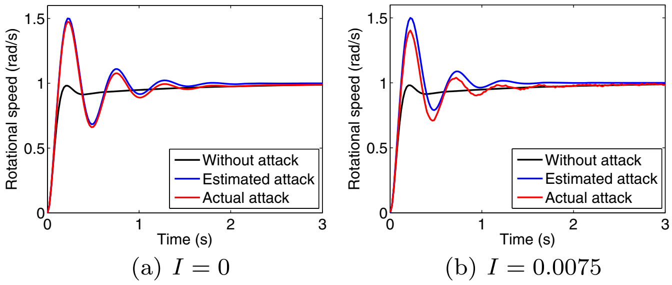
In rows (I) and (II) of Table 3, it is possible to see that, when $0 \leq I \leq 0.005$, the data provided by the BSA-based Active System Identification attacks produce accurate data injection attacks, either with the statistical refinement of [6] or the statistical refinement proposed in the present work. In these data injection attacks, all overshoots lie between 49.49 and 49.78% – i.e. close to the goal of 50%. However, for $0.0075 \leq I \leq 0.01$, the data injection attacks of row (I) – i.e. using the models estimated by BSA-based attacks and refined as in [6] – produce a collateral behavior on the attacked system. They cause expressive steady state errors in the motor's rotational speed, as indicated, for instance,

Fig. 8 Data injection attack using models estimated by a BSA-based attack and refined as herein proposed



(a) $I = 0$

(b) $I = 0.0075$



(a) $I = 0$ (b) $I = 0.0075$

Fig. 9 Data injection attack using models estimated by a PSO-based attack and statistically refined

in Fig. 7b. On the other hand, for $0.0075 \leq I \leq 0.01$, when the statistical refinement proposed in the present work is applied to the BSA-based Active System Identification attacks, the estimated models eliminate the mentioned collateral effects on the data injection attacks. This can be seen in the example shown in Fig. 8b, for $I = 0.0075$, where the response of the actual attack is close to the response of the estimated attack, without a steady state error and with an overshoot of 46.91%. The reason for these different performances is explained by the impact of the statistical refinement in the root locus analysis. When only an outlier coefficient $g_{i,u}$ is eliminated – as in [6] –, instead of eliminating the whole solution S_u from where it belongs – as herein proposed –, the roots of the open-loop transfer function suffer a distortion. For instance, in these simulations, when $0.0075 \leq I \leq 0.01$, the statistical refinement of [6] modifies a pole of $G(z)$ that should be 1. This pole exists due to the use of the PI controller – a premise known by the attacker – and, when modified, influences the adjustment of a and b of Eq. 16. On the other hand, by eliminating the whole solution S_u containing an outlier coefficient $g_{i,u}$, the mean estimated coefficients of $G(z)$ preserve the interdependencies necessary to produce less distorted roots. Note that, as shown in row (III) of Table 3 and in Fig. 9, the PSO-based attacks produce less accurate data injection attacks than the BSA-based attacks statistically refined as proposed in this work. It is worth mentioning that the data injection attacks designed with the models estimated by the PSO-based attacks do not present any collateral effects, using any of the two statistical refinement methods. In both cases, as explained in Section 7.1, the whole solution S_u containing an outlier is eliminated from the set of solutions, producing less distortion in the roots of $G(z)$.

Moreover, with the exception of the attacks of row (I) for $0.0075 \leq I \leq 0.01$, all data injection attacks achieved satisfactory results. However, it is shown that the accuracy of the data injection attack, in general, decreases as the noise intensity increases during the Active System Identification attack.

8 Conclusion

The present work defines and proposes an Active System Identification attack that may be launched over NCSs. The proposed attack is implemented based on two bio-inspired algorithms: the BSA and the PSO. This work demonstrates that the proposed Active System Identification attack is capable to accurately support the design of other sophisticated cyber-physical attacks in NCSs. The results show that the best performance, in general, is achieved by the BSA-based attacks when statistically refined with the method proposed in this paper, specially in the presence of the higher noise intensities.

The capability of the attack to achieve its goal is demonstrated even when: no meaningful information is passing through the NCS's communication links (i.e. when the system had achieved a steady state); the attacker intercepts the communication of the NCS at a single point; and the NCS is noisy.

For future work, we plan to investigate possible techniques to improve the performance of the attack in face of higher noise intensities. Also, we plan – and encourage other researchers – to investigate countermeasures to identify and prevent Active System Identification attacks.

Last, but not least, we plan to improve proposed attack to make it capable to identify systems with uncertain number of unknown coefficients. Preliminary results indicate that, when the number of coefficients is smaller than in the actual system, the algorithm is not able to make the estimated output $\hat{y}_a(k)$ converge to the known $y_a(k)$. In this case, it is not possible to have $\min f_j = 0$, and the global minimum values found by the metaheuristics tends to be high. From the point of view of the attacker, this may be an indicative that the number of coefficients – or dimensions of the metaheuristic – have to be increased, in order to allow $\hat{y}_a(k)$ to match $y_a(k)$. On the other hand, when the number of coefficients is higher than in the actual system, it is possible to have $\min f_j \approx 0$. However, simulations indicate that, even when $\min f_j \approx 0$, the exceeding coefficients does not tend to 0. In this case, the analysis to eliminate the unnecessary coefficients is not straightforward and still has to be developed, in order to make the algorithm robust to uncertainty with respect to the number of unknown coefficients.

Acknowledgments We appreciate the valuable comments and suggestions of the reviewers that contributed to the great improvement of the original version of this paper.

References

1. Amin S, Litrico X, Sastry S, Bayen AM (2013) Cyber security of water scada systems part i: analysis and experimentation of stealthy deception attacks. IEEE Trans Control Syst Technol 21(5):1963–1970
2. Bou-Harb E, Debbabi M, Assi C (2014) Cyber scanning: a comprehensive survey. IEEE Commun Surv Tutorials 16(3):1496–1519
3. Chen X, Song Y, Yu J (2012) Network-in-the-loop simulation platform for control system. In: Asiasim 2012. Springer, pp 54–62
4. Civicioglu P (2013) Backtracking search optimization algorithm for numerical optimization problems. Appl Math Comput 219(15):8121–8144
5. Dasgupta S, Routh A, Banerjee S, Agilageswari K, Balasubramanian R, Bhandarkar S, Chattopadhyay S, Kumar M, Gupta A (2013) Networked control of a large pressurized heavy water reactor (phwr) with discrete proportional-integral-derivative (pid) controllers. IEEE Trans Nucl Sci 60(5):3879–3888
6. de Sa AO, da Costa Carmo LFR, Machado RCS (2017) Bio-inspired active attack for identification of networked control systems. In: 10th EAI international conference on bio-inspired information and communications technologies (BICT). ACM, pp 1–8
7. de Sa AO, da Costa Carmo LFR, Machado RCS (2017) Covert attacks in cyber-physical control systems. IEEE Trans Ind Inf 13(4):1641–1651. https://doi.org/10.1109/TII.2017.2676005
8. El-Sharkawi M, Huang C (1989) Variable structure tracking of dc motor for high performance applications. IEEE Trans Energy Convers 4(4):643–650
9. Farooqui AA, Zaidi SSH, Memon AY, Qazi S (2014) Cyber security backdrop: a scada testbed. In: Computing, communications and IT applications conference (comcomap), 2014 IEEE. IEEE, pp 98–103
10. George NV, Panda G (2012) A particle-swarm-optimization-based decentralized nonlinear active noise control system. IEEE Trans Instrum Meas 61(12):3378–3386
11. Guha D, Roy PK, Banerjee S (2016) Application of backtracking search algorithm in load frequency control of multi-area interconnected power system. Ain Shams Eng J
12. Kennedy R, Eberhart JE (1995) Particle swarm optimization. In: Proceedings of 1995 IEEE international conference on neural networks, pp 1942–1948
13. Langner R (2011) Stuxnet: dissecting a cyberwarfare weapon. IEEE Secur Priv 9(3):49–51
14. Long M, Wu C-H, Hung JY (2005) Denial of service attacks on network-based control systems: impact and mitigation. IEEE Trans Ind Inf 1(2):85–96
15. Öncü S, Ploeg J, van de Wouw N, Nijmeijer H (2014) Cooperative adaptive cruise control: network-aware analysis of string stability. IEEE Trans Intell Transp Syst 15(4):1527–1537
16. Precup R-E, Balint A-D, Radac M-B, Petriu EM (2015) Backtracking search optimization algorithm-based approach to pid controller tuning for torque motor systems. In: 2015 9th annual IEEE international systems conference (syscon). IEEE, pp 127–132
17. Sabău Ş, Oară C, Warnick S, Jadbabaie A (2017) Optimal distributed control for platooning via sparse coprime factorizations. IEEE Trans Autom Control 62(1):305–320
18. Shi Y, Huang J, Yu B (2013) Robust tracking control of networked control systems: application to a networked dc motor. IEEE Trans Ind Electron 60(12):5864–5874
19. Si ML, Li HX, Chen XF, Wang GH (2010) Study on sample rate and performance of a networked control system by simulation. In: Advanced materials research, vol 139. Trans Tech Publ, pp 2225–2228
20. Smith R (2011) A decoupled feedback structure for covertly appropriating networked control systems. In: Proceedings of the 18th IFAC world congress 2011, vol 18. IFAC-papersonline
21. Smith RS (2015) Covert misappropriation of networked control systems: presenting a feedback structure. IEEE Control Syst 35(1):82–92
22. Snoeren AC, Partridge C, Sanchez LA, Jones CE, Tchakountio F, Schwartz B, Kent ST, Strayer WT (2002) Single-packet ip traceback. IEEE/ACM Trans Networking (ToN) 10(6):721–734
23. Stallings W (2006) Cryptography and network security: principles and practices. Pearson Education India, Delhi
24. Teixeira A, Shames I, Sandberg H, Johansson KH (2015) A secure control framework for resource-limited adversaries. Automatica 51:135–148
25. Tran T, Ha QP, Nguyen HT (2007) Robust non-overshoot time responses using cascade sliding mode-pid control. Journal of Advanced Computational Intelligence and Intelligent Informatics 11(10):1224–1231
26. Tulleken HJ (1990) Generalized binary noise test-signal concept for improved identification-experiment design. Automatica 26(1):37–49
27. Uong S, Ngamroo I (2015) Coordinated control of dfig wind turbine and svc for robust power system stabilization. In: 2015 12th international conference on electrical engineering/electronics, computer, telecommunications and information technology (ECTI-CON). IEEE, pp 1–6

APPENDIX C

de Sá *et al. Journal of Internet Services and Applications* (2018) 9:2
DOI 10.1186/s13174-017-0074-z

Journal of Internet Services
and Applications

RESEARCH Open Access

A controller design for mitigation of passive system identification attacks in networked control systems

Alan O. de Sá[1,2*], Luiz F. R. da Costa Carmo[1,3] and Raphael C. S. Machado[3,4]

Abstract

The literature regarding attacks in Networked Control Systems (NCS) indicates that covert and accurate attacks must be designed based on an accurate knowledge about the model of the attacked system. In this context, the literature on NCS presents the Passive System Identification attack as a metaheuristic-based tool to provide the attacker with the required system models. However, the scientific literature does not report countermeasures to mitigate the identification process performed by such passive metaheuristic-based attack. In this sense, this work proposes the use of a randomly switching controller as a countermeasure for the Passive System Identification attack, in case of failure of other conventional security mechanisms – such as encryption, network segmentation and firewall policies. This novel countermeasure aims to hinder the identification of the controller, so that the model obtained by the attacker is imprecise or ambiguous, in such a way that the attacker hesitates to launch covert or model-dependent attacks against the NCS. The simulation results indicate that this countermeasure is capable to mitigate the mentioned attack at the same time that it performs a satisfactory plant control.

Keywords: Networked control system (NCS), Cyber-physical systems, Security, System identification attacks, Switching controller

1 Introduction

A Networked Control System (NCS) is constituted by a physical plant whose dynamics is controlled by a digital controller – i.e. a computational system – through a communication network which, indeed, integrates the cyberspace to the physical domain. The integration of controllers and physical processes via communication networks aims to provide these systems with better operational and management capabilities, as well as reduce costs. By virtue of these advantages, the number of NCSs applied to industrial processes and critical infrastructure systems is increasing [1–10]. A diagram of an NCS is depicted in Fig. 1, wherein $G(z)$ is the transfer function of the plant, $C(z)$ is the control function executed by the controller and both devices are interconnected through the forward and a feedback streams. The forward stream carries the control signals from the controller to the plant's actuators. The feedback stream, in turn, carries the sensed data from the plant to the controller.

Despite the advantages provided by the NCSs, the integration of controllers and physical plants through a communication network also exposes such control systems to threats originated in the cyber domain. In this context, there is a research effort to characterize vulnerabilities and propose security solutions for NCSs.

Recent researches on the security of NCSs demonstrate the development of a set of sophisticated attacks [6, 11, 12] that, to be covert and accurate, are designed based on the models of the attacked system. For instance, in [12, 13][1], the authors present an attack where false data is injected in the communication process of an NCS to degrade the service performed by a plant. The changes driven by this attack are dimensioned so that the modifications in the

*Correspondence: alan.oliveira.sa@gmail.com
[1] Institute of Mathematics/NCE, Federal University of Rio de Janeiro, Av. Athos da Silveira Ramos, 274, 68.530 Rio de Janeiro, Brazil
[2] Admiral Wandenkolk Instruction Center, Brazilian Navy, Enxadas Island, Guanabara Bay, Rio de Janeiro, Brazil
Full list of author information is available at the end of the article

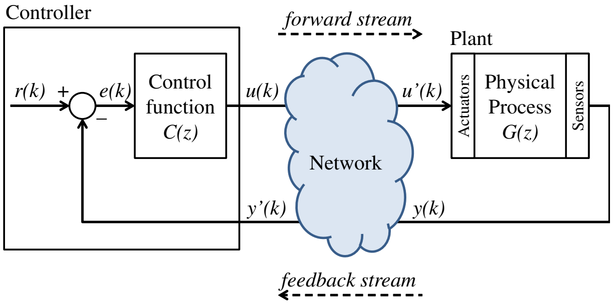


Fig. 1 Networked control system (NCS) [12]

plant's behavior are physically difficult to be perceived. For this reason, this attack is classified as physically covert [12]. To ensure that the attack proposed in [12] is physically covert, the authors indicate that the attacker must plan the offensive based on an accurate knowledge about the system dynamics – otherwise the consequences of the attack may be unpredictable. In this case, the unpredictable behavior of the plant can provide physical evidence that it is being manipulated, drawing the attention to the possibility of a cyber-physical attack.

One possible way to obtain such knowledge about the NCS is through conventional intelligence operations, performed to collect information regarding the design of the system. Another way to gather information about the targeted system is through a Cyber-Physical Intelligence attacks [12]. To this end, the authors of [12] propose a metaheuristic-based Passive System Identification attack, which aims to collect information about the plant's transfer function $G(z)$ and the controller's control function $C(z)$ of an NCS. As shown in Fig. 2 (draw based on the

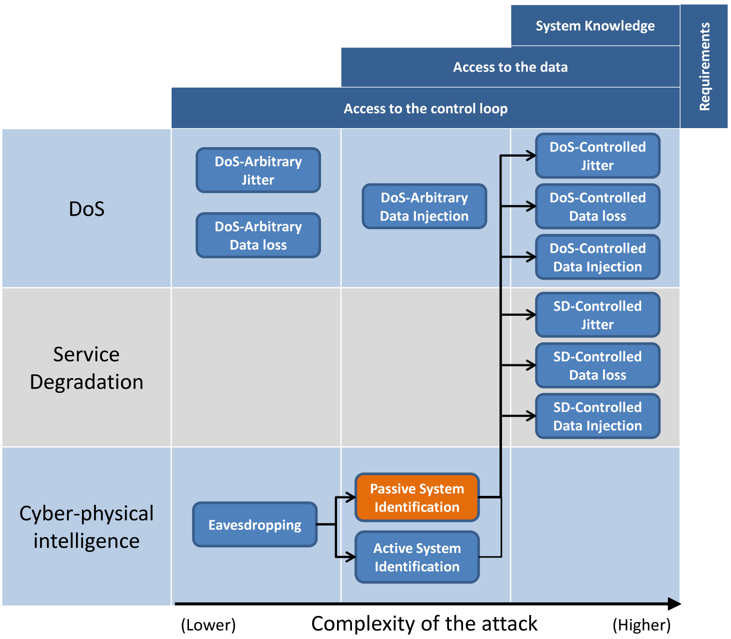


Fig. 2 Classification and requirements of cyber-physical attacks in NCSs

taxonomy proposed in [12]), the Passive System Identification attack constitutes a path to build sophisticated model-dependent attacks, once they are capable to provide the attacker with the required system knowledge. Indeed, the results of [12] demonstrate the effectiveness of the Passive System Identification attack in supporting the design of covert/model-dependent attacks.

Although the authors of [12] encourage the development of countermeasures for the Passive System Identification attack, the scientific literature – to the best of our knowledge – does not report countermeasures to mitigate the identification process performed by such passive metaheuristic-based attack. In this sense, this work aims to discuss and propose a countermeasure for the mentioned attack.

The straightforward countermeasure to prevent the success of a System Identification attack in an NCS is to avoid unauthorized access to the control loop using, for example, network segmentation, demilitarized zones (DMZ), firewall policies and implementing specific network architectures, such as recommended in [14]. A complementary countermeasure – in case the attacker is capable to access the control loop – is to hinder the access to the data flowing in the NCS using, for example, symmetric-key encryption algorithms, hash algorithms and a timestamp strategy to form a secure transmission mechanism between the controller and the plant, as proposed in [15]. However, when the mentioned countermeasures fail and the attacker gain access to the data flowing in the NCS, the alternative to prevent the attacker to obtain the model of the system is to hinder the analysis of the captured data – i.e. make the System Identification algorithm inaccurate/ineffective.

One possible strategy to cause difficulties to the System Identification algorithm is to have, in the NCS, specific control functions that are, at the same time, harder to be identified and capable to control the plant. Based on this reasoning, the contribution of this work is the proposal of a randomly switching controller design as a feasible countermeasure to mitigate the Passive System Identification attack proposed in [12]. As far as we know, there is no other countermeasure reported in the literature that mitigates the Passive System Identification attack by hindering the analysis of signals captured from the NCS.

The rest of this paper is organized as follows: First, in Section 2, some related works are presented. Later, in Section 3, the Passive System Identification attack and a subsequent Data Injection attack are described, in order to provide the underlying information necessary to comprehend the countermeasure proposed in this paper. Then, in Section 4, the switching controller is presented and discussed as a countermeasure for the Passive System Identification attack. After that, Section 5 presents simulation results, where the performance of the switching controller is analyzed from the countermeasure and control perspectives. Finally, in Section 6, some conclusions and possible future works are presented.

2 Related works

The launch of cyber-physical attacks in real world systems, such as the case of the Stuxnet [16] worm, raised the concern of governments and NCS owners, and is motivating the research on cybersecurity of industrial and critical infrastructure facilities. In this context, recent studies demonstrate the development of a set of sophisticated attacks that, to achieve a high level of covertness and accuracy, rely on the knowledge about the model of the attacked system. As recognized by the literature on NCS [12, 17], System Identification attacks are considered a key step in the development of those sophisticated attacks. So, this section presents a review on attacks in NCSs, giving special attention to the role that System Identification attacks play in the context of the cybersecurity of these control systems.

In [18], the authors evaluate the impact of delay jitter and packet loss in an NCS under a Denial of Service (DoS) attack. The conception of such DoS attack does not take into account the models of the controller and physical plant of the attacked NCS (i.e. these models are not known by the attacker). Therefore, to affect the physical process, the attacker arbitrarily floods the network, causing jitter and packet loss in the communication links of the NCS. In this tactic, the excess of packets in the network may reveal the attack, allowing the implementation of countermeasures such as packet filtering [18] or blocking the malicious traffic on its origin [19]. Additionally, as stated in [12], the arbitrary intervention in a system which the models are unknown may lead the plant to an extreme physical behavior, which is not desired if a physically covert [12] attack is intended.

In [4], the authors demonstrate an attack where false signals are transmitted to the controller and the actuator of an NCS. The false signals are randomly generated by the attacker, aiming to cause the instability of the plant (a DC motor). To evaluate this arbitrary data injection attack, the authors propose a testbed for Supervisory Control and Data Acquisition (SCADA) system, using TrueTime (a MATLAB/Simulink based tool). Such arbitrary data injection attack does not require a previous knowledge about the models of the plant and its controller. Therefore, the desired physical effect and the covertness of the attack cannot be ensured due to the unpredictable consequences of the injection of random false signals in a system which the model is not known.

In [20], the authors analyze a wide variety of attacks in NCSs and establish the requirements for the attacks in terms of model knowledge, disclosure and disruption resources. In their work, it is stated that the design of covert attacks requires a high level of knowledge about the model of the attacked system. In [6, 11, 21], examples of covert attacks that agree with the statement provided in [20] are proposed and analyzed. In [11, 21], the attacker, acting as a man-in-the-middle (MitM), injects false data in the forward stream of the NCS to take control of the plant. Then, to make the attack covert, the attacker uses the model of the attacked plant to compute the data injected in the feedback stream. The covertness of the attack proposed in [21] is analyzed from the perspective of the signals arriving at the controller and, as demonstrated in [11], it depends on the difference between the actual model of the plant and the model known by the attacker. In [6], the attacker, aware of the model of the NCS, injects data in its communication links to covertly steal water from the Gignac canal system located in Southern France.

In [6, 11, 20, 21], although the attacks are designed based on the models of the NCS, the authors do not describe how these models are obtained by the attacker. It is just stated that the models, used for the design of the covert/model-dependent attacks, are previously known by the attacker. In order to fill this gap, [12] and [17] propose two new kinds of attack to estimate the models of the attacked system: the Passive System Identification attack [12]; and the Active System Identification attack [17]. As shown in Fig. 2 – and, according to the taxonomy proposed in [12] –, these attacks belong to the category of Cyber-physical Intelligence attacks.

The Passive System Identification attack [12] – formerly referred to as System Identification attack[2] – does not need to inject signals in the NCS to estimate its models. However, the effectiveness of the Passive System Identification attack depends on the occurrence of events – not controlled by the attacker – to produce signals that carry meaningful information for the system identification algorithm. This attack passively estimates the transfer functions of both controller and plant by simply eavesdropping the forward and the feedback streams of the system. On the other hand, the Active System Identification attack constitutes an alternative to the Passive System Identification attack, in situations where the attacker cannot wait so long for the occurrence of such meaningful signals. In the Active System Identification attack, as described in [17], the attacker estimates the open-loop transfer function of the NCS by injecting an attack signal and eavesdropping its response at a single point of interception.

A synthesis of the attacks referred in this section is presented in Table 1. Based on these works, it is possible to verify how useful may be a System Identification attack for the design of covert/model-dependent attacks in NCSs. However, in the scientific literature, we still do not find specific countermeasures to mitigate the identification process performed by the attack proposed in [12]. In this context, this work proposes a countermeasure to mitigate such metaheuristic-based Passive System Identification attack, even when the attacker gets access to the data that is transmitted in the NCS.

3 Covert attack for service degradation

For the sake of completeness, this section describes the attack proposed in [12], in order to provide the information necessary to comprehend the countermeasure proposed in the present work. The attack consists of the joint operation of two attacks: the Passive System Identification Attack, detailed in Section 3.1; and the SD-Controlled Data Injection attack (model-dependent), detailed in Section 3.2. Section 3.3 presents simulation data that demonstrate the effectiveness of the Passive System Identification attack when supporting the design of SD-Controlled Data Injection attacks. These data, obtained from [12], are used as a reference for the evaluation of the proposed countermeasure.

3.1 Passive system identification attack

The Passive System Identification attack, proposed in [12], is intended to assess the coefficients of the plant's transfer function $G(z)$ and the controller's control function $C(z)$ of an NCS. To do so, the attack is modeled as an optimization problem, where the transfer function of the attacked device – be it a controller or plant – is estimated by minimizing a specific fitness function. This modeling is explained in Section 3.1.1. To minimize the mentioned fitness function, the attack uses the Backtracking Search Optimization Algorithm (BSA) [22], briefly described in Section 3.1.2.

3.1.1 Modeling the passive system identification attack as an optimization problem

If the input $i(k)$ and output $o(k)$ signals of an attacked device are known, the model of such device can be assessed by applying the known $i(k)$ in an estimated model, which must be adjusted until its estimated output $\hat{o}(k)$ converges to $o(k)$. In the present attack, the estimated model of the attacked device is iteratively adjusted by the BSA, that minimizes the fitness function herein presented, until the estimated model converges to the actual model of the real device.

To establish the fitness function, firstly, it must be considered a generic LTI system, whose transfer function $Q(z)$ is represented by (1):

$$Q(z) = \frac{O(z)}{I(z)} = \frac{a_n z^n + a_{n-1} z^{n-1} + \ldots + a_1 z^1 + a_0}{z^m + b_{m-1} z^{m-1} + \ldots + b_1 z^1 + b_0}, \tag{1}$$

Table 1 Synthesis of the related attacks

Attack	Method	Knowledge about the system?	How the knowledge is obtained?
Stuxnet *worm* [16]	Modifications in the	Yes	Experiments in a real system
Long, et al. [18]	*Jitter* and packet loss	None	N/A
Farooqui, et al. [4]	Data injection	None	N/A
Smith [11, 21]	Data injection	Yes	Not described
Teixeira [20]	Packet loss	None	N/A
	Data injection	Yes	Not described
Amin [6]	Data injection	Yes	Not described
SD-Controlled [12]	Data injection	Yes	Passive system identification attack
de Sá, et al. [17]	Data injection [a]	Yes	Active system identification attack

[a] In [17], the data injection is not used to cause the disruption or degradation of the plant. The data is injected in the NCS to support the Active System Identification attack

wherein $I(z)$ is the input of the system, $O(z)$ is the output of the system, n and m are the order of the numerator and the denominator, respectively, and $[\,a_n, a_{n-1}, \ldots a_1, a_0]$ and $[\,b_{m-1}, b_{m-2}, \ldots b_1, b_0]$ are the coefficients of the numerator and the denominator, respectively, that are intended to be found by the Passive System Identification attack. Also, it must be considered that $i(k)$ and $o(k)$ represent the sampled input and output of the system, respectively, where $I(z) = \mathcal{Z}[\,i(k)]$, $O(z) = \mathcal{Z}[\,o(k)]$, k is the number of the sample and $\mathcal{Z}$ represents the Z-transform operation.

In this Passive System Identification attack, $i(k)$ and $o(k)$ are firstly captured by an eavesdropping [23, 24] attack, during a monitoring period T. To deal with the eventual loss of samples, that may not be received by the attacker during T, the algorithm holds the value of the last received sample, according with (2), wherein $x(k)$ can either be $i(k)$ or $o(k)$:

$$x(k) = \begin{cases} x(k-1) & \text{if the sample } k \text{ is lost;} \\ x(k) & \text{otherwise.} \end{cases} \tag{2}$$

Then, after acquiring $i(k)$ and $o(k)$, the captured $i(k)$ is applied to the input of an estimated model, that is described by a transfer function whose coefficients $[\,a_{n,j}, a_{n-1,j}, \ldots a_{1,j}, a_{0,j}, b_{m-1,j}, b_{m-2,j}, \ldots b_{1,j}, b_{0,j}]$ are the coordinates of an individual j of the BSA. The application of $i(k)$ to the input of the estimated model results in an output signal $\hat{o}_j(k)$. After obtaining $\hat{o}_j(k)$, the fitness f_j of the individual j is computed comparing the output $o(k)$ – captured from the attacked device – with the output $\hat{o}_j(k)$ of the estimated model, according with (3):

$$f_j = \frac{\sum_{k=0}^{N} (o(k) - \hat{o}_j(k))^2}{\mathbb{K}}, \tag{3}$$

wherein $\mathbb{K}$ is the number of samples that exist during the monitoring period T. Note that, if the attacker does not lose any sample of $i(k)$ and $o(k)$ during T, then $\min f_j = 0$ when $[\,a_{n,j}, a_{n-1,j}, \ldots a_{1,j}, a_{0,j}, b_{m-1,j}, b_{m-2,j}, \ldots b_{1,j}, b_{0,j}] = [\,a_n, a_{n-1}, \ldots a_1, a_0, b_{m-1}, b_{m-2}, \ldots\ b_1, b_0]$, i.e. when the estimated model converges to the actual model of the attacked device.

It is possible to establish an analogy between this System Identification attack and the Known Plaintext cryptanalytic attack [25], wherein $i(k)$ and $o(k)$ correspond to the plaintext and ciphertext, respectively, the form of the generic transfer function $Q(z)$ corresponds to the encryption algorithm and the actual coefficients of $Q(z)$ corresponds to the secret key.

3.1.2 Backtracking search algorithm

In this section, the basic concepts of the BSA are briefly described, in order to provide a clear comprehension regarding the parameters of the algorithm that are adjusted for the attack. The BSA is a bio-inspired metaheuristic that searches for solutions of optimization problems using the information obtained by past generations – or iterations. According to [22], its search process is metaphorically analogous to the behavior of a social group of animals that, at random intervals returns to hunting areas previously visited for food foraging. The general, evolutionary like, structure of the BSA is shown in Algorithm 1.

```
Algorithm 1: BSA
begin
    Initialization;
    repeat
        Selection-I;
        Generate new population
            Mutation;
            Crossover;
        end
        Selection-II;
    until Stopping Condition;
end
```

At the Initialization stage, the algorithm generates and evaluates the initial population $\mathcal{P}_0$ and sets the historical population $\mathcal{P}_{hist}$. The latter constitutes the BSA's memory that, in the Selection-I stage, is updated with historical coordinates visited by the individuals.

During the first selection stage (Selection-I), the algorithm randomly determines, based on a uniform distribution U, whether the current population $\mathcal{P}$ should be kept as the new historical population, and thus replace $\mathcal{P}_{hist}$ (i.e. if $a < b \mid a, b \sim U(0, 1)$, then $P_{hist} = P$). Subsequently, at every iteration, it shuffles the individuals of $\mathcal{P}_{hist}$ (having $\mathcal{P}_{hist}$ been replaced or not).

The mutation operator creates $\mathcal{P}_{mod}$, which is the preliminary version of the new population $\mathcal{P}_{new}$). It does so according to (4):

$$\mathcal{P}_{mod} = \mathcal{P} + \eta \cdot \Gamma(\mathcal{P}_{hist} - \mathcal{P}), \tag{4}$$

wherein η is empirically adjusted through simulations and $\Gamma \sim \mathbb{N}(0, 1)$, with $\mathbb{N}$ being a normal standard distribution. Thus, $\mathcal{P}_{mod}$ is the result of the movement of $\mathcal{P}$'s individuals in the directions established by vector $(\mathcal{P}_{hist} - \mathcal{P})$ and η controls the displacements' amplitude.

In order to create the final version of $\mathcal{P}_{new}$, the crossover operator randomly combines, also following a uniform distribution, individuals from $\mathcal{P}_{mod}$ and others from $\mathcal{P}$.

At the second selection stage (Selection-II), the algorithm firstly evaluates the individuals of $\mathcal{P}_{new}$ using the fitness function f_j (3). After that, individuals of $\mathcal{P}$ (i.e. individuals before applying the mutation and crossover operators) are replaced by individuals of $\mathcal{P}_{new}$ (i.e. individuals obtained after mutation and crossover) with better fitness. Hence, $\mathcal{P}$ includes only new individuals that evolved. While the stopping condition has not yet been reached, the algorithm iterates. Otherwise, it returns the best solution found.

Note that the algorithm has two parameters that are empirically adjusted: the size $|\mathcal{P}|$ of its population $\mathcal{P}$; and η, that establishes the amplitude of the movements of the individuals of $\mathcal{P}$. The parameter η must be adjusted to assign to the algorithm good exploration and exploitation capabilities. With these parameters adjusted, the BSA is used to search for the global minimum of the fitness function described in Section 3.1.1 and, therefore, discover the model of the attacked device.

3.2 SD-Controlled data injection attack

The SD-Controlled Data Injection attack is a model-dependent attack, which the purpose is to reduce the MTBF of the plant and/or reduce the efficiency of the physical process that it performs, by inserting false data in the control loop of the NCS. At the same time, this attack is designed to be physically covert [12].

One way to degrade a physical service is through the induction of an overshoot during the transient response of a plant. The overshoots, or peaks occurred when the system exceeds the targeted value during the transient response, can cause stress and possibly damage physical systems such as mechanical, chemical and electromechanical systems [26, 27]. Additionally, once they occur in a short period, the overshoots are difficult to be noticed by a human observer. Another way to degrade the service of a plant is causing a constant steady state error on it, i.e. producing a constant error when $t \to \infty$. A low proportion steady state error, besides being difficult to be perceived by a human observer, may reduce the efficiency of the physical process or, occasionally, stress and damage the system in the mid/long term.

In the SD-Controlled Data Injection attack, to achieve either of the two mentioned effects, i.e. an overshoot or a constant steady state error, the attacker interfere in the NCS's communication process by injecting false data into the system in a controlled way. To do so, the attacker act as a MitM that executes an attack function $M(z)$, as presented in Fig. 3, wherein $U'(z) = M(z)U(z)$, $U(z) = \mathcal{Z}[u(k)]$ and $U'(z) = \mathcal{Z}[u'(k)]$. The function $M(z)$ is designed based on the models of the plant and the controller, both obtained through the Passive System Identification attack, described in Section 3.1. The effectiveness of the attack, therefore, depends on the design of $M(z)$, which in turn depends on the accuracy of the System Identification attack. It is worth mentioning that, in Fig. 3, although the MitM is placed in the forward stream, it is also possible to perform an attack by interfering in the feedback stream of the NCS.

3.3 Performance of the covert attack for service degradation

This section presents the results of the joint operation of the Passive System Identification attack and the SD-Controlled Data Injection attack. These results, obtained from [12], demonstrate the effectiveness of the Passive System Identification attack when accomplishing its task in an NCS without the countermeasure proposed in this paper.

The attacked NCS has the same architecture of the NCS shown in Fig. 1. It consists of a Proportional-Integral (PI) controller that controls the rotational speed of a DC motor. The PI control function $C_1(z)$ and the DC motor transfer function $G(z)$ are represented by (5) and (6), respectively:

$$C_1(z) = \frac{c_{1,1}z - c_{2,1}}{z - 1} \tag{5}$$

$$G(z) = \frac{g_1 z + g_2}{z^2 - g_3 z + g_4} \tag{6}$$

wherein $c_{1,1} = 0,1701$, $c_{2,1} = -0,1673$, $g_1 = 0,3379$, $g_2 = 0,2793$, $g_3 = -1,5462$ and $g_4 = 0,5646$. The sample

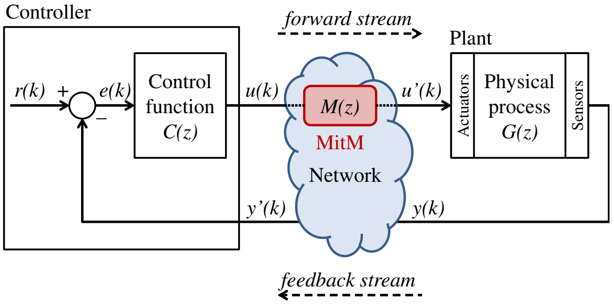


Fig. 3 MitM attack [12]

rate of the system is 50 samples/s and the set point $r(k)$ is a unitary step function.

It is considered that the structure of the Eqs. (5) and (6) are previously known by the attacker given that, as a premise, he/she knows that the target is an NCS that controls a DC motor using a PI controller. Therefore, the goal of the Passive System Identification attack is to discover $g_1, g_2, g_3, g_4, c_{1,1}$ and $c_{2,1}$.

Each time that the DC motor is turned on, the forward and the feedback streams are captured by the attacker during a period $T = 2s$. All initial conditions are considered 0, by the time that the motor is turned on. To assess $[g_1, g_2, g_3, g_4]$, the attacker considers the forward stream as the input and the feedback stream as the output of the estimated plant. In the opposite way, to assess $[c_{1,1}, c_{2,1}]$, the attacker considers the feedback stream as the input and the forward stream as the output of the estimated controller.

According to [12], in these simulations, the BSA population has 100 individuals and $\eta = 1$. To assess the coefficients of the controller $[c_{1,1}, c_{2,1}]$, the algorithm was executed for 600 iterations. To assess the coefficients of the plant $[g_1, g_2, g_3, g_4]$, the number of iterations was increased to 800, due to the higher number of dimensions of the search space in this case. The limits of each dimension of the search space are $[-10, 10]$.

In [12], the authors also demonstrate the robustness of the Passive System Identification attack in the face of sample loss. To evaluate such robustness, they considered four different rates l of sample loss: 0%, 5%, 10% and 20%. For each rate of sample loss, 100 different simulations were executed.

Figure 4 shows the mean estimated values of g_1, g_2, g_3, g_4, $c_{1,1}$ and $c_{2,1}$, considering the four mentioned rates of sample loss. All mean estimated values are represented with a Confidence Interval (CI) of 95%. The actual values of the coefficients of $C_1(z)$ and $G(z)$ are also depicted in Fig. 4. Additionally, the statistics (mean and standard deviation) of the estimated coefficients are presented in Table 2.

Regarding to the coefficients of $G(z)$, Fig. 4 shows that the difference between the mean and the actual values of g_1, g_2, g_3 and g_4 tends to raise with the increase of sample loss. It is also possible to note that the accuracy of the coefficients of $C_1(z)$ is better than the accuracy of the coefficients of $G(z)$, for all rates of sample loss. The means of $c_{1,1}$ and $c_{2,1}$ are closer to their actual values, with a smaller CI. In fact, the optimization process is more effective when computing the coefficients of $C_1(z)$ due to its smaller search space (which that has only two dimensions instead of the four dimensions of the $G(z)$ problem). In Fig. 4, it is possible to verify that, in all cases, the CIs tend to grow with the increase of the sample loss. The same thing occurs with the standard deviations shown in Table 2.

Despite the relative loss of accuracy of the Passive System Identification attack due to the increase of sample loss, such inaccuracy is not expressive even in the worst case (i.e. when $l = 20\%$). This behavior indicates the robustness of the Passive System Identification attack in the face of the loss of samples.

After estimating the models of the attacked plant and its respective control function, the next step is to design the data injection attack. In this sense, the authors of [12] designed an SD-Controlled Data Injection attack aiming to cause an *overshoot* of 50% in the rotational speed of the motor. As shown in Fig. 3, this SD-Controlled Data Injection attack is performed by a MitM in the forward stream. The attack was simulated in MATLAB, aiming to evaluate its accuracy when supported by the Passive System Identification attack.

The attack function executed by the MitM is $M(z) = \mathcal{K}_o$. Performing a root locus analysis considering the obtained models, the attacker adjusts $\mathcal{K}_o$ to make the system underdamped, with a peak of rotational speed 50% higher than its steady state speed. The values of

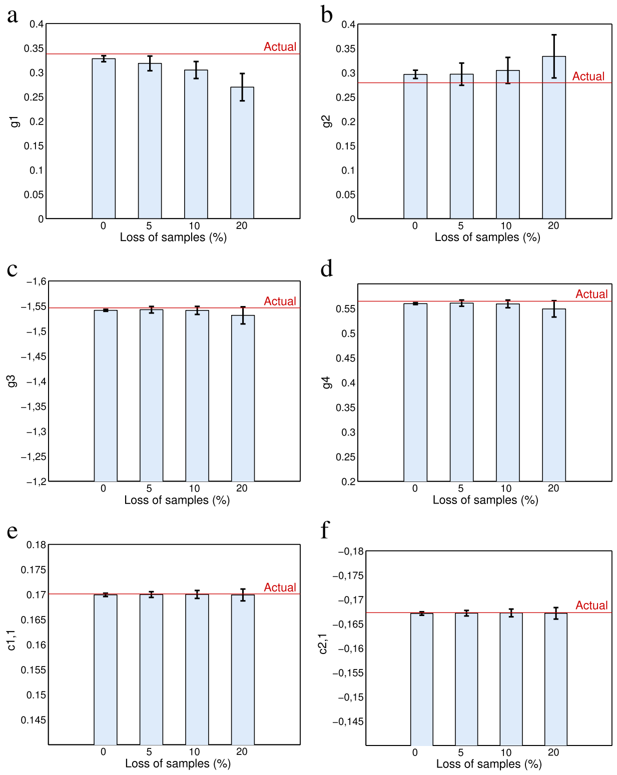


Fig. 4 Mean estimated coefficients of $G(z)$ and $C_1(z)$, in face of different rates of sample loss [12]. **a** g_1 of $G(z)$. **b** g_2 of $G(z)$. **c** g_3 of $G(z)$. **d** g_4 of $G(z)$. **e** $c_{1,1}$ of $C_{1(z)}$. **f** $c_{2,1}$ of $C_{1(z)}$

$\mathcal{K}_o$ were adjusted considering the mean estimated coefficients shown in Table 2. Table 3 shows the values of $\mathcal{K}_o$, estimated considering different rates of sample loss during the Passive System Identification attack, as well as the overshoots obtained with the respective $\mathcal{K}_o$ in the real model. In Fig. 5 it is possible to compare the response of the system without attack, with the response of the system with an attack aiming the overshoot of 50%. The curves referred as *estimated attack*, represent the results predicted by the attacker when the designed attack function $M(z)$ is applied to the estimated model – i.e. the model discovered by the attacker through to the Passive System Identification attack. On the other hand, the curves referred as *actual attack* represent the response of the actual system in the face of the same attack function $M(z)$. In other words, the curve *estimated attack* is the result achieved in a first moment, during the design stage of the attack, and the

Table 2 Statistics of the results obtained with different rates of sample loss [12]

Loss of	Mean						Standard deviation					
samples	g_1	g_2	g_3	g_4	$c_{1,1}$	$c_{2,1}$	g_1	g_2	g_3	g_4	$c_{1,1}$	$c_{2,1}$
0%	0.32793	0.29652	-1.54121	0.55983	0.16991	-0.16712	0.03097	0.04288	0.00986	0.00944	0.00167	0.00178
5%	0.31835	0.29689	-1.54251	0.56085	0.16997	-0.16719	0.07572	0.11523	0.03322	0.03194	0.00287	0.00287
10%	0.30473	0.30461	-1.54110	0.55925	0.16999	-0.16724	0.08781	0.13483	0.04076	0.03922	0.00397	0.00399
20%	0.26963	0.33352	-1.53119	0.54916	0.16989	-0.16716	0.14120	0.22378	0.08596	0.08313	0.00596	0.00598

Table 3 Values of $\mathcal{K}_o$ and overshoots obtained with the attacks [12]

	Sample loss in the passive system identification attack			
	0%	5%	10%	20%
$\mathcal{K}_o$	4.0451	4.0745	4.0828	3.796
Overshoot in the real model	48.90%	49.43%	49.57%	45.94%

curve *actual attack* is the result obtained in a second moment, when the designed attack is launched over the actual system. It is noteworthy that the attack to the actual model – represented by the *actual attack* curve – presents, in the time domain, a response quite similar to the attack estimated with the model obtained by the Passive System Identification attack – represented by the *estimated attack* curve. This can be verified not only in the case where the system is identified with 0% of sample loss, but also in the worst considered case, i.e. with 20% of sample loss. It is worth mentioning that all responses presented in Fig. 5 converge to the setpoint (1 rad/s).

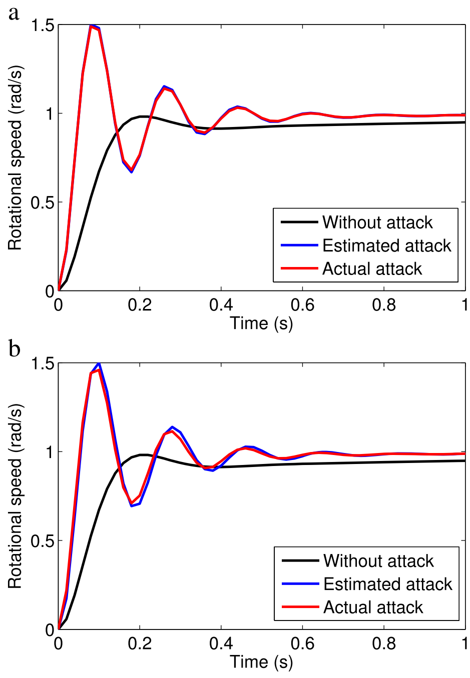


Fig. 5 Response of the plant to SD-Controlled Data Injection attacks designed to cause an overshoot of 50% in the rotational speed of the motor [12]. **a** Attack based on the data obtained without sample loss. **b** Attack based on the data obtained with 20% of sample loss

According to Table 3, it is possible to state that the SD-Controlled Data Injection attack, when supported by the Passive System Identification attack, is capable to accurately modify the physical response of the system, considering all evaluated rates of sample loss. In the worst case, i.e. with 20% of sample loss, it caused an overshoot of 45.94% (quite close to the desired 50%). Such accuracy allows the attacker to keep his offensive under control, leading the system to a behavior that is predefined as physically covert and capable to degrade the service performed by the plant under attack. These simulations provide conclusive data regarding the effectiveness of the Passive System Identification attack when it is used as a tool to support the design of a covert/model-dependent attack.

It is noteworthy that the manipulation of the rotational speed of a DC motor is used only to exemplify a physically covert interference in an NCS. This example is chosen due to the human difficulties to accurately estimate the rotation speed of objects under certain conditions. It is known, for instance, that under some conditions the apparent rotation speed is affected by the stimulus configuration (defined by the shape, size, and other characteristics of the rotating object) [28, 29]. Intuitively, it can be considered that, under those conditions, the perception of 50% of overshoot in the rotation speed may also be difficult to be perceived, especially because of its short duration. Although the authors of [12] use this example in their paper, it is worth mentioning that the concept of a physically covert attack can be extended to other interferences where, as defined in [12], the physical effects cannot be easily noticed or identified by a human observer, or can eventually be understood as a consequence of another cause, other than an attack.

4 Mitigation using switching controllers

As discussed in Section 1, one possible strategy to mitigate the Passive System Identification attack is to build the NCS with specific transfer functions that are harder to be identified. Therefore, it is necessary to analyze the two transfer functions $C(z)$ and $G(z)$, shown in Fig. 1, to verify what can be done to hinder the identification of the NCS. Regarding the plant, it is not desired or even feasible to modify its transfer function $G(z)$ just to make it harder to be identified. This follows from the simple fact that the plant's transfer function is a consequence of the physical structure of the controlled system. In other words, modify $G(z)$ means to modify the physical process being controlled, which is not convenient. However, it is reasonable to think about the design of controllers that are capable to meet, simultaneously, two objectives:

Objective I - Comply with the control requirements of the plant. In general, the primary requirement is to preserve the stability of the system. However, additional requirements – such as low settling time, low overshoot, etc. – may be considered depending on the process being controlled.

Objective II - Hinder the identification of the controller, so that the model obtained by the attacker is imprecise or ambiguous, in such a way that the attacker hesitates to launch covert or model-dependent attacks against the NCS.

Considering these two objectives, this work proposes the use of randomly switching controllers to mitigate Passive System Identification Attacks and, thus, prevent the design of covert/model-dependent attacks. Note that, the use of a switching controller does not avoid the identification of the plant's transfer function $G(z)$ by the Passive System Identification attack described in Section 3.1. Regardless of the controller switchings, the plant's transfer function is still an LTI system that can be identified by the mentioned System Identification attack, based on the analysis of the plant's input and output signals.

A Switching Controller, shown in Fig. 6, is composed by a set of N control functions $C_i(z)$, $i \in \mathcal{I} = \{1, \ldots, N\}$, that are switched by a switching rule S, to perform the control of a plant $G(z)$. If all control functions $C_i(z)$ and the plant's transfer function $G(z)$ are linear, as the NCS herein discussed, then the system is referred as a *switched linear system* (SLS). For the sake of clarity, but without loss of generality, in the present work, the switching controller is represented and discussed with only two control functions $C_1(z)$ and $C_2(z)$ – i.e. $N = 2$.

In a conventional switching controller [30–33], whose sole objective is to control the plant, the switching rule S, in general, orchestrates the switching events based on the plant and/or network behaviors. However, in the solution proposed in this work, the switching rule is not driven by the plant and/or network behaviors.

To achieve both Objectives I and II, the switching rule herein proposed operates as the Markov chain shown in Fig. 7. In this scheme, the control functions are switched at random intervals, in accordance with the probabilities $p_{11}(l)$, $p_{12}(l)$. $p_{21}(l)$ and $p_{12}(l)$, wherein l is the number of sampling intervals occurred since the last switch. The probabilities, $p_{12}(l)$ and $p_{21}(l)$ are taken from the probability density function (PDF) shown in Fig. 8, wherein a is the minimum number of sampling intervals that the system have to remain in the same state and b is the maximum number of sampling intervals that the system can remain in the same state. Note that $p_{11}(l) = 1 - p_{12}(l)$ and $p_{22}(l) = 1 - p_{21}(l)$.

The reason to switch at random intervals is that, according to [34], if the switching times are known, the identification of the SLS is straightforward. However, when the switching times are not available, the identification of the SLS turns into a nontrivial task. Moreover, even if the attacker obtain the plant's transfer function $G(z)$ and – somehow – discovers the control functions $C_i(z)$, the random switching rule still hinders the covert/model-dependent attack described in Section 3.2. This follows from the simple fact that it is more difficult to synchronize the interference caused by the covert/model-dependent attacks with the controller states, which are switched at random intervals.

However, despite the benefits that the switchings can bring from the point of view of a countermeasure, it can affect the stability of the NCS. Even if all subsystems of an SLS are stable, there are situations in which the switching events can make the SLS unstable. According to [7, 35], to be stable under arbitrary and unrestricted switchings, the SLS must meet two conditions:

1. All its subsystems must be asymptotically stable; and

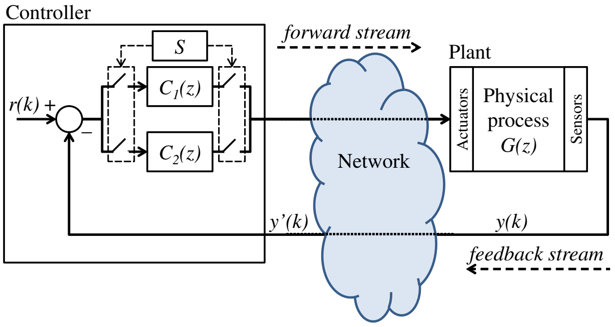


Fig. 6 NCS with a switching controller

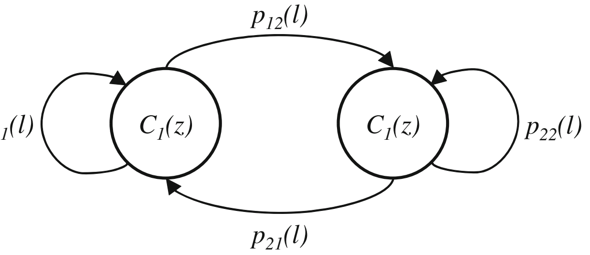


Fig. 7 Markov chain switching rule

2. There must exist a common Lyapunov function for all of its subsystems.

Note that, in the case of the NCS shown in Fig. 6, each subsystem is constituted by the plant transfer function $G(z)$ arranged in a closed loop with one control function $C_i(z)$. So, to make the NCS stable under arbitrary and unrestricted switching, all control functions $C_i(z)$, $i \in \mathcal{I} = \{1, 2\}$, have to be designed in order to meet the two aforementioned conditions.

Another valid strategy to obtain stability in an SLS with stable subsystems is by restricting the switching events. This can be done, for example, by establishing a minimum *dwell time* – i.e. the time between two consecutive switches. In an SLS, the instability generated when switching among two – or more – stable subsystems is caused by the failure to absorb the energy increase, caused by the switchings [35]. Intuitively, it is reasonable to think that if the SLS stays at stable subsystems long enough – using a slow switching rule – it becomes able to avoid the energy increase caused by the switchings, maintaining the desired stability. As proved in [36], it is always possible to preserve the stability of an SLS when all the subsystems are stable and the dwell time is sufficiently large. Actually, it is not critical if the SLS occasionally have a smaller dwell time, provided this does not occur too frequently. As demonstrated in [37], if all the subsystems are exponentially stable, then the SLS remains exponentially stable provided that the *average dwell time* is sufficiently large. In [38], this concept of *average dwell-time* is extended to the discrete-time switched systems – which is the case of an NCS endowed with the proposed countermeasure.

In the present work, instead of designing $C_1(z)$ and $C_2(z)$ to make the SLS stable under arbitrary and unrestricted switchings – i.e. meeting both conditions 1 and 2 – the restricted switching strategy is used. Thus, $C_1(z)$ and $C_2(z)$ are firstly designed based on the root-locus analysis, in order to make each subsystem stable. Then, the overall stability of the SLS is obtained by adjusting the parameters a and b of the PDF shown in Fig. 8, aiming an *average dwell-time* that makes the NCS stable.

Besides being adjusted for stability, parameters a and b also have to be adjusted to hinder the system identification attack. So, concerning Objective I, specifically for the sake of stability, a and b are increased as much as possible to ensure the minimum *average dwell-time* required for stability. On the other hand, concerning Objective II, a and b are adjusted to make the Passive System Identification Attack as much imprecise/ambiguous as possible, which not necessarily occur with high dwell

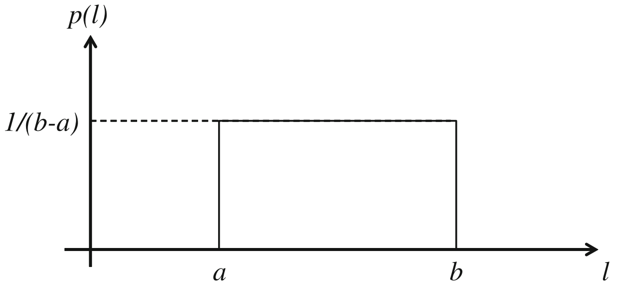


Fig. 8 PDF of p_{12} and p_{21}

times. In this sense, in this work, a and b are empirically adjusted in order to satisfy the two potentially conflicting objectives.

5 Results

As mentioned in Section 4, the design of the switching controller must meet simultaneously two objectives: hinder the identification process; and comply with the plant's control requirements. The results concerning these two objectives are presented in Sections 5.1 and 5.2, respectively, in order to demonstrate the feasibility of the solution from both perspectives. Additionally, Section 5.3 demonstrates the impact caused in the SD-Controlled Data Injection attack, described in Section 3.2, when the Passive System Identification Attack is mitigated by the proposed countermeasure.

In Sections 5.1 and 5.2, the results obtained with the proposed countermeasure are compared with the results obtained in an NCS without the proposed countermeasure – i.e. endowed with a non-switching controller. For this comparison, the NCS specified in Section 3.3 (with a non-switching controller) is used as reference.

The NCS with the proposed countermeasure has the same architecture shown in Fig. 6 and controls a DC motor whose transfer function is also defined by (6) – i.e. it controls the same plant that is controlled by the NCS with a non-switching controller described in Section 3.3. The sample rate of this system is also 50 samples/s and the set point $r(k)$ is a unitary step function. The switching controller has two control functions: $C_1(z)$, that is the same control function (5) of the non-switching controller; and $C_2(z)$ defined by (7),

$$C_2(z) = \frac{c_{1,2}z + c_{2,2}}{z-1}. \tag{7}$$

wherein $c_{1,2} = 0.001$ and $c_{2,2} = 0.0002$. So, the NCS with the switching controller is an SLS with two subsystems. The control functions $C_1(z)$ and $C_2(z)$ are designed to make each subsystem stable – when separately analyzed – and are randomly switched based on the switching rule defined by the Markov chain and the PDF shown in Figs. 7 and 8, respectively. The parameters a and b of the PDF were empirically adjusted to $a = 40$ and $b = 60$, in order to meet Objectives I and II defined in Section 4. Regarding Objective I, it is worth mentioning that a and b were empirically adjusted aiming, primarily, the global stability of the SLS. However, the settling time and the overshoot of the plant are also evaluated in Section 5.2.

5.1 Mitigating the passive system identification attack

This section presents the results obtained by the Passive System Identification attack, when attacking both switching and non-switching controllers. For each controller, 100 attack simulations were performed. The parameters of the BSA are the same as those defined in Section 3.3, and the forward and feedback streams are also captured by the attacker during a period $T = 2s$ (100 samples). To evaluate the proposed countermeasure, we considered the scenario where the attacker obtained the best performance in Section 3.3 – i.e. without packet loss.

The coefficients estimated by all attack simulations (100 for each controller) are presented in Fig. 9. Recall that the non-switching controller just have one control function $C_1(z)$, while the switching controller has two control functions $C_1(z)$ and $C_2(z)$. Note that the actual values of the coefficients $[c_{1,1}, c_{2,1}]$ and $[c_{1,2}, c_{2,2}]$ of the two control functions $C_1(z)$ and $C_2(z)$, respectively, are also depicted in Fig. 9. By observing Fig. 9a and b, it is possible to state that the estimated coefficients of the non-switching controller are precise and accurate. In this case, the estimated coefficients are concentrated close to the actual values of $c_{1,1}$ and $c_{2,1}$. This concentration indicates that, with the non-switching controller, the

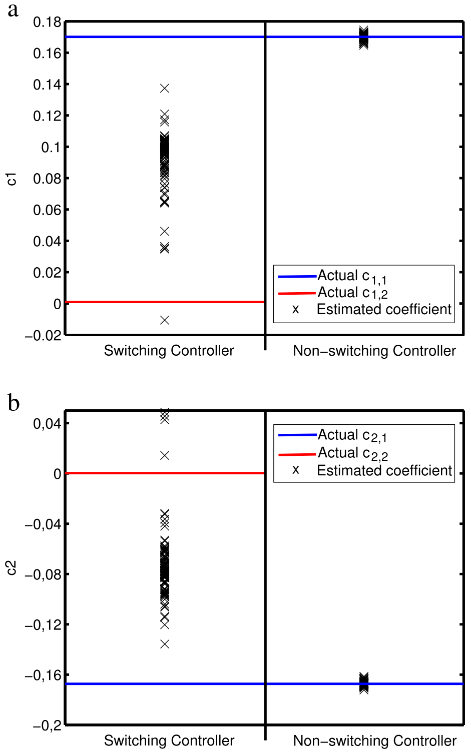


Fig. 9 Coefficients estimated by the passive system identification attack. **a** $c_{1,1}$ of $C_1(z)$ and $c_{1,2}$ of $C_2(z)$. **b** $c_{2,1}$ of $C_1(z)$ and $c_{2,2}$ of $C_2(z)$

Passive System Identification attack provides the information and the confidence that the attacker needs to design a covert/model-dependent attack – such as the SD-Controlled Data Injection attack demonstrated in Section 3.3. On the other hand, Fig. 9 shows that the use of the switching controller causes the dispersion of the estimated coefficients, reducing the precision and the accuracy of the Passive System Identification attack. With the switchings, the set of estimated values are spread and does not accurately indicate any of the coefficients of $C_1(z)$ and $C_2(z)$. It is worth mentioning that this spreading has a dissuasive effect. It increases the uncertainty of the attacker regarding the model of the attacked controller, in such way that the attacker may hesitate to proceed with his intention of a covert/model-dependent attack.

The impact of the switching controller in the attack performance can also be verified through the analysis of the global minimum values obtained for the fitness function (3). With the switching controller, the global minimum values of all attack simulations are between 2.64×10^{-04} and 8.53×10^{-04} (the mean is 7.42×10^{-04}, and the standard deviation is 1.70×10^{-04}). On the other hand, with the non-switching controller, all global minimum values are between 1.70×10^{-09} and 1.44×10^{-06} (the mean is 1.84×10^{-07}, and the standard deviation is 2.70×10^{-07}). Recall that, as discussed in Section 3.1.1, without sample loss, the minimum value of (3) is $\min f_j = 0$ when the attacked device is perfectly identified. So, the higher order of the global minimum values obtained with the switching controller also demonstrates the effectiveness of the proposed countermeasure. From the attacker point of view, these higher global minimum values may indicate that the Passive System Identification attack was not effective in obtaining the model of the attacked device. In this sense, the attacker must hesitate to launch covert/model-dependent attacks based on the information gathered by the Passive System Identification attack.

Another way to evaluate the impact of the proposed countermeasure in the Passive System Identification attack is through the zero-pole maps shown in Fig. 10. Figure 10a shows the zeros estimated by the simulations using the non-switching controller. Figure 10b, in turn, shows the zeros estimated by the simulations using the switching controller. Note that, in the simulations with the non-switching controller, the estimated zeros accurately meet the actual zero of $C_1(z)$. On the other hand, Fig. 10b shows that when the proposed countermeasure is used, the estimated zeros are spread and do not concur for the actual zeros of $C_1(z)$ and $C_2(z)$ – i.e. the control functions of the switching controller.

It must be considered the possibility that the attacker, after some time, detects that the controller is changing its behavior over the time like a switching controller. In this case, it is reasonable to think that the attacker would try to estimate the control functions based on smaller monitoring periods T, to avoid measurements containing switching events. Considering this hypothesis, the performance of the Passive System Identification attack is evaluated using the following monitoring periods T: 0.2s, 0.4s, 0.6s, 0.8s, 1.0s and 1.2s. Note that the maximum T in which the attacker can measure a signal without switchings is $T_b = 0.02b = 1.2s$. Therefore, to evaluate this tactic (of reducing T), the Passive System Identification attack is performed firstly during the execution of $C_1(z)$ and, after that, during the execution of $C_2(z)$. For the identification of $C_1(z)$ all monitoring periods start at $t = 0s$. For the identification of $C_2(z)$ all monitoring periods start at the first switching event (when $C_2(z)$ starts to be executed).

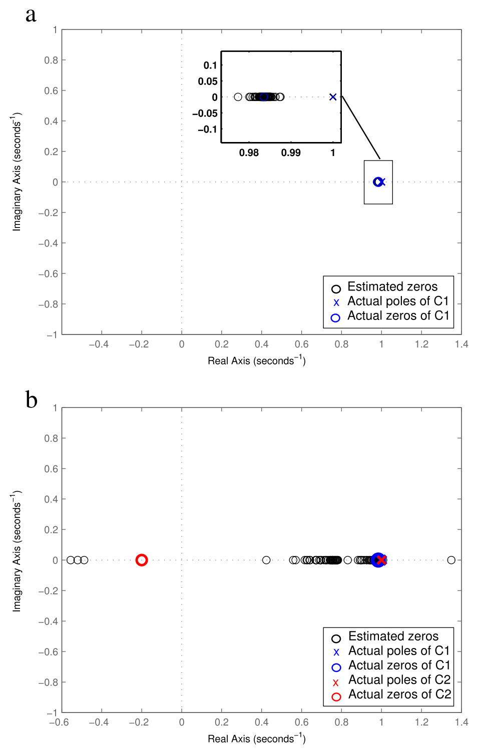


Fig. 10 Zeros and poles estimated by the Passive System Identification attack. **a** Using the non-switching controller. **b** Using the switching controller

For each control function and each monitoring period, 33 attack simulations were executed. Figure 11 shows the

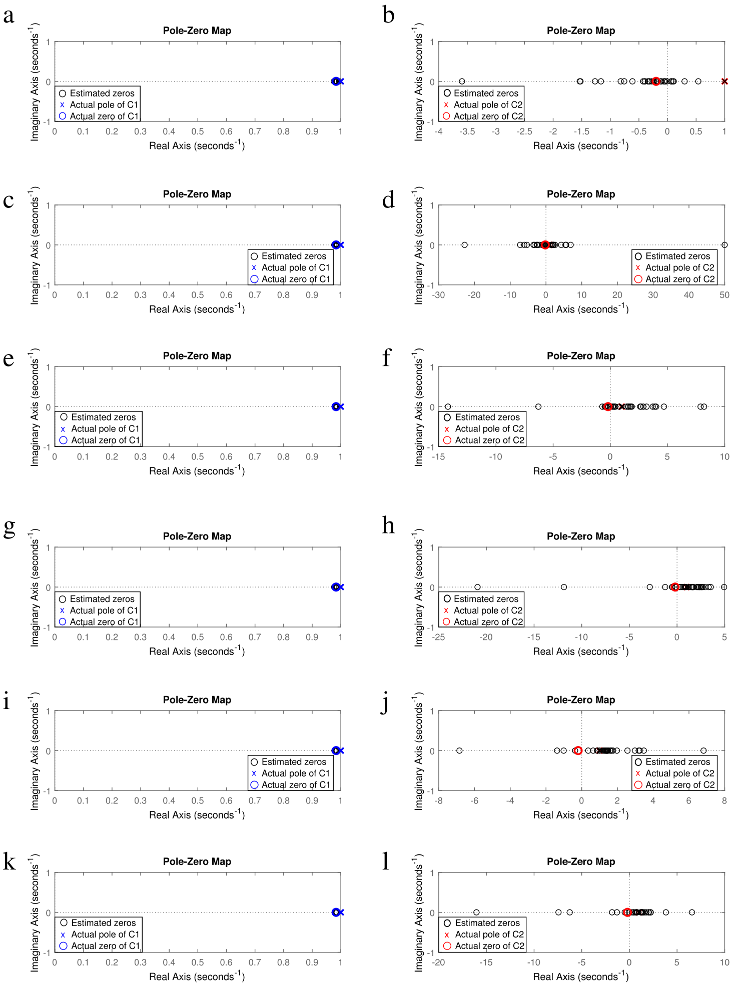


Fig. 11 Zeros and poles estimated by the Passive System Identification attack for smaller monitoring periods T (without a switching event during T). **a** Identifying C_1 with $T = 0.2s$ starting at $t = 0$. **b** Identifying C2 with $T = 0.2s$ starting at the first switching event. **c** Identifying C_1 with $T = 0.4s$ starting at $t = 0$. **d** Identifying C_2 with $T = 0.4s$ starting at the first switching event. **e** Identifying C_1 with $T = 0.6s$ starting at $t = 0$. **f** Identifying C_2 with $T = 0.6s$ starting at the first switching event. **g** Identifying C_1 with $T = 0.8s$ starting at $t = 0$. **h** Identifying C_2 with $T = 0.8s$ starting at the first switching event. **i** Identifying C_1 with $T = 1.0s$ starting at $t = 0$. **j** Identifying C_2 with $T = 1.0s$ starting at the first switching event. **k** Identifying C_1 with $T = 1.2s$ starting at $t = 0$. **l** Identifying C_2 with $T = 1.2s$ starting at the first switching event

estimated zeros of $C_1(z)$ and $C_2(z)$ considering each of the mentioned monitoring periods T. It is possible to verify that, for these monitoring periods, the estimated zeros of $C_1(z)$ are quite close to the actual zero. However, although $C_1(z)$ was satisfactorily identified with small T, Fig. 11 shows that, for all T, the estimated zeros of $C_2(z)$ are spread and do not accurately meet the actual zero of $C_2(z)$. These results indicate that small monitoring periods T may not be enough to identify some control functions, such as happened with $C_2(z)$. In this case, the switching controller arises as a good strategy to limit the available monitoring period, which causes difficulties for this metaheuristic-based Passive System Identification attack. Additionally, it is worth mentioning that even if the attacker somehow identifies all control functions $C_i(z)$, the random switching rule still mitigates the launch of a subsequent covert/model-dependent attack. As discussed in Section 4, this follows from the fact that it is more difficult to synchronize the interference caused by a covert/model-dependent attack with the controller states, which are switched at random intervals. Moreover, it is not trivial to find a single $M(z)$ capable to produce the intended controlled behavior for all $C_i(z)$ – in case the attacker choose this tactic to overcome the need to synchronize the covert/model-based attack.

The spreading of the estimated zeros in Fig. 10b, the inaccuracy of the estimated coefficients shown in Fig. 9, and the higher global minimum values found by the BSA demonstrate the effectiveness of the switching controllers in mitigating the Passive System Identification attack. With the proposed countermeasure, it is possible to state that the model obtained by the attacker is imprecise/ambiguous in such a way that the attacker may hesitate to launch a subsequent covert/model-dependent attack. Therefore, Objective II defined in Section 4 is met.

If an attacker, aiming to cause an overshoot of 50% in $y(k)$ (for example), implements an attack function $M(z)$ in the forward stream of an NCS, as shown in Fig. 3, then $y(k)$ is defined by (8):

$$y(k) = \mathcal{Z}^{-1}\left[\frac{C(z)M(z)G(z)}{1 + C(z)M(z)G(z)}R(z)\right]. \tag{8}$$

Similarly, if the attacker implements $M(z)$ in the feedback stream, then $y(k)$ is defined by (9):

$$y(k) = \mathcal{Z}^{-1}\left[\frac{C(z)G(z)}{1 + C(z)M(z)G(z)}R(z)\right]. \tag{9}$$

Note that in both cases, in the presence of an attack, the dynamics of $y(k)$ rely on $C(z)$, $G(z)$ and $M(z)$, considering that $R(z) = \mathcal{Z}\left[u(k)\right]$ is a step function. Therefore, if the attacker aims to cause an overshoot of 50% in $y(k)$, the design of $M(z)$ will require the knowledge of $C(z)$ and $G(z)$. The results shown in this section indicate that, with the proposed countermeasure, the attacker cannot accurately estimate the control functions of the NCS using the Passive System Identification attack. Therefore, even if the attacker is still able to identify the plant model (which is not mitigated by this countermeasure), he/she will not be able to design $M(z)$ to cause the 50% overshoot based only on the model of the plant, regardless of whether $M(z)$ is implemented in the forward or the feedback stream.

5.2 Complying the control requirements

In this section, the performance of the proposed countermeasure is analyzed from the control perspective. The aim of the simulations herein presented is to identify the possible impacts that the countermeasure may produce in the behavior of the plant. This analysis encompasses the following control aspects: stability; overshoot; and settling time. Considering these aspects, the performance of the switching controller is compared with the performance of the non-switching controller. Given the stochastic nature of the proposed countermeasure, which randomly switches among two control functions, the mentioned aspects are evaluated through a set of 100,000 simulations.

Figure 12 shows the responses of the plant, in the time domain, with and without the proposed countermeasure. The responses obtained with the proposed countermeasure – i.e. using the switching controller – are represented by the highlighted area. The bounds of this area are drawn based on the maximum and minimum values of the output $y(t)$ of the plant, considering all 100,000 simulations. In other words, when using the proposed countermeasure, all output signals $y(t)$ provided by the simulations are inside this area. The deterministic response of the plant without this countermeasure – *i.e.* when using the non-switching controller – is represented by the red line depicted in Fig. 12. Note that, for $0 \leq t \leq 0.8s$ the responses using the switching controller are the same as the response using the non-switching controller. This is caused by the minimum number of sampling intervals that the system has to remain in the same state, which is set to $a = 40$ samples (or $0.8s$, in the time domain).

Based on Fig. 12, considering all 100,000 simulations, it is possible to verify that the NCS with the proposed countermeasure is stable and the output of the plant does not present a stationary error – it always converges to the set point of 1 rad/s. In these aspects, from the control perspective, the proposed countermeasure presents the same performance as the non-switching controller. Also, the highlighted area indicates that the overshoots obtained with the countermeasure are not expressive, not exceeding 2.93% of the set point.

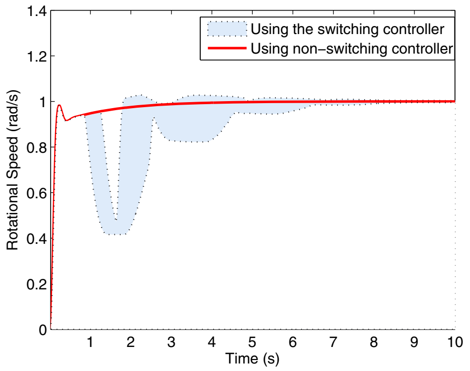


Fig. 12 Response of the plant in the time domain

However, due to the successive switchings, it is possible to verify in Fig. 12 that the settling time obtained with the proposed countermeasure is higher than the settling time obtained with the non-switching controller. With the non-switching controller, the deterministic settling time of the plant is 2.4s. On the other hand, with the switching controller, the settling time t_s of the plant is stochastic and depends on the random sequence of dwell times occurred before achieving t_s. The settling times of all 100,000 simulations using the switching controller are represented in the histogram shown in Fig. 13. The minimum and maximum settling times are 2.88s and 6.42s, respectively, and the mean is $4.2827s \pm 0.0146s$, with a confidence interval of 95%. It indicates that, regarding the settling time, the proposed countermeasure is less efficient than the non-switching controller.

It is worth mentioning that Fig. 12 exemplifies the behavior of the proposed countermeasure and compare its performance with the performance of an NCS with a non-switching controller. From this figure, it is possible to observe a behavioral profile that allows the evaluation of characteristics such as overshoot, settling time and stability. Regarding the latter, the stability of systems based on the average dwell time technique can be verified by the theory proposed in [38], which demonstrates the feasibility of the proposed countermeasure in terms of stability.

Note in Fig. 12 that the random switching rule adds to the system a variable (however, controlled and stable) behavior, which could reduce the ability of a human observer to detect slight manipulations caused by a physically covert attack. However, it is noteworthy that when an attacker designs a physically covert attack, as a premise, he/she does not aim to explore or manipulate physical behaviors that are easy to be noticed by a human observer. Instead of this, the attacker would manipulate physical behaviors that are not accurately perceived by a human observer. In this case, it is reasonable to consider that the variations caused by the switching controller will not significantly contribute for the poor perception of malicious and covert interferences that would naturally not be perceived by a human observer (even when a non-switching controller is used).

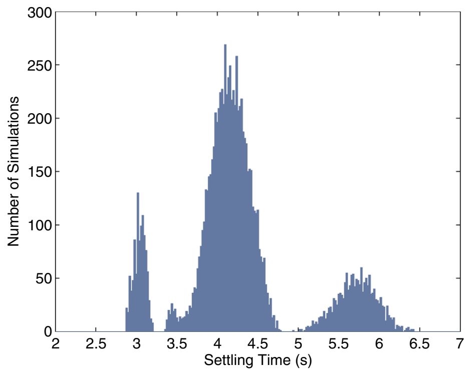


Fig. 13 Histogram of the settling time when using the proposed countermeasure

From the control perspective, the performance of the proposed countermeasure is satisfactory and, with the results presented in Section 5.1, indicates the feasibility of meeting both Objectives I and II, simultaneously. In the simulations of this section, the control provided by the switching controller presents a performance similar to the performance of the non-switching controller. The primary requirement of Objective I – i.e. stability – is met and the overshoots caused by the countermeasure, with the specified configurations, are not expressive. However, the simulations indicate an increase of the settling time of the plant, which may not be an issue, but have to be analyzed in the face of the specific process being controlled. In this sense, the results denote the existence of a tradeoff between hindering the identification attack and increasing the settling time of the system, which must be taken into account when deciding for using this countermeasure.

5.3 Impact in the controlled data injection attack

Consider that the attacker was not dissuaded by the uncertainties caused by the proposed countermeasure in the identification of the controller. Doing so, the aim of this section is to evaluate the impact of the proposed countermeasure in the design of an SD-Controlled Data injection attack.

The SD-Controlled Data Injection attack simulated in this section also aims to cause an overshoot of 50% in the rotational speed of the DC motor defined by (6), such as the attack described in Section 3.3. According to Section 3.2, to perform an SD-Controlled Data Injection attack, the attack function $M(z)$ must be designed based on the models of the plant and its controller.

As discussed in Section 4, the identification of the plant's transfer function $G(z)$ is not impacted by the use of the switching controller. So, the same $G(z)$ estimated in Section 3.3 (with a non-switching controller) is used in this section to design $M(z)$. Specifically, the coefficients used for $G(z)$ are the mean estimated coefficients shown in Table 2 for 0% of sample loss (which is the most accurate estimated model of $G(z)$). Regarding the model of the controller, as described in [12], $M(z)$ is designed considering the mean of the coefficients estimated for the switching controller. Then, performing a root locus analysis, the attacker designs the attack function (10), to make the system underdamped with a peak of rotational speed 50% higher than its steady state speed.

$$M(z) = 1.2815 \tag{10}$$

In Fig. 14, it is possible to compare the response that the attacker expects to obtain (referred as *Expected response*) with the responses that (10) actually produces (referred as *Actual responses*) when implemented in the real system. The *Expected response* represents what the attacker would obtain by simulating (10) in the forward stream of an NCS built with the models provided by the Passive System Identification attack. The *Actual responses* are represented by the highlighted area, whose bounds are drawn based on the maximum and minimum values of the output $y(t)$ of the plant, considering 100,000 simulations with (10) in the forward stream of the actual NCS. It means that, when (10) is implemented in the NCS all output signals $y(t)$ provided by the actual plant are inside this area.

It is worth mentioning that the aim of Fig. 14 is not to evaluate the stability of the proposed system after the execution of the SD-Controlled Data Injection attack (although in these simulations this system remained stable even after the execution of $M(z)$). The aim of Fig. 14 is to demonstrate that, with the proposed countermeasure, the interference produced by the attacker is not what he/she intended with the mentioned Data Injection attack. Note that, the actual responses of the plant are significantly different from the response that the attacker expects to obtain with the SD-Controlled Data Injection attack. These results are in contrast to the results achieved in the NCS with the non-switching controller, where the attack was accurate and executed exactly what was planned by the attacker, as shown in Section 3.3. With the proposed countermeasure, the maximum overshoot achieved by the plant was 10.12% (instead of the desired 50%). Notwithstanding, the highlight of these simulations is the fact that, with the proposed countermeasure, the information provided by the Passive System Identification attack is not useful to support the design covert/model-dependent attacks. This inaccurate information may lead the attacker to cause unpredictable results in the system, which may either be ineffective (not causing the desired degradation on the plant) or extreme (reducing the physical or cybernetic covertness of the attack). This analysis is consistent with the reasoning provided in Section 5.1. It demonstrates that when the NCS is endowed with the proposed countermeasure, the attacker must hesitate to launch a covert/model-dependent attack due to the inaccuracy of the Passive System Identification attack.

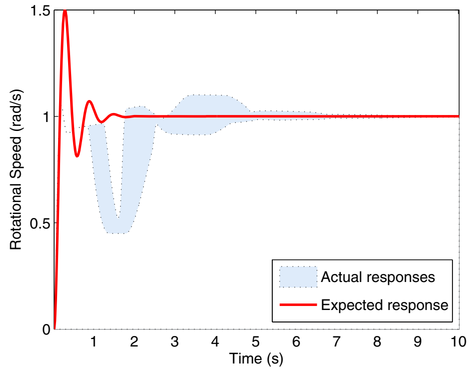


Fig. 14 Results of an SD-Controlled Data Injection attack in a system with the proposed countermeasure

Note that the countermeasure proposed in this paper aims to mitigate the Passive System Identifications attacks when the attacker is trying to obtain information about the control functions of the NCS. Consequently, it prevents the use of accurate information about these control functions in the design of a covert/model-dependent attack (such as a data injection attack in the forward stream of an NCS aiming to cause an overshoot or a steady state error). For instance, in an SD-Controlled Data Injection attack performed in the forward stream of the NCS, the attacker cannot cause a steady state error by just adding a step signal to $u(k)$, because the PI control functions will adjust the control signal to bring $y(k)$ back to 1 rad/s. Adding a ramp signal to $u(k)$ can cause a steady error in $y(k)$ for a while. However, it may not be a good strategy for the attacker, because at some time the controller and $u(k)$ will saturate, leading the plant to extreme behaviors (which is not desired if the attacker aims a physically covert attack). The alternative to cause a steady state error through the manipulation of the forward stream is to implement the attack function $M(z)$ exemplified in [12] which, to be designed, requires the knowledge about the controller and plant. Without the knowledge about the coefficients of the numerator of the PI control function, for example, the gain of $M(z)$ cannot be adjusted to cause the exact steady deviation of $y(k)$ that the attacker intends to cause. This makes the attack described in [12] model-dependent and, in this case, the countermeasure herein proposed is useful to hinder the attacker from obtaining the knowledge about the control functions of the NCS. On the other hand, in a system with an unitary feedback, it is possible to manipulate the steady state error of the plant by injecting data in the feedback stream, even when the attacker does not know the models of the plant and the controller. In this case, the manipulation of $y(k)$ can be interpreted as the direct manipulation of set point $r(k)$, which determines the steady state of the system. This attack, performed in the feedback stream is an example of data injection attack that is not model-dependent and, thus, should be mitigated by an additional countermeasure (complementary to the countermeasure proposed in this paper).

6 Conclusion

In this work, a randomly switching controller is proposed as a countermeasure for the Passive System Identification attack [12], in case of failure of other conventional security mechanisms – such as encryption, network segmentation and firewall policies. The simulations demonstrate that this countermeasure is capable to mitigate the mentioned attack, making the model obtained by the attacker imprecise and ambiguous. At the same time, the simulations demonstrate that the performance of the proposed countermeasure is satisfactory from the control perspective. Considering the control aspects, in general, the proposed countermeasure presents a performance similar to the performance of a non-switching controller, with an increase in the system's settling time. Therefore, when deciding for using this countermeasure, it must be considered the existence of a tradeoff between mitigate the identification attack and increase the settling time of the system – which, depending on the plant, is not necessarily a drawback.

As future work, we plan to evaluate the performance of this countermeasure when mitigating other system identification attacks/algorithms. Also, we encourage the development of a heuristic or an analytical method capable to provide control functions and switching rules that maximize the performance of the countermeasure in both mentioned objectives: comply with the plant's control requirements; and hinder the identification process.

Endnotes

[1] de Sa et al. [12] is an extended version of [13].

[2] The Passive System Identification attack was originally referred, in [12], as System Identification attack. However, with the introduction of the Active System Identification attack in [17], its designation was reviewed to Passive System Identification attack, in order to evince the differences between the two attacks.

Acknowledgements
This research was partially supported by the Brazilian research agencies CNPq and FAPERJ.

Authors' contributions
All authors contributed in all stages of this work, as well as read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details
[1]Institute of Mathematics/NCE, Federal University of Rio de Janeiro, Av. Athos da Silveira Ramos, 274, 68.530 Rio de Janeiro, Brazil. [2]Admiral Wandenkolk Instruction Center, Brazilian Navy, Enxadas Island, Guanabara Bay, Rio de Janeiro, Brazil. [3]National Institute of Metrology, Quality and Technology, Av. Nossa Senhora das Graças, 50, Rio de Janeiro, Brazil. [4]Rio de Janeiro Federal Center for Technological Education, Av. Maracanã, 229, Rio de Janeiro, Brazil.

Received: 19 September 2017 Accepted: 6 December 2017
Published online: 05 February 2018

References

1. Tipsuwan Y, Chow MY, Vanijjirattikhan R. An implementation of a networked pi controller over ip network. In: Industrial Electronics Society, 2003. IECON'03. The 29th Annual Conference of the IEEE. Roanoke: IEEE. 2003. p. 2805–810.
2. Gupta RA, Chow MY. Networked control system: overview and research trends. Ind Electron IEEE Trans. 2010;57(7):2527–35.
3. Zhang L, Xie L, Li W, Wang Z. Security solutions for networked control systems based on des algorithm and improved grey prediction model. Int J Comput Netw Inf Secur. 2013;6(1):78.
4. Farooqui AA, Zaidi SSH, Memon AY, Qazi S. Cyber security backdrop: A scada testbed. In: Computing, Communications and IT Applications Conference (ComComAp). Beijing: IEEE. 2014. p. 98–103.
5. Chow MY, Tipsuwan Y. Network-based control systems: a tutorial. In: Industrial Electronics Society, 2001. IECON'01. The 27th Annual Conference of the IEEE. Denver: IEEE. 2001. p. 1593–1602.
6. Amin S, Litrico X, Sastry S, Bayen AM. Cyber security of water scada systems part i: analysis and experimentation of stealthy deception attacks. IEEE Trans Control Syst Technol. 2013;21(5):1963–70.
7. Dasgupta S, Routh A, Banerjee S, Agilageswari K, Balasubramanian R, Bhandarkar S, Chattopadhyay S, Kumar M, Gupta A. Networked control of a large pressurized heavy water reactor (phwr) with discrete proportional-integral-derivative (pid) controllers. IEEE Trans Nucl Sci. 2013;60(5):3879–88.
8. Ferrara A, Sacone S, Siri S. Model-based event-triggered control for freeway traffic systems. In: Event-based Control, Communication, and Signal Processing (EBCCSP), 2015 International Conference On. Krakow: IEEE. 2015. p. 1–6.
9. Singh R, Kuchhal P, Choudhury S, Gehlot A. Wireless controlled intelligent heating system using hpso. Procedia Comput Sci. 2015;48:600–5.
10. Xia YQ, Gao YL, Yan LP, Fu MY. Recent progress in networked control systems - a survey. Int J Autom Comput. 2015;12(4):343–67.
11. Smith RS. Covert misappropriation of networked control systems: Presenting a feedback structure. Control Syst IEEE. 2015;35(1):82–92.
12. de Sa AO, da Costa Carmo LFR, Machado RCS. Covert attacks in cyber-physical control systems. IEEE Trans Ind Inform. 2017;13(4):1641–51. doi:10.1109/TII.2017.2676005.
13. de Sa AO, da Costa Carmo LFR, Machado RCS. Ataques furtivos em sistemas de controle físicos cibernéticos. In: Anais do XVI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg). Niterói, Rio de Janeiro: SBC. 2016. p. 128–41.
14. Stouffer K, Pillitteri V, Lightman S, Abrams M, Hahn A. Nist special publication 800-82, revision 2: Guide to industrial control systems (ics) security. Gaithersburg: National Institute of Standards and Technology; 2015.
15. Pang ZH, Liu GP. Design and implementation of secure networked predictive control systems under deception attacks. IEEE Trans Control Syst Technol. 2012;20(5):1334–42.
16. Langner R. Stuxnet: Dissecting a cyberwarfare weapon. Secur Priv IEEE. 2011;9(3):49–51.
17. de Sa AO, da Costa Carmo LFR, Machado RCS. Bio-inspired active attack for identification of networked control systems. In: 10th EAI International Conference on Bio-inspired Information and Communications Technologies (BICT). New Jersey: ACM; 2017. p. 1–8. doi:10.4108/eai.22-3-2017.152407.
18. Long M, Wu CH, Hung JY. Denial of service attacks on network-based control systems: impact and mitigation. Ind Inform IEEE Trans. 2005;1(2): 85–96.
19. Snoeren AC, Partridge C, Sanchez LA, Jones CE, Tchakountio F, Schwartz B, Kent ST, Strayer WT. Single-packet ip traceback. IEEE/ACM Trans Networking (ToN). 2002;10(6):721–34.
20. Teixeira A, Shames I, Sandberg H, Johansson KH. A secure control framework for resource-limited adversaries. Automatica. 2015;51:135–48.

21. Smith R. A decoupled feedback structure for covertly appropriating networked control systems. In: Proceedings of the 18th IFAC World Congress 2011. Milano: IFAC-PapersOnLine; 2011.
22. Civicioglu P. Backtracking search optimization algorithm for numerical optimization problems. Appl Math Comput. 2013;219(15):8121–44.
23. Khatri S, Sharma P, Chaudhary P, Bijalwan A. A taxonomy of physical layer attacks in manet. Int J Comput Appl. 2015;117(22):6–11.
24. Zou Y, Wang G. Intercept behavior analysis of industrial wireless sensor networks in the presence of eavesdropping attack. IEEE Trans Ind Inform. 2016;12(2):780–7.
25. Stallings W. Cryptography and Network Security: Principles and Practices. New Jersey: Pearson Education India; 2006.
26. El-Sharkawi M, Huang C. Variable structure tracking of dc motor for high performance applications. Energy Convers IEEE Trans. 1989;4(4):643–50.
27. Tran T, Ha QP, Nguyen HT. Robust non-overshoot time responses using cascade sliding mode-pid control. J Adv Comput Intell Intel Inform. 2007;11:1224–1231.
28. Nishida S. Advancement of motion psychophysics: review 2001–2010. J Vis. 2011;11(5):11–11.
29. Blair CD, Goold J, Killebrew K, Caplovitz GP. Form features provide a cue to the angular velocity of rotating objects. J Exp Psychol Hum Percept Perform. 2014;40(1):116.
30. Skafidas E, Evans RJ, Savkin AV, Petersen IR. Stability results for switched controller systems. Automatica. 1999;35(4):553–64.
31. Liberzon D, Morse AS. Basic problems in stability and design of switched systems. IEEE Control Syst. 1999;19(5):59–70.
32. Safaei FRP, Ghiocel SG, Hespanha JP, Chow JH. Stability of an adaptive switched controller for power system oscillation damping using remote synchrophasor signals. In: Decision and Control (CDC), 2014 IEEE 53rd Annual Conference On. Los Angeles: IEEE. 2014. p. 1695–1700.
33. Ferrara A, Sacone S, Siri S. A switched ramp-metering controller for freeway traffic systems. IFAC-PapersOnLine. 2015;48(27):105–10.
34. Wang J. Identification of switched linear systems. PhD thesis. 2013.
35. Lin H, Antsaklis PJ. Stability and stabilizability of switched linear systems: a survey of recent results. IEEE Trans Autom Control. 2009;54(2):308–22.
36. Morse AS. Supervisory control of families of linear set-point controllers-part i. exact matching. IEEE Trans Autom Control. 1996;41(10): 1413–31.
37. Hespanha JP, Morse AS. Stability of switched systems with average dwell-time. In: Decision and Control, 1999. Proceedings of the 38th IEEE Conference On. Phoenix: IEEE. 1999. p. 2655–660.
38. Zhai G, Hu B, Yasuda K, Michel AN. Qualitative analysis of discrete-time switched systems. In: American Control Conference, 2002. Proceedings of the 2002. Anchorage: IEEE. 2002. p. 1880–1885.

APPENDIX D

Ataques Furtivos em Sistemas de Controle Físicos Cibernéticos

Alan Oliveira de Sá[1,2], **Luiz F. Rust da Costa Carmo**[1,3], **Raphael C. S. Machado**[3]

[1]Programa de Pós-Graduação em Informática - Instituto Tércio Pacitti / IM,
Universidade Federal do Rio de Janeiro, 21.941-901, RJ – Brasil

[2]Centro de Instrução Almirante Wandenkolk – Marinha do Brasil,
Ilha das Enxadas, Baía de Guanabara – Rio de Janeiro – RJ – Brasil

[3]Instituto Nacional de Metrologia, Qualidade e Tecnologia (Inmetro)
Av. Nossa Senhora das Graças, 50, Xerém, Duque de Caxias, 25.250-020, RJ – Brasil

alan.oliveira.sa@gmail.com, {lfrust,rcmachado}@inmetro.gov.br

Abstract. *The advantages of using communication networks to interconnect controllers and physical plants motivate the increasing number of Networked Control Systems, in industrial and critical infrastructure facilities. However, this integration also exposes such control systems to new threats, typical of the cyber domain. In this context, studies have been conduced, aiming to explore vulnerabilities and propose security solutions for cyber-physical systems. In this paper, it is proposed a covert attack for system degradation, which is planned based on the intelligence gathered by another attack, herein proposed, referred as System Identification attack. The simulation results demonstrate that the joint operation of the two attacks is capable to affect, in a covert and accurate way, the physical behavior of a system.*

Resumo. *As vantagens do uso de redes de comunicação para interconectar controladores e plantas físicas tem motivado o crescente número de Sistemas de Controle em Rede, na indústria e em infraestruturas críticas. Entretanto, esta integração expõe tais sistemas a novas ameaças, típicas do domínio cibernético. Neste contexto, estudos têm sido realizados com o objetivo de explorar as vulnerabilidades e propor soluções de segurança para sistemas físico-cibernéticos. Neste artigo é proposto um ataque furtivo de degradação de serviço o qual é planejado com base nos dados colhidos por um outro ataque, ora proposto, denominado de System Identification. Os resultados de simulação demonstram que a operação conjunta dos dois ataques é capaz de afetar de forma furtiva e acurada o comportamento físico de um sistema.*

1. Introdução

A integração de sistemas usados para controlar processos físicos por meio de redes de comunicação visa atribuir a tais sistemas melhores capacidades operacionais e gerenciais, bem como reduzir custos. Em face destas vantagens, existe a tendência de um crescente número de processos industriais e sistemas de infraestruturas críticas controlados por Sistemas de Controle em Rede, ou *Networked Control Systems* (NCS) [Farooqui et al. 2014], também referidos como *Network-Based Control Systems* (NBCS) [Long et al. 2005]. Um NCS, conforme apresentado na Figura 1, consiste de uma planta física, descrita por uma função de transferência $G(z)$, um controlador, o qual executa uma função de controle

$C(z)$, e uma rede de comunicação que interconecta ambos os dispositivos para a transmissão de sinais de controle e de realimentação. Os sinais de controle são transmitidos do controlador para os atuadores da planta. Os sinais de realimentação são transmitidos dos sensores da planta para o controlador.

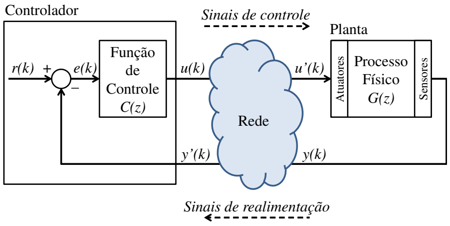


Figura 1. Sistema de Controle em Rede (ou NCS)

Ao mesmo tempo em que traz uma série de vantagens, a integração de controladores e plantas físicas em malha fechada por meio de redes de comunicação também expõe tais sistemas a novas ameaças, típicas do domínio cibernético. Neste contexto, estudos vêm sendo realizados, com o objetivo de caracterizar vulnerabilidades e propor soluções de segurança em NCSs.

Uma possível forma de atacar um NCS se dá pela intervenção em seu *software*, *i.e.* por meio de alterações na configuração ou mesmo no código executado pelo controlador, seguindo estratégia similar àquela utilizada pelo *worm* Stuxnet [Langner 2011]. Outra maneira possível para um atacante afetar um NCS é por meio de interferências no seu processo de comunicação. Basicamente, um atacante pode interferir nos sinais de controle e/ou de realimentação de três diferentes modos: induzindo *jitter* (atrasos variáveis), causando a perda de pacotes de dados, ou mesmo injetando dados falsos na comunicação.

No presente trabalho, desenvolvemos um ataque onde são injetados dados falsos no processo de comunicação de um NCS, demonstrando a possibilidade de degradação do serviço realizado por uma planta por meio de alterações sutis em seu comportamento físico. Esta intervenção tem por objetivo reduzir a eficiência da planta ou mesmo lhe causar danos em médio/longo prazo. Cabe ressaltar que uma intervenção descontrolada no NCS pode levar a uma avaria imediata da planta, ou mesmo causar alterações de grande proporção em seu funcionamento, o que pode resultar na descoberta do ataque e no eventual insucesso da operação. Sendo assim, as alterações impelidas pelo ataque ora proposto são dimensionadas para que a mudança de comportamento da planta seja fisicamente de difícil percepção, motivo pelo qual classificamos o ataque como fisicamente furtivo.

Para garantir que o ataque a um NCS seja fisicamente furtivo, o atacante deve planejar sua ofensiva com base em um conhecimento acurado sobre a dinâmica do sistema, caso contrário, as consequências do ataque podem ser imprevisíveis. Uma forma de adquirir tal conhecimento é por meio de operações de inteligência convencionais, desempenhadas para colher informações sobre o projeto e a dinâmica do NCS. Outra forma de obter informações sobre o sistema a ser atacado é por meio de o que classificamos neste trabalho como ataques de *Cyber-Physical Intelligence*. Neste sentido, também propomos no presente artigo um ataque de identificação de sistemas, ou *System Identification*, que visa obter informações sobre a função de transferência $G(z)$ da planta e da função de controle $C(z)$ do controlador. Este ataque é baseado no Algoritmo de Busca por Retrocesso, ou *Backtracking Search Optimization algorithm* (BSA) [Civicioglu 2013]. Note que, tanto o ataque de degradação de serviço por meio da injeção de dados, quanto o ataque *System Identification* requerem acesso aos sinais transmitidos no NCS, o que pode ser dificultado por técnicas de criptografia. Entretanto, não se pode negligenciar a possibilidade de acesso a tais dados por meio de ataques de criptoanálise ou mesmo de força bruta.

O presente trabalho motivou a formalização de uma série de conceitos relacionados a furtividade e inteligência no contexto da segurança físico-cibernética. Sendo assim, uma contribuição complementar do artigo é a proposição de uma nomenclatura que abarque toda uma nova classe de ataques aos sistemas físico-cibernéticos. A taxonomia proposta estabelece uma nova abordagem quanto à furtividade de ataques a sistemas físico-cibernéticos, os quais devem ser analisados sob dois aspectos simultaneamente: o aspecto físico e o aspecto cibernético.

É digno de nota que o objetivo deste trabalho não é facilitar ataques furtivos de degradação de serviço em sistemas de controle físico-cibernéticos. O objetivo deste trabalho é demonstrar o grau de acurácia que pode ser obtido neste tipo de ataque, sobretudo quando apoiado por ataques de *System Identification*, e, portanto, encorajar a pesquisa de contramedidas para tais ataques. O restante do artigo é organizado da seguinte forma: Primeiramente, na Seção 2, são apresentados alguns trabalhos relacionados. Em seguida, na Seção 3, é proposta uma taxonomia referente aos ataques físico-cibernéticos em malhas de controle de NCSs. Na Seção 4, é feita a descrição de um ataque do tipo *System Identification*. Na Seção 5, é definido um ataque furtivo de degradação de serviço. Na Seção 6, são apresentados os resultados obtidos em simulações de ataques furtivos de degradação de serviço, apoiados por ataques *System Identification*. Finalmente, na Seção 7, são apresentadas algumas conclusões e possibilidades de trabalhos futuros.

2. Trabalhos Relacionados

A possibilidade de ataques físico-cibernéticos se tornou uma realidade após o lançamento do *worm* Stuxnet [Langner 2011] e tem motivado pesquisas concernentes à segurança de NCSs. Nesta seção são apresentados alguns trabalhos relacionados ao assunto.

Em [Long et al. 2005] os autores propõem dois modelos de fila para avaliar o impacto do *jitter* e da perda de pacotes em um NCS sob ataque. O ataque não é planejado com base em um conhecimento prévio sobre os modelos do controlador e da planta. Sendo assim, para afetar o comportamento físico dos sistema, o atacante inunda a rede com um tráfego adicional, causando *jitter* e perda de pacotes de forma arbitrária. Nesta tática, o excesso de pacotes na rede pode reduzir a furtividade do ataque, permitindo a adoção de contramedidas tais como a filtragem de pacotes ou o bloqueio do tráfego malicioso na sua origem [Long et al. 2005]. Adicionalmente, a ação arbitrária sobre um modelo desconhecido pode levar o sistema a comportamentos físicos extremos, o que não é desejável se for almejado um ataque furtivo.

Em [Farooqui et al. 2014], os autores apresentam uma plataforma de testes para sistemas SCADA (*Supervisory Control and Data Acquisition*). Os mesmos demonstram um ataque onde são enviados dados falsos para o controlador e para o atuador do NCS. No artigo, os dados falsos injetados durante a comunicação têm valores randômicos e visam fazer com que um motor DC perca a sua estabilidade. Este tipo de ataque não demanda um conhecimento prévio sobre o NCS. Em contrapartida, o efeito físico desejado e a furtividade não podem ser garantidos em virtude das consequências imprevisíveis que podem surgir da aplicação de sinais aleatórios em um sistema cujo modelo é desconhecido.

Mais recentemente, em [Teixeira et al. 2015], os autores fornecem um quadro geral contendo a análise de uma grande variedade de métodos de ataque em NCSs. Em sua classificação, os mesmos estabelecem que ataques furtivos em NCSs requerem um alto nível de conhecimento sobre o sistema atacado. Exemplos de ataques furtivos são apresentados em [Smith 2011, Smith 2015]. Nestes trabalhos os ataques são desempenhados por um *man-in-the-middle* (MitM), onde o atacante necessita injetar dados tanto no enlace de controle quanto no de realimentação, bem como conhecer o modelo da planta que está sendo controlada. A furtividade destes ataques, que depende da diferença entre o modelo real da planta e o modelo utilizado pelo atacante, é analisada do ponto de vista dos sinais que chegam para o controlador, sem abordar se os efeitos físicos causados na planta são perceptíveis, ou se são furtivos perante um observador humano.

Nos trabalhos [Teixeira et al. 2015, Smith 2011, Smith 2015], onde é requerido um conhecimento sobre o modelo do NCS atacado, não é descrito como este este conhecimento é obtido pelo atacante. Considera-se apenas que o modelo é previamente conhecido para subsidiar o planejamento do ataque. A ação conjunta, ora proposta, de um ataque furtivo de degradação de serviço, apoiado por um ataque *System Identification*, visa preencher este hiato, demonstrando como os dados do NCS podem ser obtidos e como um ataque furtivo pode se beneficiar disto. A Tabela 1 apresenta uma síntese das características dos ataques apresentados nesta seção.

Tabela 1. Síntese dos ataques mencionados

Ataque	Método de ataque	Conhecimento sobre o modelo	Como o modelo é obtido
Long, *et al.* [Long et al. 2005]	Indução de *jitter* e perda de pacotes	Nenhum	N/A
Stuxnet *worm* [Langner 2011]	Modificações no código do PLC	Sim	Experimentos em um sistema real
Farooqui, *et al.* [Farooqui et al. 2014]	Injeção de dados	Nenhum	N/A
Smith [Smith 2011, Smith 2015]	Injeção de dados	Sim	Não descrito
Teixeira [Teixeira et al. 2015]	Perda de pacotes	Nenhum	N/A
	Injeção de dados	Sim	Não descrito

3. Taxonomia

Nesta Seção é apresentada uma taxonomia relativa aos possíveis ataques a sistemas de controle físico-cibernéticos. Na Seção 3.1, os ataques são brevemente descritos e classificados de acordo com a forma como agem no NCS. Na Seção 3.2, é proposta uma nova abordagem para a análise da furtividade de ataques à sistemas físico-cibernéticos.

3.1. Classificação dos ataques

Ataques a sistemas físico-cibernéticos podem atuar tanto nos seus dispositivos – *i.e.* no controlador, atuadores e sensores da planta – quanto em seus sistemas de comunicação, afetando os sinais de controle e de realimentação. Como premissa, devemos considerar que o *serviço* que se pretende atacar/proteger em tal sistema é o trabalho executado pelo processo físico, controlado por um NCS.

Considerando a definição supracitada de serviço em NCSs, os ataques podem ser classificados em três categorias distintas, como apresentado na Figura 2:

- *Denial-of-Service* (DoS) [Hussain et al. 2003]: em um NCS, os ataques DoS compreendem todos os tipos de ataques físico-cibernéticos que neguem a operação do processo físico, interrompendo a execução do serviço que a planta controlada se propõe a fazer. O ataque resulta, por exemplo, em comportamentos que podem desligar a planta ou mesmo destruí-la em um curto prazo.
- *Service Degradation* (SD): os ataques do tipo SD consistem em intervenções maliciosas que são executadas na malha de controle visando reduzir a eficiência do serviço, *i.e.* a eficiência do processo físico, ou mesmo reduzir o tempo médio entre falhas, ou *mean time between failure* (MTBF), da planta em médio/longo prazo.
- *Cyber-physical Intelligence* (CPI): o conceito de *Cyber-physical Intelligence*, aqui proposto, é diferente do conceito onde sistemas físico-cibernéticos são integrados com sistemas inteligentes [Ramos et al. 2011]. Na presente taxonomia, os ataques do tipo CPI compreendem as ações que são desempenhadas na malha de controle do NCS com o objetivo de colher informações sobre a operação do sistema e/ou sobre o seu projeto. Estes ataques têm por objetivo adquirir as informações necessárias para o planejamento de ataques furtivos e controlados, ou mesmo para subsidiar ações de *replay* [Langner 2011].

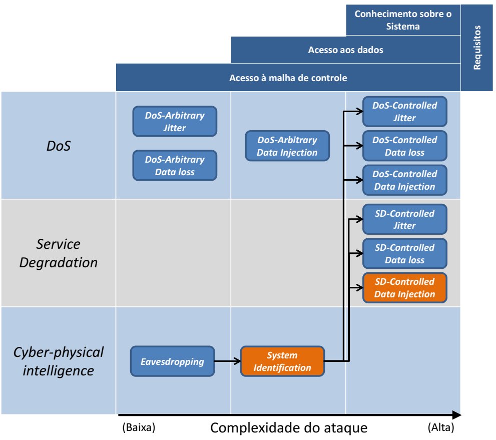


Figura 2. Classificação e requisitos dos ataques físico-cibernéticos atuantes na malha de controle de um NCS.

Na Figura 2, são apresentados seis tipos de ataques DoS, bem como os seus respectivos requisitos. Destes seis tipos de ataque, os menos complexos são os arbitrários:

- *DoS-Arbitrary Jitter*: neste tipo de ataque, o atraso dos sinais de controle e /ou realimentação é alterado arbitrariamente, sem um conhecimento prévio do modelo do NCS, com o objetivo de levar o sistema a uma instabilidade ou a uma condição que cause a interrupção do processo físico. Este ataque requer somente o acesso à malha de controle, uma vez que o mesmo pode se dar pelo simples consumo de recursos do sistema, tal como a banda dos enlaces de comunicação, ou mesmo recursos computacionais dos equipamentos que fazem parte da malha de controle.

- *DoS-Arbitrary Data Loss*: neste tipo de ataque, o atacante impede que os dados cheguem aos atuadores e/ou controladores. O atacante efetua um *jamming* arbitrário nos sinais de comunicação, sem um conhecimento prévio do modelo do NCS, levando o sistema à instabilidade ou a uma condição que cause a interrupção do processo físico. Cabe ressaltar que alguns ataques do tipo *DoS-Arbitrary Jitter* podem evoluir para um ataque *DoS-Arbitrary Data Loss*, caso atrasos de maior proporção venham a causar a perda de pacotes. Assim como no ataque *DoS-Arbitrary Jitter*, este ataque só requer o acesso à malha de controle do NCS.
- *DoS-Arbitrary Data Injection*: nestes ataques, o atacante envia dados falsos e arbitrários ao controlador, como se estes tivessem sido enviados pelos sensores, e/ou para os atuadores, como se tivessem sido enviados pelo controlador. Os dados são injetados na malha de controle do NCS sem o conhecimento prévio de seu modelo. Este ataque é mais complexo que os ataques *DoS-Arbitrary Jitter* e *DoS-Arbitrary Data Loss*, uma vez que requer o acesso aos dados que fluem na malha de controle do NCS.

Os ataques do tipo *DoS-Controlled* – *DoS-Controlled Jitter*, *DoS-Controlled Data Loss* e *DoS-Controlled Data Injection* – apresentados na Figura 2, interferem na malha de controle do NCS da mesma forma que seus respectivos ataques *DoS-Arbitrary*. A diferença entre um ataque *DoS-Controlled* e um ataque *DoS-Arbitrary* é que, no primeiro, a interferência causada pelo atacante é precisamente planejada e executada, visando alcançar com exatidão o comportamento desejado que leva o sistema à interrupção do serviço físico, de uma forma mais eficiente. Assim, para alcançar tal eficiência, um ataque *DoS-Controlled* requer um conhecimento acurado do modelo do NCS, *i.e.* das funções de transferência da planta e do controlador, as quais devem ser analisadas para o planejamento do ataque.

Referente aos ataques SD, devemos considerar três diferentes tipos de ataque – *SD-Controlled Jitter*, *SD-Controlled Data Loss* e *SD-Controlled Data Injection* – conforme apresentado na Figura 2. A diferença entre um ataque *SD-Controlled* e um ataque *DoS-Controlled* é que o primeiro não tem a intenção de interromper o processo físico em um curto prazo. O ataque visa manter o processo funcionando com a eficiência reduzida ou, por vezes, causar a deterioração física e gradual dos dispositivos controlados. Para que isto ocorra, os ataques *SD-Controlled* requerem um conhecimento prévio e acurado sobre o NCS. Caso contrário o ataque pode, por razões não previstas, evoluir para um ataque DoS, causando a interrupção do processo físico.

O conhecimento sobre o sistema, requerido tanto nos ataques *DoS-Controlled* e *SD-Controlled*, pode ser obtido por meio de ataques CPI, conforme apresentado na Figura 2. O primeiro, e mais simples, ataque CPI é o *eavesdropping* [Khatri et al. 2015], que consiste em simplesmente capturar os sinais de controle e de realimentação transmitidos. O segundo ataque CPI, proposto neste artigo, é o *System Identification*, o qual visa obter informações sobre a função de transferência da planta e a função de controle do controlador por meio da análise dos sinais que trafegam na rede. Os ataques CPI por si só não impactam no funcionamento do NCS, mas são uma poderosa ferramenta para planejar ataques *DoS-Controlled* e *SD-Controlled* eficientes.

3.2. Furtividade Cibernética vs. Física

A furtividade de um ataque corresponde à sua capacidade de não ser percebido ou detectado. No caso de ataques físico-cibernéticos em NCSs, a furtividade deve ser analisada simultaneamente em dois domínios diferentes: o domínio cibernético; e o domínio físico. Neste sentido, é apresentada nesta seção a definição de o que é um ataque *ciberneticamente furtivo* e o que é um ataque *fisicamente furtivo*:

- Ataques ciberneticamente furtivos: são ataques que têm baixa probabilidade de serem detectados por algoritmos que monitoram os softwares, a comunicação e os dados do sistema, ou por sistemas que monitoram a dinâmica da planta.
- Ataques fisicamente furtivos: são ataques que causam efeitos físicos que não são facilmente percebidos ou identificados por um observador humano. O ataque modifica sutilmente alguns comportamentos do sistema de forma a afetar fisicamente a planta, mas o efeito não é facilmente percebido ou, eventualmente, pode ser entendido como uma consequência cuja causa seja outra, diferente de um ataque.

4. Ataque de Identificação de Sistema

O ataque de Identificação de Sistemas, ou *System identification*, aqui apresentado visa estimar os coeficientes da função de transferência da planta $G(z)$ e da função de controle $C(z)$ do controlador. Ambas as funções são de sistemas Lineares e Invariantes no Tempo (LIT). O ataque usa o Algoritmo de Busca por Retrocesso, ou *Backtracking Search Algorithm* (BSA), proposto em [Civicioglu 2013] e resumidamente descrito em [de Sá et al. 2016], como metaheurística para minimizar a função de aptidão apresentada nesta Seção.

O BSA é um algoritmo evolucionário que utiliza informações obtidas por gerações – ou iterações – passadas para buscar soluções em problemas de otimização. O algoritmo possui dois parâmetros que são empiricamente ajustados: o tamanho da sua população P; e η, descrito em [de Sá et al. 2016], que estabelece a amplitude do deslocamento dos indivíduos de P. O parâmetro η deve ser ajustado visando atribuir ao algoritmo tanto uma boa capacidade exploração, quanto de refinamento da busca.

Se a entrada $i(k)$ e a saída $o(k)$ de um dispositivo real do NCS são conhecidas, seu modelo interno pode ser inferido aplicando a entrada conhecida $i(k)$ em um modelo estimado, que deve ser ajustado até que a sua saída estimada $\hat{o}(k)$ convirja para $o(k)$. Neste sentido, o BSA é usado para ajustar iterativamente o modelo estimado, minimizando uma função de aptidão específica, até que o modelo estimado convirja para o modelo real do dispositivo do NCS, o qual pode ser um controlador ou uma planta.

Para estabelecer a função de aptidão, devemos primeiramente considerar o sistema LIT genérico, cuja função de transferência $Q(z)$ pode ser representada por (1):

$$Q(z) = \frac{O(z)}{I(z)} = \frac{a_n z^n + a_{n-1} z^{n-1} + ... + a_1 z^1 + a_0}{z^m + b_{m-1} z^{m-1} + ... + b_1 z^1 + b_0}, \tag{1}$$

onde $I(z)$ é a entrada do sistema, $O(z)$ é a sua saída, n e m correspondem a ordem do numerador e do denominador, respectivamente, e $[a_n, a_{n-1}, ...a_1, a_0]$ e $[b_{m-1}, b_{m-2}, ...b_1, b_0]$ são os coeficientes do numerador e do denominador, respectivamente, os quais pretende-se estimar com o presente algoritmo de Identificação de Sistemas. Consideremos ainda que $i(k)$ e $o(k)$ representam as amostras da entrada e da saída do sistema, respectivamente, onde $I(z) = \mathcal{Z}[i(k)]$, $O(z) = \mathcal{Z}[o(k)]$, k é o número da amostra e $\mathcal{Z}$ representa a operação da transformada Z.

Neste ataque de identificação de sistemas, $i(k)$ e $o(k)$ são primeiramente capturados por um ataque do tipo *eavesdropping* [Khatri et al. 2015], por exemplo, durante um período T. Para lidar com eventuais perdas de amostras, que podem não ser recebidas pelo atacante durante T, o algoritmo retém o valor da última amostra recebida, conforme (2), onde $x(k)$ pode ser tanto $i(k)$ quanto $o(k)$.

$$x(k) = \begin{cases} x(k-1) & \text{se a amostra } k \text{ é perdida;} \\ \\ x(k) & \text{senão.} \end{cases} \tag{2}$$

Em seguida, após capturar $i(k)$ e $o(k)$, o sinal $i(k)$ é aplicado à entrada de um modelo estimado, descrito por função de transferência cujos coeficientes $[a_{n,j}, a_{n-1,j}, ...a_{1,j}, a_{0,j}, b_{m-1,j}, b_{m-2,j}, ...b_{1,j}, b_{0,j}]$ são as coordenadas de um indivíduo j do BSA. A aplicação de $i(k)$ ao modelo estimado resulta em um sinal de saída $\hat{o}_j(k)$. Após obter $\hat{o}_j(k)$, a função de aptidão f_j do indivíduo j é calculada comparando a saída $o(k)$, capturada no dispositivo atacado, com a saída do modelo estimado $\hat{o}_j(k)$, de acordo com (3):

$$f_j = \frac{\sum\limits_{k=0}^{N} (o(k) - \hat{o}_j(k))^2}{N}, \tag{3}$$

onde N é o número de amostras que existem durante o período de monitoração T. Note que, se o atacante não perder nenhuma amostra de $i(k)$ e $o(k)$ durante T, então $\min f_j = 0$ quando $[a_{n,j}, a_{n-1,j}, ...a_{1,j}, a_{0,j}, b_{m-1,j}, b_{m-2,j}, ...b_{1,j}, b_{0,j}] = [a_n, a_{n-1}, ...a_1, a_0, b_{m-1}, b_{m-2}, ...b_1, b_0]$, *i.e.* quando o modelo estimado converge para o modelo real.

É possível estabelecer uma analogia entre este ataque de identificação de sistemas e o ataque de criptoanálise do tipo *known plaintext*, onde $i(k)$ e $o(k)$ correspondem aos textos simples e cifrado, respectivamente, o formato da função de transferência genérica $Q(z)$ corresponde ao algoritmo de criptografia e os coeficientes reais de $Q(z)$ correspondem à chave criptográfica.

5. Ataque Furtivo para Degradação do Serviço

Com base na taxonomia apresentada na Seção 3.1, o ataque descrito nesta Seção é classificado como do tipo *SD-Controlled Data Injection*. Seu propósito é reduzir o MTBF da planta e/ou reduzir a eficiência do processo físico que a mesma executa, através da inserção de dados falsos na malha de controle. Ao mesmo tempo, o atacante deseja que o ataque atenda ao requisito de ser fisicamente furtivo, *i.e.* com um efeito físico de difícil percepção por um observador humano, ou entendido como uma consequência cuja causa não seja um ataque – conforme definido na seção Seção 3.2.

Uma das maneiras de degradar um serviço físico é por meio da indução de um *overshoot* durante o regime transitório da planta. *Overshoots*, ou picos no regime transitório, podem causar estresse e, eventualmente, danos à sistemas físicos como por exemplo sistemas mecânicos, químicos e eletromecânicos [El-Sharkawi and Huang 1989, Tran et al. 2007]. Adicionalmente, por ocorrerem em curto espaço de tempo, os *overshoots* são de difícil percepção pelo observador humano. Outra forma de degradar o serviço é causar um erro estacionário constante na planta, ou seja, fazer com que a saída da mesma tenha um erro constante quando $t \to \infty$. Erros estacionários de pequena proporção, além de serem de difícil percepção pelo observador humano, podem reduzir a eficiência do processo físico e, eventualmente, estressar e danificar o sistema em médio/longo prazo.

Neste ataque, para alcançar qualquer um dos dois efeitos citados, *i.e.* um *overshoot* ou um erro estacionário constante, o atacante intervém no processo de comunicação do NCS a fim de injetar, de forma controlada, dados falsos no sistema. Para tal, o atacante atua como um MitM que executa uma função de ataque $M(z)$, conforme apresentado na Figura 3, onde $U'(z) = M(z)U(z)$, $U(z) = \mathcal{Z}[u(k)]$ e $U'(z) = \mathcal{Z}[u'(k)]$. A função $M(z)$ é projetada com base nos dados da planta e do controlador, obtidos no ataque do tipo *System Identification* descrito na Seção 4. A eficácia do ataque, portanto, depende do projeto de $M(z)$, que por sua vez depende da acurácia do ataque de *System Identification*. Cabe ressaltar que, apesar de na Figura 3 o MitM atuar nos sinais de controle, é possível, também, que o mesmo atue nos sinais de realimentação do NCS. O MitM pode ser estabelecido tanto em redes cabeadas quanto, eventualmente, em redes sem fio conforme em [Hwang et al. 2008].

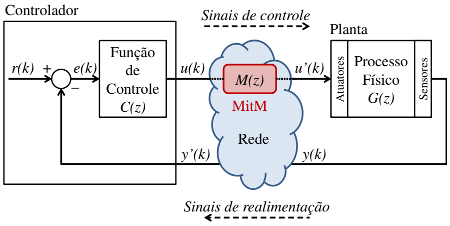


Figura 3. Ataque MitM

6. Resultados

Nesta seção são apresentados os resultados obtidos em simulações que combinam ataques do tipo *System Identification* com ataques *SD-Controlled* fisicamente furtivos. Na Seção 6.1, é apresentado o modelo do sistema atacado. Na Seção 6.2 são apresentados os resultados obtidos pelo ataque do tipo *System Identification*. Na Seção 6.3 são apresentados os resultados obtidos com simulações de ataques do tipo *SD-Controlled Data Injection*, fisicamente furtivos, planejados com base nos dados do ataque de *System Identification*.

6.1. Modelo do Sistema

O NCS atacado tem a mesma arquitetura do NCS apresentado na Figura 1, e consiste em um controlador Proporcional-Integral (PI) que controla a velocidade de rotação de um motor DC. A função de controle $C(z)$ e a função de transferência $G(z)$ do motor DC foram extraídas de [Long et al. 2005]. Tais equações são representadas por (4):

$$C(z) = \frac{c_1 z - c_2}{z - 1} \qquad\qquad G(z) = \frac{g_1 z + g_2}{z^2 - g_3 z + g_4} \tag{4}$$

onde $c_1 = 0,1701$, $c_2 = -0,1673$, $g_1 = 0,3379$, $g_2 = 0,2793$, $g_3 = -1,5462$ e $g_4 = 0,5646$. A taxa de amostragem do sistema é 50 amostras/s e o *set point* $r(k)$ é uma função degrau unitário. O atraso na rede não é considerado nestas simulações.

6.2. Resultados da Identificação do Sistema

Nesta Seção, o desempenho do algoritmo de Identificação de Sistemas é avaliado por meio um conjunto de simulações realizadas no MATLAB. A ferramenta SIMULINK foi utilizada para calcular a saída $\hat{o}_j$ dos modelos estimados, cujos coeficientes são as coordenadas de um indivíduo j do BSA.

A estrutura das equações representadas por (4) são previamente conhecidas pelo atacante, uma vez que, como premissa, este sabe que o alvo é um NCS que controla um motor DC por meio de um controlador PI. Nestas simulações, o objetivo do ataque de *System Identification* é descobrir g_1, g_2, g_3, g_4, c_1 e c_2, levando em consideração cenários em que o atacante eventualmente perde amostras durante a coleta dos sinais de controle e de realimentação.

Toda vez que o motor DC é ligado, os sinais de controle e de realimentação são capturados pelo atacante durante um período $T = 2s$. no momento em que o motor é ligado, todas as condições iniciais são consideradas 0. Os coeficientes de $G(z)$, $[g_1, g_2, g_3, g_4]$, e os coeficientes de $C(z)$, $[c_1, c_2]$, são calculados separadamente considerando que, apesar da malha fechada, $G(z)$ e $C(z)$ são funções independentes. Para estimar $[g_1, g_2, g_3, g_4]$, o atacante considera que o sinal de controle é a entrada e que o sinal de realimentação é a saída da planta. Já para estimar $[c_1, c_2]$, o atacante considera que o sinal e realimentação é a entrada e que o sinal de controle é a saída do controlador.

Para simular a perda de amostras, são consideradas quatro taxas de perda l diferentes: 0, 0,05, 0,1 e 0,2. Assim, uma amostra é perdida pelo atacante toda vez que $l < \mathcal{P}$, onde $\mathcal{P} \sim U(0, 1)$ e U é uma distribuição uniforme. Para cada taxa de perda são executadas 100 simulações diferentes.

No BSA, a população utilizada contém 100 indivíduos e η, empiricamente ajustado, é 1. Para estimar os coeficientes do controlador $[c_1, c_2]$, são executadas 600 iterações do algoritmo. Já para estimar os coeficientes da planta $[g_1, g_2, g_3, g_4]$, o número de iterações é aumentado para 800, devido ao maior número de dimensões do espaço de busca neste caso. Os limites de cada dimensão do espaço de busca são $[-10, 10]$.

A Figura 4 apresenta a média de 100 valores estimados para g_1, g_2, g_3, g_4, c_1 e c_2, com um Intervalo de Confiança (IC) de 95%, considerando diferentes taxas de perda de amostras. Os valores reais dos coeficientes de $C(z)$ e $G(z)$ também são representados na Figura 4. Note que a amplitude das escalas das Figuras 4(a), 4(b), 4(c) e 4(d) é diferente da amplitude das escalas das Figuras 4(e) e 4(f), em virtude dos menores IC de c_1 e c_2. Adicionalmente, algumas estatísticas referentes aos resultados obtidos são apresentadas na Tabela 2.

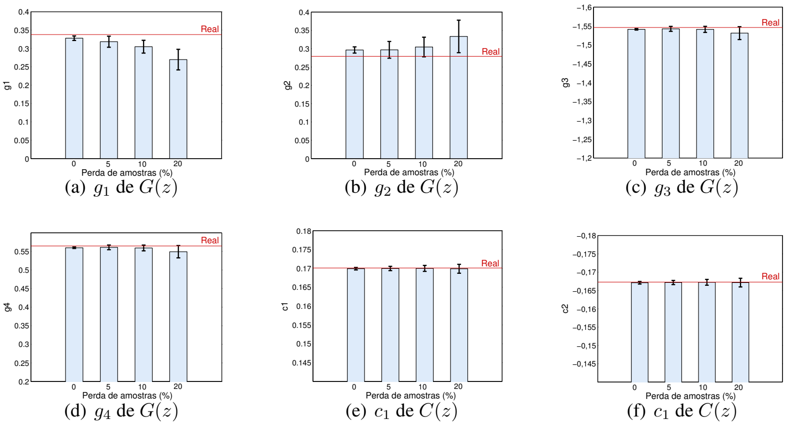
Tabela 2. Estatísticas dos resultados com diferentes perdas de amostras

Perda de amostras	Média						Desvio Padrão					
	g_1	g_2	g_3	g_4	c_1	c_2	g_1	g_2	g_3	g_4	c_1	c_2
0%	0.32793	0.29652	-1.54121	0.55983	0.16991	-0.16712	0.03097	0.04288	0.00986	0.00944	0.00167	0.00178
5%	0.31835	0.29689	-1.54251	0.56085	0.16997	-0.16719	0.07572	0.11523	0.03322	0.03194	0.00287	0.00287
10%	0.30473	0.30461	-1.54110	0.55925	0.16999	-0.16724	0.08781	0.13483	0.04076	0.03922	0.00397	0.00399
20%	0.26963	0.33352	-1.53119	0.54916	0.16989	-0.16716	0.14120	0.22378	0.08596	0.08313	0.00596	0.00598
Perda de amostras	Assimetria(*)						Curtose					
	g_1	g_2	g_3	g_4	c_1	c_2	g_1	g_2	g_3	g_4	c_1	c_2
0%	-1.21214	1.23278	1.75298	-1.73202	-0.64331	0.79458	0.18846	0.19433	0.21259	0.21218	0.15119	0.16472
5%	-2.34607	1.64875	1.35284	-1.41346	-0.42288	0.36037	0.08094	0.10527	0.09412	0.09802	0.02540	0.03118
10%	-2.52938	1.97711	1.18018	-1.26045	-0.23379	0.13377	0.16833	0.17123	0.25041	0.24811	0.24361	0.23429
20%	-3.24122	1.75186	1.68335	-1.71055	-0.40055	0.37927	0.21292	0.21127	0.25054	0.24932	0.23883	0.24441

(*) Calculado de acordo com o 2^o coeficiente de assimetria de Pearson.

De acordo com a Tabela 2 as distribuições de g_1, g_2, g_3 e g_4 possuem forte assimetria, enquanto as distribuições de c_1 e c_2 têm assimetria moderada. Em relação a curtose, as distribuições de todos os coeficientes de $G(z)$ e $C(z)$ são leptocúrticas. Entretanto, analisando a Tabela 2, não é possível identificar uma clara tendência de aumento ou diminuição de assimetria e curtose em face do aumento da perda de amostras.

Na Figura 4, é possível constatar que em todos os casos os ICs tendem a crescer com o aumento da perda de amostras. O mesmo ocorre com os desvios padrão apresentados na Tabela 2. Referente aos coeficientes de $G(z)$, a Figura 4 mostra que a diferença entre a média e o valor real de g_1, g_2, g_3 e g_4 também tende a crescer com o aumento da perda de amostras. Cabe ressaltar que o desempenho do algoritmo no cálculo de g_3 e g_4 é melhor do que no cálculo de g_1 e g_2, tanto no que diz respeito a média quanto a amplitude do IC. Atribuímos este comportamento à maior sensibilidade que a saída de $G(z)$ tem às variações de seus polos do que às variações de seus zeros. Isto significa que, neste problema, f_j cresce mais com os erros de g_3 e g_4 do que com os erros de g_1 e g_2, fazendo com que a população do BSA convirja de forma mais acurada em g_3 e g_4.



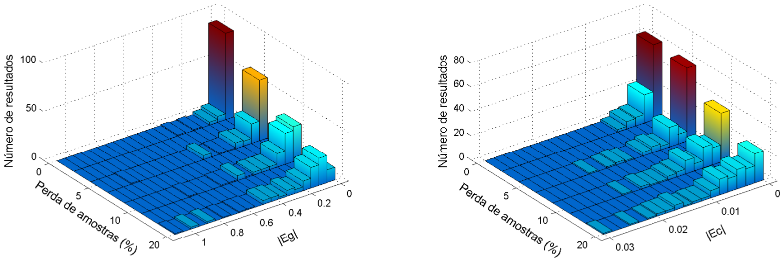
(a) g_1 de $G(z)$ (b) g_2 de $G(z)$ (c) g_3 de $G(z)$

(d) g_4 de $G(z)$ (e) c_1 de $C(z)$ (f) c_1 de $C(z)$

Figura 4. Média, com IC de 95%, dos coeficientes estimados de $G(z)$ e $C(z)$, em face de diferentes taxas de perda de amostras.

Na Figura 4 é possível também verificar que a acurácia obtida no cálculo dos coeficientes de $C(z)$ é melhor do que a acurácia dos coeficientes de $G(z)$, para todas as taxas de perda de amostras. As médias de c_1 e c_2 são mais próximas dos seus valores reais, com um menor IC. De fato, o processo de otimização é mais eficiente no cálculo dos coeficientes de $C(z)$ devido ao menor tamanho do espaço de busca, que possui apenas duas dimensões ao invés das quatro existentes no problema de $G(z)$.

Como uma forma adicional de avaliar o desempenho do algoritmo, foram calculados $|E_g| = |\mathcal{G}_r - \mathcal{G}_e|$ e $|E_c| = |\mathcal{C}_r - \mathcal{C}_e|$ que sintetizam o erro de estimativa dos coeficientes de $G(z)$ e $C(z)$, respectivamente, para cada solução encontrada. $\mathcal{G}_r$ e $\mathcal{G}_e$ são vetores contendo os coeficientes reais e estimados de $G(z)$, respectivamente. Já $\mathcal{C}_r$ e $\mathcal{C}_e$ são vetores contendo os coeficientes reais e estimados de $C(z)$, respectivamente. Os histogramas de $|E_g|$ e $|E_c|$ são apresentados na Figura 5, considerando as diferentes taxas de perda de amostras mencionadas. Os histogramas mostram graficamente que $|E_g|$ e $|E_c|$, que correspondem ao módulo do erro dos coeficientes estimados de $G(z)$ e $C(z)$, respectivamente, tendem a apresentar valores maiores à medida que a perda de amostras aumenta. Isto também pode ser confirmado pelo aumento do desvio padrão dos coeficientes de $G(z)$ e $C(z)$ apresentados na Tabela 2. Entretanto, de acordo com a Figura 5, a moda destes erros permanecem próximas de zero em todos os casos de perda avaliados.

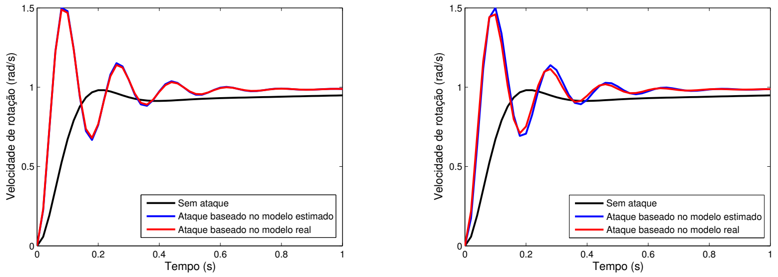


(a) Distribuição dos $|E_g|$ (b) Distribuição dos $|E_c|$

Figura 5. Histogramas de $|E_g|$ e $|E_c|$ em face das diferentes perdas de amostras

6.3. Resultados do Ataque de Degradação do Serviço

Nesta seção são apresentados os resultados obtidos em simulações de ataque do tipo *SD-Controlled Data Injection*, realizados por um MitM atuando no enlace de controle do NCS, conforme na Figura (3). Os ataques foram simulados no MATLAB, com o objetivo de avaliar a acurácia de ataques planejados com base nos resultados da Seção 6.2, obtidos pelo ataque de *System Identification*. Foram realizados dois conjuntos de ataques. O primeiro, visa causar um *overshoot* de 50% na velocidade de rotação do motor. O segundo, visa causar um erro estacionário de -10% na velocidade de rotação do motor em regime permanente.



(a) Ataque baseado nos dados obtidos sem perda de amostras

(b) Ataque baseado nos dados obtidos com 20% de perda de amostras

Figura 6. Resposta do sistema a ataques planejados com o propósito de causar um *overshoot* de 50% da velocidade de rotação do motor.

No ataque visando o *overshoot*, a função executada pelo atacante é $M(z) = \mathcal{K}_o$. Por meio da análise do lugar das raízes, traçado com base nos modelos levantados, o atacante ajusta o valor de $\mathcal{K}_o$ para que o sistema se torne subamortecido com um pico de velocidade de rotação 50% maior do que a velocidade em regime permanente. Os valores de $\mathcal{K}_o$ foram ajustados com base nas médias dos coeficientes levantados na Seção 3.2. A Tabela 3 apresenta os valores de $\mathcal{K}_o$, estimados considerando as diferentes situações de perda de amostras no ataque de *System Identification*, bem como os *overshoots* obtidos com os respectivos $\mathcal{K}_o$ no modelo real. Na Figura 6 é possível comparar a resposta do sistema sem ataque com a resposta ao ataque visando o *overshoot* de 50%. É possível verificar, ainda, que o ataque ao modelo real apresenta, no domínio do tempo, uma resposta bem próxima ao ataque projetado com base no modelo obtido pelo ataque de *System*

Identification, tanto no caso em que o sistema foi identificado com 0% de perdas, quanto no pior caso considerado, com 20% de perdas. Cabe ressaltar que todas as respostas apresentadas na Figura 6 convergem para 1 rad/s.

No ataque cujo propósito é causar um erro estacionário de -10% na velocidade de rotação do motor, o atacante executa a função (5):

$$M(z) = \frac{\mathcal{K}_{Ess}(z-1)}{z-0,94}, \tag{5}$$

onde $\mathcal{K}_{Ess}$ é ajustado com base nos dados de identificação do sistema, considerando cada condição de perda de amostras. O pólo de $M(z)$ é adicionado com o objetivo de permitir que ocorra um erro estacionário no sistema. O zero de $M(z)$ visa formatar o lugar das raízes a fim de que haja um $\mathcal{K}_{Ess}$ estável que leve o sistema a um erro estacionário de -10%. A Tabela 3 apresenta os valores de $\mathcal{K}_{Ess}$ adotados considerando as diferentes situações de perda de amostras no ataque de *System Identification*, bem como os respectivos erros estacionários alcançados no modelo real.

Tabela 3. Valores de $\mathcal{K}_o$, $\mathcal{K}_{Ess}$ e resultados obtidos com os ataques

	Perda de amostras no ataque *System Identification*			
	0 %	5 %	10 %	20 %
$\mathcal{K}_o$	4,0451	4,0745	4,0828	3,796
Overshoot no modelo real	48,90 %	49,43 %	49,57 %	45,94 %
$\mathcal{K}_{Ess}$	5,7471	5,7803	5,8140	5,8823
Erro estacionário no modelo real	-10%	-10%	$-9,9\%$	$-9,8\%$

De acordo com os dados na Tabela 3, é possível afirmar que os ataques *SD-Controlled Data Injection*, projetados com base nos dados colhidos pelo ataque *System Identification*, foram capazes de modificar de forma acurada a resposta do sistema físico, considerando todas as condições de perda avaliadas. No pior caso, *i.e.* com 20% de perda de amostras, o *overshoot* foi de $45,94\%$ e o erro estacionário foi de $-9,8\%$, bem próximos dos valores desejados de 50% e -10%, respectivamente. Tal acurácia, permite que a resposta do sistema se mantenha controlada e próxima a um comportamento pré-definido como fisicamente furtivo para o sistema em questão.

7. Conclusões

Este trabalho propõe um ataque fisicamente furtivo de degradação de serviço, cujo desempenho depende do conhecimento sobre a planta atacada e seu controlador. Para adquirir tal conhecimento, é proposto um ataque de *System Identification*, baseado no algoritmo BSA. A eficácia do ataque de *System Identification* é demonstrada e o seu desempenho é estatisticamente analisado em face de diferentes taxas de perda de amostra. Os resultados alcançados nos ataques fisicamente furtivos de degradação de serviço, dimensionados com base nos dados levantados pelo *System Identification*, demonstram o elevado grau de acurácia que pode ser obtido com a combinação dos ataques. No pior caso, *i.e.* com 20% de perda de amostras durante a identificação do sistema, o atacante foi capaz de causar na planta um *overshoot* de $45,94\%$ e um erro estacionário de $-9,8\%$, bem próximos dos valores desejados de 50% e -10%, respectivamente. Em ambas as ações fisicamente furtivas, a acurácia do ataque garante que estas não evoluam para alterações de comportamento fisicamente mais perceptíveis.

Como trabalho futuro, encorajamos a pesquisa de técnicas capazes de evitar, ou dificultar, ataques fisicamente furtivos planejados com dados obtidos por ataques *System Identification*. Neste sentido, planejamos investigar contramedidas que possam dificultar a obtenção de informações sobre os sistemas de controle físico-cibernéticos, as quais são essenciais para o planejamento de ataques furtivos e controlados.

Referências

Civicioglu, P. (2013). Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematics and Computation*, 219(15):8121–8144.

de Sá, A. O., Nedjah, N., and de Macedo Mourelle, L. (2016). Distributed efficient localization in swarm robotic systems using swarm intelligence algorithms. *Neurocomputing*, 172:322–336.

El-Sharkawi, M. and Huang, C. (1989). Variable structure tracking of dc motor for high performance applications. *Energy Conversion, IEEE Transactions on*, 4(4):643–650.

Farooqui, A. A., Zaidi, S. S. H., Memon, A. Y., and Qazi, S. (2014). Cyber security backdrop: A scada testbed. In *Computing, Communications and IT Applications Conference (ComComAp), 2014 IEEE*, pages 98–103. IEEE.

Hussain, A., Heidemann, J., and Papadopoulos, C. (2003). A framework for classifying denial of service attacks. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 99–110. ACM.

Hwang, H., Jung, G., Sohn, K., and Park, S. (2008). A study on mitm (man in the middle) vulnerability in wireless network using 802.1 x and eap. In *Information Science and Security, 2008. ICISS. International Conference on*, pages 164–170. IEEE.

Khatri, S., Sharma, P., Chaudhary, P., and Bijalwan, A. (2015). A taxonomy of physical layer attacks in manet. *International Journal of Computer Applications*, 117(22).

Langner, R. (2011). Stuxnet: Dissecting a cyberwarfare weapon. *Security & Privacy, IEEE*, 9(3):49–51.

Long, M., Wu, C.-H., and Hung, J. Y. (2005). Denial of service attacks on network-based control systems: impact and mitigation. *Industrial Informatics, IEEE Transactions on*, 1(2):85–96.

Ramos, C., Vale, Z., and Faria, L. (2011). Cyber-physical intelligence in the context of power systems. In *Future Generation Information Technology*, pages 19–29. Springer.

Smith, R. (2011). A decoupled feedback structure for covertly appropriating networked control systems. In *Proceedings of the 18th IFAC World Congress 2011*, volume 18. IFAC-PapersOnLine.

Smith, R. S. (2015). Covert misappropriation of networked control systems: Presenting a feedback structure. *Control Systems, IEEE*, 35(1):82–92.

Teixeira, A., Shames, I., Sandberg, H., and Johansson, K. H. (2015). A secure control framework for resource-limited adversaries. *Automatica*, 51:135–148.

Tran, T., Ha, Q. P., and Nguyen, H. T. (2007). Robust non-overshoot time responses using cascade sliding mode-pid control. *Journal of Advanced Computational Intelligence and Intelligent Informatics*.

APPENDIX E

Bio-inspired Active Attack for Identification of Networked Control Systems

Alan Oliveira de Sá
Institute of Mathematics/NCE,
Federal University of
Rio de Janeiro
ZIP Code: 21.941-901
CIAW, Brazilian Navy
Rio de Janeiro – RJ – Brazil
alan.oliveira.sa@gmail.com

Luiz F. R. da C. Carmo
Institute of Mathematics/NCE,
Federal University of
Rio de Janeiro
ZIP Code: 21.941-901
National Institute of Metrology,
Quality and Technology
Duque de Caxias, RJ, Brazil
lfrust@inmetro.gov.br

Raphael C. S. Machado
National Institute of Metrology,
Quality and Technology
Duque de Caxias, RJ, Brazil
rcmachado@inmetro.gov.br

ABSTRACT

The use of communication networks to interconnect controllers and physical plants in industrial and critical infrastructure facilities exposes such control systems to threats typical of the cyber domain. In this sense, studies have been done to explore vulnerabilities and propose security solutions for Networked Control System (NCS). From the point of view of the control theory, the literature indicates that stealthy and accurate cyber-physical attacks must be planned based on an accurate knowledge about the model of the NCS. However, most literature about these attacks does not indicate how such knowledge is obtained by the attacker. So, to fill this hiatus, it is proposed and evaluated in this paper an Active System Identification attack, where the attacker injects data on the NCS to learn about its model. The attack is implemented based on two bio-inspired metaheuristics, namely: Backtracking Search Optimization Algorithm (BSA); and Particle Swarm Optimization (PSO). The results indicate a better performance of the BSA-based attack, especially when the captured signals contain white Gaussian noise. The goal of this paper is to demonstrate the degree of accuracy that this attack may achieve, highlighting the potential impacts and encouraging the research of possible countermeasures.

CCS Concepts

•**Security and privacy** → **Formal security models;** Cryptanalysis and other attacks; •**Computing methodologies** → **Search methodologies; Computational control theory;**

Keywords

Security, Cyber-Physical Systems, Networked Control Systems, System Identification, Backtracking Search Algorithm, Particle Swarm Optimization

BICT'17, March 15-16, 2017, New Jersey, USA

http://dx.doi.org/xx.xxxx/xxxxxxx.xxxxxxx

1. INTRODUCTION

System identification, *i.e.* the action of building mathematical models of dynamic systems, is often used to obtain the model of physical processes aiming to subsidize the design of their respective control systems. However, it can also be considered a key step for the execution of stealth – or covert, as mentioned in [16, 17, 20] – attacks against Networked Control Systems (NCS). Indeed, to reduce the probability to be detected by algorithms that monitor the dynamics of the controlled plant, the attacker must have an accurate model of the targeted system, such as demonstrated in [16, 17, 20].

A possible strategy to obtain information about the model of the targeted system is through passive System Identification attacks, as reported in [5]. In this technique, the attacker eavesdrops the communications between the controller, actuators and sensors of the NCS until enough information is collected to determine the parameters of the plant and its control system. Such passive approach can make the system identification to last for a long time, until meaningful information transits at the eavesdropped communication line. The situation is even worse if the system is on steady state, because no meaningful information may transit through the NCS's communication links for a long time – indeed, the information content of the signals measured under steady operating conditions is often insufficient for identification purposes [22]. This attacker's constraint may be overcome by Active System Identification attacks, which, as far as we know, is not reported in the literature.

In this sense, in the present work, we propose an active attack for the identification of NCSs. Our approach was inspired by the classic active cryptanalytic attacks – chosen plaintext and chosen cypher text –, where the attacker inserts messages in the crypto-engine, in opposition to passive attacks – cyphertext-only, known plaintext –, where the attacker simply listen the communication channels and passively collects information [19].

In the attack herein proposed, a specially tailored signal is inserted by the attacker in an NCS communication channel and, by observing the behavior of the system in closed-loop, the attacker determines the parameters of its open-loop transfer function. To do so, the attacker just needs to intercept one communication channel of the NCS, where the attacker both insert the attack signal and listen the consequent system response. The knowledge of the NCS's open-loop transfer function, obtained through this attack, is useful for the design of other sophisticated attacks. For instance, if an attacker learns the open-loop transfer function of an NCS, it is possible to further design attacks capable to accurately change the transient response and/or steady state response of the plant, such as demonstrated in [5], causing, for example, stationary errors or overshoots on the plant. A stationary error may reduce the efficiency of the physical process, while overshoots may cause stress and possibly damages [6, 21] to the plant, reducing its mean time between failure (MTBF).

The present Active System Identification attack is developed based on two bio-inspired metaheuristics, whose results are analyzed and compared, namely: the Backtracking Search Optimization algorithm (BSA) [4]; and the Particle Swarm Optimization (PSO) [10]. If the attack signal $a(k)$ and the consequent response $y_a(k)$ of an NCS is known, its open-loop transfer function can be assessed by applying $a(k)$ in an estimated model, which is adjusted until its estimated output $\hat{y}_a(k)$ matches $y_a(k)$. In this sense, the BSA and the PSO are used to iteratively adjust the parameters of an estimated model, by minimizing a specific fitness function, until the estimated model converges to the actual model of the NCS. The BSA and the PSO are chosen to perform this task due to their capability to converge to good solutions, such as demonstrated in [9, 13, 23, 24, 8] specifically for control system problems.

It is worth mentioning that the Active System Identification attack herein proposed is different from the active attacks performed to identify vulnerabilities of protocols and applications within the layers of the OSI model, such as the active scanning process used to identify network services [2].The attack herein proposed aims to identify the physical model of a plant that, in an NCS, lies above the application layer of the OSI model.

The goal of this paper is to demonstrate the degree of accuracy that such attack may achieve, highlighting its potential impacts and encouraging the research of countermeasures capable to prevent or detect the execution of this kind of attack. The remainder of this paper is organized as follows. In Section 2, we review the literature on NCS attacks, with focus on the intelligence gathered to subsidize their design. In Sections 3 and 4, there are provided brief descriptions of the BSA and PSO, respectively. In Section 5, it is described the Active System Identification attack, herein proposed. In Section 6, there are presented and compared the results achieved by the proposed attack, using both metaheuristics, in simulations where the NCS is constituted by a DC motor and a proportional-integral (PI) controller. Section 7 contains our final considerations.

2. RELATED WORKS

The possibility of large impact cyber-physical attacks became unprecedentedly concrete after the launch of the Stuxnet worm [11] and has been motivating researches concerning the security of NCSs. In this section, it is presented a review of the literature related to this subject.

In [12] the authors propose two queueing models that are used to evaluate the impact of delay jitter and packet loss in an NCS under attack. The attack is not designed taking into account the models of the controller and the physical plant. Such models are unknown by the attacker. Thus, to affect the plant's behavior, the attacker arbitrarily floods the network with traffic, causing jitter and packet loss. In this method of attack, the excess of packets in the network can reduce the stealthiness of the attack, allowing the adoption of countermeasures, such as packet filtering [12] or blocking the malicious traffic on its origin [18]. Moreover, the arbitrary intervention in a system which the models are unknown may lead the plant to an extreme physical behavior, which is not desired if a stealth attack is intended.

In [7], it is presented a testbed for Supervisory Control and Data Acquisition (SCADA) using TrueTime – a MATLAB/Simulink based tool. The authors demonstrate an attack where a malicious agent transmits false signals to the controller and actuator of an NCS. The false signals are randomly generated, aiming to make a DC motor lose its stability. This kind of attack does not require a previous knowledge about the plant and controller of the NCS. The drawback is that the desired physical effect and the stealthiness of the attack can not be ensured due to the unpredictable consequences of the application of random false signals to a system which the model is not known.

A general framework for the analysis of a wide variety of attacks over NCSs is provided in [20]. The authors classify and establish the requirements for the attacks in terms of the model knowledge, disclosure and disruption resources. In their work, it is stated that covert attacks require high level of knowledge about the model of the targeted system. Exemples of covert attacks that agree with this statement are provided in [16, 17]. In these works the attacks are performed by a man-in-the-middle (MitM), where the attacker needs to know the model of the plant under attack and also inject false data in both the forward and the feedback streams. The stealthiness of the attacks described in [16, 17] is analyzed from the perspective of the signals arriving to the controller, and depends on the difference between the actual model of the plant and the model known by the attacker. In [1], it is demonstrated another stealth attack where the attacker, aware of the system's model, injects an attack signal in the NCS to steal water from the Gignac canal system located in Southern France.

Table 1: Synthesis of the related attacks

Attack	Method	System knowledge	How the knowledge is obtained
Stuxnet *worm* [11]	Modifications in the PLC code	Yes	Experiments in a real system
Long, *et al.* [12]	Inducing *jitter* and packet loss	None	N/A
Farooqui, *et al.* [7]	Data injection	None	N/A
Smith [16, 17]	Data injection	Yes	Not described
Teixeira [20]	Packet loss	None	N/A
	Data injection	Yes	Not described
Amin [1]	Data injection	Yes	Not described
SD-Controlled [5]	Data injection	Yes	Passive system identification

In [1, 16, 17, 20], where it is required a previous knowledge about the models of the NCS under attack, it is not described how this knowledge is obtained by the attacker. It is just stated that a model is previously known to subsidize the design of the attack. More recently, in [5], the authors propose a System Identification attack to fill this hiatus. They demonstrate how the data required for the design of Denial-of-Service (DoS) or Service Degradation (SD) attacks may be obtained through a passive System Identification attack. The attack proposed in [5] does not need to inject signals on the NCS to estimate its models. However, it depends on the occurrence of events, that are not controlled by the attacker, to produce signals that carry meaningful information for the system identification algorithm. The Active System Identification attack herein proposed, constitutes an alternative to the passive System Identification attacks in situations where the attacker may not wait so long for the occurrence o such meaningful signals. A synthesis of the characteristics of the attacks referred in this section is presented in Table 1.

3. BACKTRACKING SEARCH ALGORITHM

In this section, there are described the basic concepts of the BSA, in order to provide a clear comprehension regarding to the parameters of the algorithm that are adjusted for the attack. The BSA is a bio-inspired metaheuristic that searches for solutions of optimization problems using the information obtained by past generations – or iterations. According with [4], its search process is metaphorically analogous to the behavior of a social group of animals that, at random intervals returns to hunting areas previously visited for food foraging. The general, evolutionary like, structure of the BSA is shown in Algorithm 1.

Algorithm 1 BSA

```
begin
    Initialization;
    repeat
        Selection-I;
        Generate new population
            Mutation;
            Crossover;
        end
        Selection-II;
    until Stopping Condition;
end
```

At the initialization stage, the algorithm generates and evaluates the initial population $\mathcal{P}_0$ and sets the historical population $\mathcal{P}_{hist}$. The latter composes the BSA's memory.

During the first selection stage (Selection-I), the algorithm randomly determines, based on an uniform distribution U, whether the current population $\mathcal{P}$ should be kept as the new historical population, and thus replace $\mathcal{P}_{hist}$ (*i.e.* if $a < b \mid a, b \sim U(0,1)$, then $P_{hist} = P$). Subsequently, it shuffles the individuals of this population.

The mutation operator creates $\mathcal{P}_{mod}$, which is the preliminary version of the new population $\mathcal{P}_{new}$). It does so according to (1):

$$\mathcal{P}_{mod} = \mathcal{P} + \eta \cdot \Gamma(\mathcal{P}_{hist} - \mathcal{P}), \qquad (1)$$

wherein η is empirically adjusted through simulations and $\Gamma \sim N(0,1)$, with N being a normal standard distribution. Thus, $\mathcal{P}_{mod}$ is the result of the movement of $\mathcal{P}$'s individuals in the directions established by vector $(\mathcal{P}_{hist} - \mathcal{P})$.

In order to create the final version of $\mathcal{P}_{new}$, the crossover operator combines randomly, also following a uniform distribution, individuals from $\mathcal{P}_{mod}$ and others from $\mathcal{P}$.

At the second selection stage (Selection-II), the algorithm evaluates, selects elements of $\mathcal{P}_{new}$ (*i.e.* individuals obtained after mutation and crossover), which should have better fitness than those in $\mathcal{P}$ (*i.e.* individuals before applying both the operators of crossover and mutation) and replaces them in $\mathcal{P}$. Hence, $\mathcal{P}$ includes only new individuals that should have evolved. While the stopping condition has not yet been reached, the algorithm iterates. Otherwise, it returns the best solution found.

Note that the algorithm has two parameters that are empirically adjusted: the size $|\mathcal{P}|$ of its population $\mathcal{P}$; and η, that establishes the amplitude of the movements of the individuals of $\mathcal{P}$. The parameter η must be adjusted to assign to the algorithm both good exploration and exploitation capabilities. With this parameters set, the BSA is used to search for the global minimum of the fitness function described in Section 5.

4. PARTICLE SWARM OPTIMIZATION

PSO has roots in the collective behavior of social models such as bird flocking and fish schooling. A particle, *i.e.* the basic element of the algorithm, represents a possible solution of a problem. Thus, the swarm represents a set of possible solutions. At each iterative cycle, the position of each particle is updated according to (2), where x_j and v_j are the position and velocity of particle j, respectively.

$$x_j(t+1) = x_j(t) + v_j(t+1) \qquad (2)$$

The computation of v_j considers three terms: the particle's inertia; the particle's cognition, which is based on the best solution found by the particle so far; and social term, which is based on global best solution found by the swarm. The velocity of particle j, at each dimension d, is defined in (3):

$$\begin{aligned} v_{jd}(t+1) = \omega v_{jd}(t) + \varphi_1 r_{1d}(t)(m_{jd} - x_{jd}(t)) \\ +\varphi_2 r_{2d}(t)(m_{gd} - x_{jd}(t)), \end{aligned} \qquad (3)$$

wherein ω is a parameter that weighs the inertia of the particle, φ_1 and φ_2 are parameters that weigh the cognitive and social terms, respectively, r_1 and r_2 are random numbers in [0,1], m_j is the best position visited by particle j so far, and m_g is the best position discovered by the swarm considering the experience of all the particles.

In order to better explore multi-dimensional search spaces, a velocity limit is imposed for each dimension d, as in (4):

$$0 \leq v_{jd} \leq \delta(max_d - min_d), \qquad (4)$$

wherein max_d and min_d are the maximum and minimum limits of the search space at each dimension d and $\delta \in [0,1]$.

The overall computation that the PSO performs to minimize a fitness function $f(x)$ is given in Algorithm 2, where x is the particle position and S is the swarm size.

Algorithm 2 PSO Algorithm

```
begin
    for each particle j, $1 \leq j \leq S$ do
        Set randomly position $x_j$ and velocity $v_j$;
        $m_j \leftarrow x_j$;
    end
    $m_g \leftarrow smallest\ m_j, 1 \leq j \leq S$;
    repeat
        for each particle j, $1 \leq j \leq S$ do
            Update velocity $v_j$, as in (3) and (4);
            Update position $x_j$, as in (2);
            $fitness \leftarrow f(x_j)$;
            $m_k \leftarrow x_j$, whenever $fitness < f(m_j)$;
            $m_g \leftarrow x_j$, whenever $fitness < f(m_g)$;
        end
    until Stopping condition;
    return $m_g$;
end
```

5. THE ACTIVE SYSTEM IDENTIFICATION ATTACK

The Active System Identification attack, herein proposed, is intended to assess the coefficients of a transfer function $G(z) = C(z)P(z)$ of an NCS, wherein $C(z)$ is the controller's control function and $P(z)$ is the plant's transfer function as shown in Figure 1. The transfer functions are all linear time-invariant (LTI). This attack is performed by a MitM that may be located either in the forward or in the feedback link. For the sake of clarity of the analysis presentation, but without loss of generality, we focus on the case where the MitM is in the feedback link, *i.e.* between the plant's sensors and the controller's input. To estimate the model of the attacked NCS, the attacker injects an attack signal $a(k)$, and measure the response of the system to such signal.

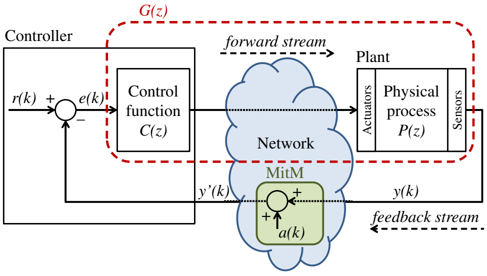


Figure 1: Active System Identification attack with a MitM in the feedback link.

The complete response of the generic NCS shown in Figure 1, considering only the inputs $R(z) = \mathcal{Z}[r(k)]$ and $A(z) = \mathcal{Z}[a(k)]$, is expressed in the z domain by (5):

$$Y(z) = \frac{G(z)}{1+G(z)}R(z) - \frac{G(z)}{1+G(z)}A(z), \tag{5}$$

wherein $Y(z) = \mathcal{Z}[y(k)]$. $\mathcal{Z}$ represents the Z-transform operation. As a premise, in a normal condition, it is considered that $a(k) = 0$ and the system is designed to make $y(k) \rightarrow q$, in such way that $y(k) \approx q\ \forall k > k_s$, *i.e.* the output $y(k)$ of the NCS converges and stabilizes at a constant value q after a certain amount of samples k_s. Indeed, it is usually one of the main aims of a control system. Now, considering $a(k) \neq 0$, the output $y(k)$, $\forall k > k_s$, may be defined approximately as (6):

$$y(k) = q - \mathcal{Z}^{-1}\left[\frac{G(z)}{1+G(z)}A(z)\right], \forall k > k_s. \tag{6}$$

Thus, after k_s, the portion of $y(k)$ caused by $r(k)$ can be eliminated by just subtracting q from (6), which leads to (7):

$$y_a(k) = y(k) - q = -\mathcal{Z}^{-1}\left[\frac{G(z)}{1+G(z)}A(z)\right], \forall k > k_s. \tag{7}$$

wherein $y_a(k)$ represents the portion of $y(k)$ caused by the attack signal $a(k)$. The value of q can be assessed by the attacker through an eavesdropping attack in the feedback stream, by just capturing $y(k)$ after the stabilization of the NCS. The subtraction of q after k_s makes the system identification attack independent of $r(k)$ $\forall k > k_s$. The Active System Identification attack now just relies on the attack signal $a(k)$, which can be chosen, and the response of the system to the attack $y_a(k)$ can be obtained in accordance with (7). The signal $y_a(k)$ starts with $a(k)$ and has the size of a monitoring period T.

If the attack input $a(k)$ and its consequent output $y_a(k)$ are known, the model of $G(z)$ can be assessed by applying the known $a(k)$ in an estimated system, defined by (8):

$$\hat{y}_a(k) = -\mathcal{Z}^{-1}\left[\frac{G_e(z)}{1+G_e(z)}\right] * a(k), \tag{8}$$

wherein $G_e(z)$ is the estimation of $G(z)$ and $\hat{y}_a(k)$ is the output of the estimated system in face of $G_e(z)$. By comparing $\hat{y}_a(k)$ with $y_a(k)$, the attacker is capable to evaluate whether $G_e(z)$ is equal/approximately $G(z)$. Note that $G_e(z)$ is a generic transfer function represented by (9):

$$G_e(z) = \frac{\alpha_n z^n + \alpha_{n-1}z^{n-1} + ... + \alpha_1 z^1 + \alpha_0}{z^m + \beta_{m-1}z^{m-1} + ... + \beta_1 z^1 + \beta_0}, \tag{9}$$

wherein n and m are the order of the numerator and the denominator, respectively, and $[\alpha_n, \alpha_{n-1}, ...\alpha_1, \alpha_0]$ and $[\beta_{m-1}, \beta_{m-2}, ...\beta_1, \beta_0]$ are the coefficients of the numerator and the denominator, respectively, that are intended to be found by this Active System Identification attack. Thus, to find $G(z)$, the coefficients of $G_e(z)$ are adjusted until the estimated output $\hat{y}_a(k)$ converges to the known $y_a(k)$.

In this sense, the BSA and the PSO are used to iteratively adjust the estimated model, by minimizing a specific fitness function presented in this section, until the estimated model $G_e(z)$ converges to the actual $G(z)$ of the real NCS. To compute the fitness of the individuals of the optimization algorithm, *i.e.* the BSA or PSO, the same attack signal $a(k)$ that provided $y_a(k)$, according with (7), is applied on the estimated system defined by (8) and (9), where the coefficients of $G_e(z)$ are the coordinates $x_j = [\alpha_{n,j}, \alpha_{n-1,j}, ...\alpha_{1,j}, \alpha_{0,j}, \beta_{m-1,j}, \beta_{m-2,j}, ...\beta_{1,j}, \beta_{0,j}]$ of an individual j of the BSA/PSO. The output $\hat{y}_{aj}(k)$ is the response of the estimated model (8) (9), in face of $a(k)$, when the coefficients of $G_e(z)$ are x_j. So, the fitness f_j of each individual j is obtained comparing $\hat{y}_{aj}(k)$ with $y_a(k)$, according with (10):

$$f_j = \frac{\sum\limits_{k=0}^{N} (y_a(k) - \hat{y}_{aj}(k))^2}{N}, \tag{10}$$

wherein N is the number of samples that exist during the monitoring period T of $y_a(k)$. Note that, if no other inputs – perturbation or noise – occur in the NCS during T, then $\min f_j = 0$ when $[\alpha_{n,j}, \alpha_{n-1,j}, ...\alpha_{1,j}, \alpha_{0,j}, \beta_{m-1,j}, \beta_{m-2,j}, ...$ $\beta_{1,j}, \beta_{0,j}] = [\alpha_n, \alpha_{n-1}, ...\alpha_1, \alpha_0, \beta_{m-1}, \beta_{m-2}, ...\beta_1, \beta_0]$, *i.e.* when the estimated $G_e(z)$ converges to $G(z)$.

An analogy may be established between this Active System Identification attack and the Chosen Plaintext cryptanalytic attack [19], wherein $a(k)$ corresponds to the chosen plaintext, $y_a(k)$ represents the ciphertext, the equations (8) and (9) together correspond to the encryption algorithm and the actual coefficients $[\alpha_n, \alpha_{n-1}, ...\alpha_1, \alpha_0]$ and $[\beta_{m-1}, \beta_{m-2}, ...\beta_1, \beta_0]$ of $G_e(z)$ correspond to the secret key.

6. RESULTS

In this section, there are presented and analyzed the results obtained with simulations of the proposed Active System Identification attack. The attacked system, shown in Figure 2, consists of a DC motor whose rotational speed is controlled by a Proportional-Integral (PI) controller. This example is chosen due to the use of DC motors in a vast number of real world control systems. Moreover, DC motors has been widely used in previous works about NCS [3, 12, 14, 15]. It is noteworthy that the model herein chosen as an example does not exhaust the potential targets for this attack. NCSs composed by another kinds of LTI devices may also be a target. However, it must be taken into account that the computational cost of the attack, when launched over different LTI systems, may vary with the number of their unknown coefficients – *i.e.* the number of dimensions of the search space explored by the optimization algorithms (BSA or PSO, in this paper).

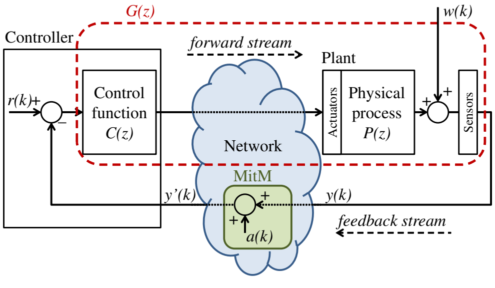


Figure 2: Active System Identification attack on noisy NCS.

The PI control function $C(z)$ and the DC motor transfer function $P(z)$, obtained from [12], are represented by (11):

$$C(z) = \frac{0.1701z - 0.1673}{z - 1}, \quad P(z) = \frac{0.3379z + 0.2793}{z^2 - 1.5462z + 0.5646}. \tag{11}$$

Thereby, the transfer function to be identified $G(z)$ – that is also the open-loop transfer function of the NCS – is defined by (12):

$$G(z) = C(z)P(z) = \frac{g_1 z^2 + g_2 z + g_3}{z^3 + g_4 z^2 + g_5 z + g_6}, \tag{12}$$

wherein $g_1 = 0.0575$, $g_2 = -0.0090$, $g_3 = -0.0467$, $g_4 = -2.5462$, $g_5 = 2.1108$ and $g_6 = -0.5646$. The sample rate of the system is 50 samples/s and the set point $r(k)$ is an unitary step function. Network delay and packet loss are not taken into account in the simulations of this paper.

The structure of the equations (11), and so the structure of (12), are previously known by the attacker once that, as a premise, it is known that the target is an NCS that controls a DC motor using a PI controller. Thus, in these simulations, the goal of the Active System Identification attack is to discover g_1, g_2, g_3, g_4, g_5 and g_6.

The chosen attack signal $a(k)$ is a discrete-time unit impulse (13):

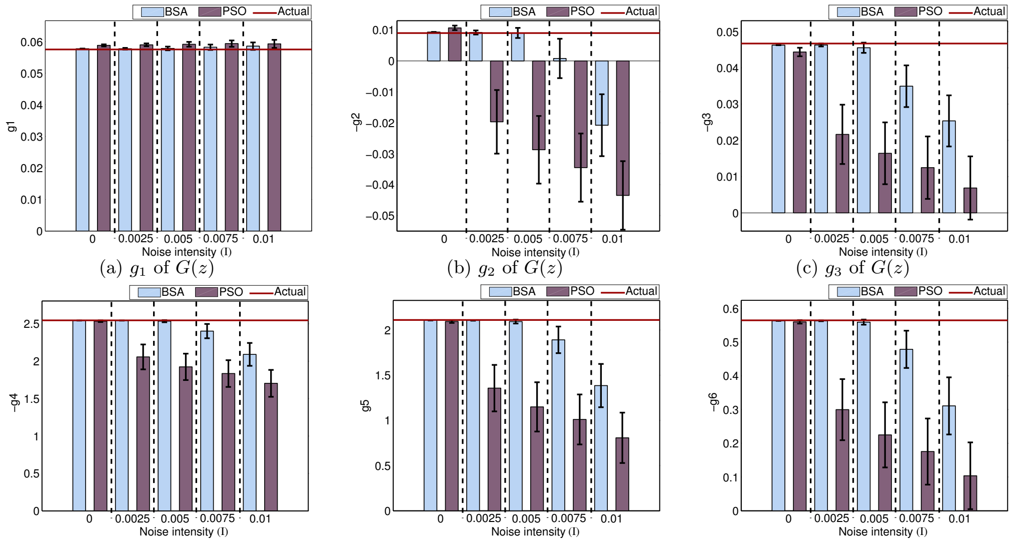
$$a(k) = \begin{cases} 1 & \text{if } k = k_a; \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

wherein k_a is the single sample in which the attacker interfere in the system by adding 1 to the feedback stream. Note that the discrete-time unit impulse is chosen to excite the NCS due to its short active time – *i.e.* one sample –, which increases the stealthiness of the attack in the time domain.

The effectiveness of the Active System Identification attacks are evaluated in both conditions with and without noise. To simulate the noise, it is inserted $w(k) \sim N(\mu, \sigma)$, indicated in Figure 2, which is a white Gaussian noise wherein N is a normal distribution, μ is its mean and σ is its standard deviation. In all simulations the mean is $\mu = 0\ rad/s$. The standard deviation is adjusted such that 95% of the amplitudes of $w(k)$ are within $\pm I$ $(I = 2\sigma)$. There are considered four different noise intensities I: 0 (no noise), 0.0025 rad/s, 0.005 rad/s and 0.01 rad/s. For each noise intensity I, there are executed 100 different simulations, for each of the mentioned metaheuristics. In each simulation, the feedback stream is captured by the attacker during a period $T = 2s$ (100 samples), starting at sample $k_a + 1$.

The attack model was implemented in MATLAB, where the simulations were carried out. The SIMULINK tool was used to compute $y_a(k)$ and $\hat{y}_{aj}(k)$ – the latter, for each individual j of the optimization algorithms. The parameters of the BSA and PSO described in Sections 3 and 4, respectively, were empirically adjusted through a set of simulations without noise $(I = 0)$. These parameters are then used for all noise conditions. In the BSA-based attacks, the parameter η is set to 1. In the PSO-based attacks, it is used the following parameters configuration: $\omega = 0.4$, $\varphi_1 = \varphi_2 = 1.5$ and $\delta = 0.1$. In both algorithms, the population is set to 100 individuals and the limits of each dimension of the search space are $[-10, 10]$. In each simulation, the BSA and the PSO are executed for 4500 iterations.

Figure 3 presents the mean estimated values of g_1, g_2, g_3, g_4, g_5 and g_6, with a Confidence Interval (CI) of 95%, for different values of noise intensity I. Note that the actual values of these coefficients are also depicted in Figure 3. In



(a) g_1 of $G(z)$
(b) g_2 of $G(z)$
(c) g_3 of $G(z)$
(d) g_4 of $G(z)$
(e) g_5 of $G(z)$
(f) g_6 of $G(z)$

Figure 3: Mean of the estimated coefficients of $G(z)$, with CI of 95%, in face of different noise intensities I.

(a) BSA and PSO, without noise
(b) BSA with $I = 0.005$
(c) PSO with $I = 0.005$

Figure 4: Response of actual and estimated systems produced by $a(k)$, in face of different noise intensities.

this Figure, it is possible to compare the results achieved by the BSA-based and the PSO-based attacks. For the computation of each outcome presented in Figure 3, there were not taken into account the results beyond two standard deviation from the mean of each set of 100 simulations. According with Figure 3, it is possible to verify that, for all coefficients of $G(z)$, both the BSA-based and PSO-based attacks present good accuracy when $I = 0$ (*i.e.* without noise, the mean values of the estimated coefficients are close to their actual values). Despite the similar and accurate performance of the two metaheuristics without noise, it is possible to state that the BSA presented a performance slightly better than the PSO in this noise condition ($I = 0$), specially with regard to the coefficients g_1, g_2 and g_3. Note that, the performance of the PSO-based attack is degraded when noise is added to the system. This performance degradation of the PSO occurs for $I \geq 0.0025$, and tends to be more expressive with the increase of I. On the other hand, from Figure 3, it is possible to verify that the BSA-based attack still present good accuracy for noise intensities up to 0.005. When $I \leq 0.005$, all coefficients estimated by the BSA-based attack present a mean close to its actual value, with a small CI. When $I \geq 0.0075$, the performance of the BSA-based attack decreases with the raise of noise in a more expressive way, being worst when $I = 0.01$. Among the six coefficients of $G(z)$, in general, the estimation of g_2 presents the lowest accuracy for both BSA-based and PSO-based attacks. We attribute this behavior to a lower sensitivity that the output $\hat{y}_a(k)$ of the estimated system has to the variation of g_2. This means that, in this problem, f_j grows faster for errors in g_1, g_3, g_4, g_5 and g_6 than for errors in g_2, making the BSA population converge less accurately in dimension g_2.

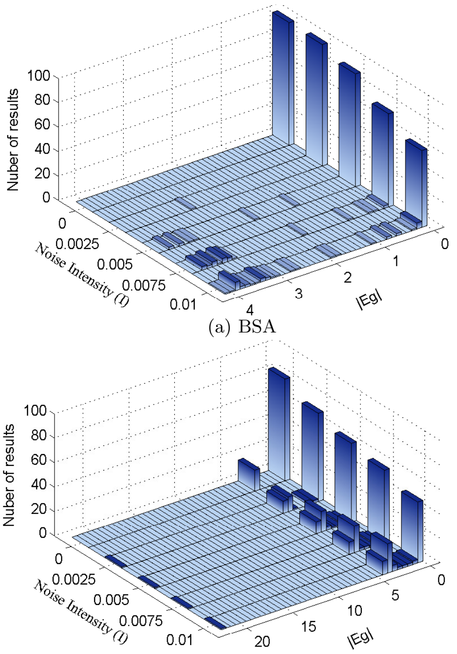
The performance of the attacks can also be evaluated in the k domain through the exemples provided in Figure 4, considering two different intensities of noise: without noise, in Figure 4(a); and with $I = 0.005$, in Figures 4(b) and 4(c). In Figure 4(a), its is shown that, without noise, the response of the system estimated by both BSA-based and PSO-based attacks matches the response of the actual system, with high accuracy. In Figure 4(b), even with a noise intensity of $I = 0.005$, the response of the system estimated by the BSA-based attack still matches the response of the actual system, indicating the convergence of $G_e(z)$ to $G(z)$ and ratifying the statistics shown in Figure 3 for the BSA with such noise intensity. On the other hand, when applying the PSO-based attack with the same noise, as exemplified in Figure 4(c), there is a slight difference between the response of the estimated system and the response of the actual system, produced by the mismatch of the estimated coefficients in the presence of such noise intensity. This exemplifies the worst performance of the PSO-based attacks when compared with the BSA-based attacks in face of the same noise intensities.

To synthesize the error of each solution found, it is computed $|E_g|$ according with (14):

$$|E_g| = \sqrt{\sum_{i=1}^{6} (g_i - g_{ei})^2}, \tag{14}$$

wherein g_i and g_{ei} are the actual and estimated coefficients of the attacked system, respectively, and i is the index number of each of the six coefficients of the model being assessed. Note that $|E_g|$ is the module of a vector composed by the error of each coefficient found, which represents another metric to evaluate the performance of each attack. The histograms of $|E_g|$ are presented in Figure 5, considering the mentioned noise intensities. It graphically shows that higher values of $|E_g|$ tend to appear more frequently as the noise intensity grows, in both BSA-based and PSO-based attacks. However, based on these histograms it is possible to verify that the mode of $|E_g|$ is close to zero for all noise intensities, using both metaheuristics. This indicates that, even in the presence of noise, most solutions present low deviations from the actual coefficients. Note that, for all noise intensities, the BSA-based attacks provide more results in the modal class – where $|E_g|$ is close to zero – than the PSO-based attacks. Moreover, the worst results of the BSA-based attacks have an $|E_g|$ about 4, when $I \geq 0.005$, while the worst results of the PSO-based attacks have an $|E_g| > 20$, when $I \geq 0.0025$. These results, together with the statistics shown in Figure 3, indicate that the performance of the Active System Identification attack is better when implemented with the BSA than with the PSO. It is worth mentioning that, to achieve these results, the BSA-based attacks consumed an average processing time $(6.68 \pm 0.47)\%$ higher than the PSO-based attacks.

In general, the outcomes indicate that, for the same amplitude of attack signal $a(k)$, the performance of the attack tends do decrease as the noise intensity increases, *i.e.* when the attack signal-to-noise ratio decreases. The minimum length of the attack signal in terms of number of manipulated samples, *i.e.* one single sample, improves the stealthiness of the attack in the k domain. On the other hand, a minimum attack signal-to-noise ratio required to guarantee the performance of this attack is a drawback with respect to its stealthiness, from the attacker's point of view. This issue makes more difficult for the attacker to approximate the amplitude of $a(k)$ from the noise amplitude, or to noise values that have higher probability to occur, which should help to increase the stealthiness of the attack signal in terms of amplitude.



(a) BSA

(b) PSO

Figure 5: Histograms of $|E_g|$ for different noise intensities.

7. CONCLUSION

The present work defines and propose an Active System Identification attack that may be launched over NCSs, in order to gather the data required for the design of other sophisticated cyber-physical attacks. The attack herein proposed is implemented based on two bio-inspired algorithms: the BSA and the PSO. It is shown that, in this problem, the BSA-based attacks provide better performance than the PSO-based attacks, specially in the presence of noise.

In general, the results indicate that the attack is capable to estimate the coefficients of the open-loop transfer function of an NCS, which is known to be enough for further manipulation of the system's behavior through conventional root locus analysis/modification. It is demonstrated the capability of the attack to achieve its goal even when:

- no meaningful information is passing through its communication links, *i.e.* when the system had achieved its steady state;
- the attacker intercepts the communication of the NCS at only one point, *i.e.* the attacker does not need to in-

tercept both forward and feedback streams to estimate the open-loop transfer function of the system;

- the NCS is noisy (particularly the BSA-based attack, for $0 \leq I \leq 0.0075$).

For future work we plan to investigate possible techniques that guarantee the performance of the attack even with small attack signal-to-noise ratio. Also, we plan – and encourage other researches – to investigate countermeasures to identify and prevent Active System Identification attacks.

8. ACKNOWLEDGMENT

This research was partially supported by the Brazilian research agencies CNPq and FAPERJ.

9. REFERENCES

[1] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen. Cyber security of water scada systems part i: analysis and experimentation of stealthy deception attacks. *IEEE Transactions on Control Systems Technology*, 21(5):1963–1970, 2013.

[2] E. Bou-Harb, M. Debbabi, and C. Assi. Cyber scanning: a comprehensive survey. *IEEE Communications Surveys & Tutorials*, 16(3):1496–1519, 2014.

[3] X. Chen, Y. Song, and J. Yu. Network-in-the-loop simulation platform for control system. In *AsiaSim 2012*, pages 54–62. Springer, 2012.

[4] P. Civicioglu. Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematics and Computation*, 219(15):8121–8144, 2013.

[5] A. O. de Sá, L. F. R. d. C. Carmo, and R. C. S. Machado. Covert attacks in cyber-physical control systems. *to appear in IEEE Transactions on Industrial Informatics, available at https://arxiv.org/abs/1609.09537*, arXiv:1609.09537, 2016.

[6] M. El-Sharkawi and C. Huang. Variable structure tracking of dc motor for high performance applications. *Energy Conversion, IEEE Transactions on*, 4(4):643–650, 1989.

[7] A. A. Farooqui, S. S. H. Zaidi, A. Y. Memon, and S. Qazi. Cyber security backdrop: A scada testbed. In *Computing, Communications and IT Applications Conference (ComComAp), 2014 IEEE*, pages 98–103. IEEE, 2014.

[8] N. V. George and G. Panda. A particle-swarm-optimization-based decentralized nonlinear active noise control system. *IEEE Transactions on Instrumentation and Measurement*, 61(12):3378–3386, 2012.

[9] D. Guha, P. K. Roy, and S. Banerjee. Application of backtracking search algorithm in load frequency control of multi-area interconnected power system. *Ain Shams Engineering Journal*, 2016.

[10] R. Kennedy, J. e Eberhart. Particle swarm optimization. In *Proceedings of 1995 IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.

[11] R. Langner. Stuxnet: Dissecting a cyberwarfare weapon. *Security & Privacy, IEEE*, 9(3):49–51, 2011.

[12] M. Long, C.-H. Wu, and J. Y. Hung. Denial of service attacks on network-based control systems: impact and mitigation. *Industrial Informatics, IEEE Transactions on*, 1(2):85–96, 2005.

[13] R.-E. Precup, A.-D. Balint, M.-B. Radac, and E. M. Petriu. Backtracking search optimization algorithm-based approach to pid controller tuning for torque motor systems. In *Systems Conference (SysCon), 2015 9th Annual IEEE International*, pages 127–132. IEEE, 2015.

[14] Y. Shi, J. Huang, and B. Yu. Robust tracking control of networked control systems: application to a networked dc motor. *IEEE Transactions on Industrial Electronics*, 60(12):5864–5874, 2013.

[15] M. L. Si, H. X. Li, X. F. Chen, and G. H. Wang. Study on sample rate and performance of a networked control system by simulation. In *Advanced Materials Research*, volume 139, pages 2225–2228. Trans Tech Publ, 2010.

[16] R. Smith. A decoupled feedback structure for covertly appropriating networked control systems. In *Proceedings of the 18th IFAC World Congress 2011*, volume 18. IFAC-PapersOnLine, 2011.

[17] R. S. Smith. Covert misappropriation of networked control systems: Presenting a feedback structure. *Control Systems, IEEE*, 35(1):82–92, 2015.

[18] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S. T. Kent, and W. T. Strayer. Single-packet ip traceback. *IEEE/ACM Transactions on Networking (ToN)*, 10(6):721–734, 2002.

[19] W. Stallings. *Cryptography and network security: principles and practices*. Pearson Education India, 2006.

[20] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson. A secure control framework for resource-limited adversaries. *Automatica*, 51:135–148, 2015.

[21] T. Tran, Q. P. Ha, and H. T. Nguyen. Robust non-overshoot time responses using cascade sliding mode-pid control. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 2007.

[22] H. J. Tulleken. Generalized binary noise test-signal concept for improved identification-experiment design. *Automatica*, 26(1):37–49, 1990.

[23] S. Uong and I. Ngamroo. Coordinated control of dfig wind turbine and svc for robust power system stabilization. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2015 12th International Conference on*, pages 1–6. IEEE, 2015.

[24] W. Xin, L. Ran, W. Yanghua, P. Yong, and Q. Bin. Self-tuning pid controller with variable parameters based on particle swarm optimization. In *Intelligent System Design and Engineering Applications (ISDEA), 2013 Third International Conference on*, pages 1264–1267. IEEE, 2013.

APPENDIX F

Use of Switching Controllers for Mitigation of Active Identification Attacks in Networked Control Systems

Alan Oliveira de Sá
Admiral Wandenkolk Instruction Center, Brazilian Navy; and Institute of Mathematics/NCE, Federal University of Rio de Janeiro, RJ, Brazil
Email: alan.oliveira.sa@gmail.com

Luiz F. R. da C. Carmo
National Institute of Metrology, Quality and Technology; and Institute of Mathematics/NCE, Federal University of Rio de Janeiro, RJ, Brazil
Email: lfrust@inmetro.gov.br

Raphael C. S. Machado
National Institute of Metrology, Quality and Technology; and Rio de Janeiro Federal Center for Technological Education, RJ, Brazil
Email: rcmachado@inmetro.gov.br

***Abstract*—The literature regarding to cyber-physical attacks in Networked Control Systems (NCS) indicates that covert and accurate attacks must be planned based on an accurate knowledge about the model of the attacked system. In this sense, the literature on NCS recognizes the Active System Identification attack as a tool to provide the attacker with the required system models. However, there is still a lack of discussion about countermeasures for this specific attack. In this sense, this work proposes the use of a randomly switching controller as a countermeasure for the Active System Identification attack. The simulation results indicate that this countermeasure is capable to mitigate the mentioned attack at the same time that it performs a satisfactory plant control.**

1. Introduction

A Networked Control System (NCS) consists of physical plant controlled by a digital controller – *i.e.* a computational system – through a communication network, which, indeed, integrates the cyberspace to the physical domain. Motivated by the increasing use of NCSs in industrial plants and critical infrastructures, and considering the cyber threats that can affect these systems, studies have been conduced to characterize vulnerabilities and propose security solutions for NCSs. In this context, the literature [1], [2], [3], [4], [5] demonstrates that a number of sophisticated – covert and accurate – attacks need to be built based on an accurate knowledge about the model of the attacked system.

Recent works [5], [6] introduced a set of System Identification attacks that may be launched against NCSs to provide the attacker with the required system models and, therefore, support the design of other sophisticated attacks. For instance, in [5], the joint operation of a Passive System Identification attack and a Data Injection attack is used to degrade, in a physically covert fashion, the service performed by a plant. It is shown that the performance of this covert data injection attack is directly affected by the accuracy of the data obtained by the Passive System Identification attack.

In [6], the authors introduce the Active System Identification Attack as a tool to provide the attacker with the required system models. Although the authors of [6] encourage the development of countermeasures for the mentioned attack, there is a lack of discussion about countermeasures for this specific attack. In this sense, this work aims to discuss and propose a countermeasure for the Active System Identification Attack. The straightforward countermeasure to prevent the success of a System Identification attack in an NCS is to avoid unauthorized access to the control loop using, for example, network segmentation, demilitarized zones (DMZ), firewall policies and implementing specific network architectures, such as established in [7]. A complementary countermeasure – in case the attacker is capable to access the control loop – is to hinder the access to the data flowing in the NCS using, for example, symmetric-key encryption algorithms, hash algorithms and a timestamp strategy to form a secure transmission mechanism between the controller and the plant, as proposed in [8]. However, when the mentioned countermeasures fail and the attacker gain access to the data flowing in the NCS, the alternative to prevent the attacker to obtain the model of the system is to hinder the analysis of the captured data – *i.e.* make the System Identification algorithm inaccurate/ineffective.

One possible strategy to cause difficulties to the System Identification algorithm is to have, in the NCS, specific control functions that are, at the same time, harder to be identified and capable to control the plant. Considering this strategy, it is proposed in this work the use of randomly switching controllers as a feasible countermeasure for the Active System Identification attack proposed in [6].

The remainder of this paper is organized as follows. Section 2, presents some related works. Section 3, describes the Active System Identification attack proposed in [6]. In Section 4, the switching controller is presented and discussed as a countermeasure for the Active System Identification attack. Section 5, presents simulation results, where the performance of the switching controller is analyzed from the countermeasure and control perspectives. Finally, Section 6 presents some conclusions and possible future works.

2. Related Works

This section presents a review on the literature encompassing covert/model-dependent attacks and System Identification attacks in NCSs. In [3], the authors analyze a wide variety of attacks in NCSs and establish the requirements for the attacks in terms of the model knowledge, disclosure and disruption resources. In their work, it is stated that high level of knowledge about the model of the attacked system is required to build covert attacks.

In [1], [2], [4], there are proposed and analyzed examples of covert attacks that agree with the statement provided in

[3]. In [1], [4], the attacker, acting as a man-in-the-middle (MitM), injects false data in the forward stream of the NCS to take control of the plant. The attacker, then, uses the model of the attacked plant to compute the data that is injected in the feedback stream to make the attack covert. The covertness of the attack proposed in [1] is analyzed from the perspective of the signals arriving to the controller and, as demonstrated in [4], it depends on the difference between the actual model of the plant and the model known by the attacker. In [2] the attacker, based on the model of the system, injects data in the NCS to covertly steal water from the Gignac canal system located in Southern France.

In [1], [2], [3], [4], where the attacks are designed and built based on the models of the targeted systems, it is not described how these models are obtained by the attacker. It is just stated that the models are previously known to subsidize the design of these covert/model-dependent attacks.

To fill this gap, in [5] and [6], the authors propose two new kinds of attack: the Passive System Identification attack [5]; and the Active System Identification attack [6]. These attacks, which belong to the category of Cyber-physical Intelligence attacks [5], are intended to estimate the models of the attacked system. The Passive System Identification attack [5] does not need to inject signals on the NCS to estimate its models. However, it depends on the occurrence of events, that are not controlled by the attacker, to produce signals that carry meaningful information for the system identification algorithm. On the other hand, the Active System Identification attack [6], constitutes an alternative to the passive System Identification attacks in situations where the attacker may not wait so long for the occurrence o such meaningful signals. To do so, as described in Section 3, the attacker estimates the open-loop transfer function of the system by injecting an attack signal in the NCS and eavesdropping its response at only one point of interception. In this work it is proposed a countermeasure to hinder the Active System Identification attack, even if the attacker gets access to the data flowing in the NCS.

3. The Active System Identification Attack

In this section, the Active System Identification attack [6] is briefly described, in order to provide the information necessary to comprehend the proposed countermeasure. The referred attack aims to estimate the coefficients of the open loop transfer function $G(z) = C(z)P(z)$ of an NCS, shown in Figure 1.To do so, the attack is performed in three stages:

- STAGE-I: As a Man in the Middle (MitM), the attacker injects $a(k)$ in the NCS, as shown in Figure 1.
- STAGE-II: The attacker eavesdrops the output $y(k)$ of the plant, during a monitoring period T, in order to obtain the response $y_a(k)$ caused by $a(k)$.
- STAGE-III: Knowing $a(k)$ and $y_a(k)$, the attacker estimates the open-loop transfer function of the system $G(z)$ by applying $a(k)$ in an estimated model $G_e(z)$, which is adjusted until its estimated output $\hat{y}_a(k)$ matches $y_a(k)$. In [6], this adjustment is performed by bio-inspired optimization algorithms.

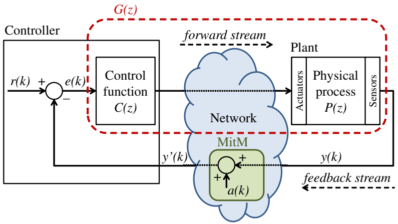


Figure 1: Active System Identification attack in an NCS.

Note that $y_a(k)$, obtained in STAGE-II, is only a portion of $y(k)$. The complete response of the system is $y(k) = y_r(k) + y_a(k)$, wherein $y_r(k)$ is the response of the system caused by $r(k)$. Considering that the system is stable, the output $y_r(k)$ caused by $r(k)$ converges and stabilizes at a constant value q after a certain amount of samples k_s. Thus, in STAGE-II, to obtain $y_a(k)$, $\forall k > k_s$, the attacker must compute $y_a(k) = y(k) - q$. This eliminates the portion of $y(k)$ caused by $r(k)$, making the identification problem dependent of only $A(z) = \mathcal{Z}[a(k)]$, as shown in (1):

$$y_a(k) = y(k) - q = -\mathcal{Z}^{-1}\left[\frac{G(z)}{1+G(z)}A(z)\right], \forall k > k_s. \tag{1}$$

wherein $\mathcal{Z}$ represents the Z-transform operation. The value of q is obtained by the attacker by capturing $y(k)$ after the stabilization of the NCS, before the injection of $a(k)$.

In STAGE-III, the attacker assess $G(z)$ by applying $a(k)$ in an estimated system, defined by (2) and (3):

$$\hat{y}_a(k) = -\mathcal{Z}^{-1}\left[\frac{G_e(z)}{1+G_e(z)}\right] * a(k), \tag{2}$$

$$G_e(z) = \frac{\alpha_n z^n + \alpha_{n-1} z^{n-1} + \ldots + \alpha_1 z^1 + \alpha_0}{z^m + \beta_{m-1} z^{m-1} + \ldots + \beta_1 z^1 + \beta_0}, \tag{3}$$

wherein $G_e(z)$ is the estimation of $G(z)$ and $\hat{y}_a(k)$ is the output of the estimated system. $[\alpha_n, \alpha_{n-1}, \ldots\alpha_1, \alpha_0]$ and $[\beta_{m-1}, \beta_{m-2}, \ldots\beta_1, \beta_0]$ are the coefficients of the numerator and denominator of $G_e(z)$, respectively, that are intended to be found by the Active System Identification attack. The order of the numerator and denominator are expressed by n and m, respectively. Thus, to find $G(z)$, the coefficients of $G_e(z)$ are adjusted until the estimated output $\hat{y}_a(k)$ converges to the known $y_a(k)$.

In [6], the Backtracking Search Optimization algorithm (BSA) [9] and the Particle Swarm Optimization (PSO) [10] are used to iteratively adjust the parameters of $G_e(z)$, by minimizing a specific fitness function, until $G_e(z)$ converges to $G(z)$. The coefficients of $G_e(z)$ are the coordinates $x_j = [\alpha_{n,j}, \alpha_{n-1,j}, \ldots\alpha_{1,j}, \alpha_{0,j}, \beta_{m-1,j}, \beta_{m-2,j}, \ldots\beta_{1,j}, \beta_{0,j}]$ of an individual j of the BSA/PSO. The fitness f_j of each individual j of the BSA/PSO is computed as (4):

$$f_j = \frac{\sum_{k=0}^{N} (y_a(k) - \hat{y}_{aj}(k))^2}{N}, \tag{4}$$

wherein N is the number of samples during T and $\hat{y}_{aj}(k)$ is the response of the estimated model (2) (3) caused by $a(k)$, when the coefficients of $G_e(z)$

are x_j. Note that $\min f_j = 0$ when $[\alpha_{n,j}, \alpha_{n-1,j}, ... \alpha_{1,j}, \alpha_{0,j}, \beta_{m-1,j}, \beta_{m-2,j}, ...\beta_{1,j}, \beta_{0,j}] = [\alpha_n, \alpha_{n-1}, ... \alpha_1, \alpha_0, \beta_{m-1}, \beta_{m-2}, ...\beta_1, \beta_0]$, *i.e.* when $G_e(z) = G(z)$.

4. Switching Controllers: a Countermeasure

As discussed in Section 1, one possible way to cause difficulties to the System Identification algorithm is to have, in the NCS, specific transfer functions that are harder to be identified. So, it is necessary to lean over $C(z)$ and $P(z)$ (see Figure 1) to verify what can be done to hinder the identification of the NCS. In the case of the plant, it is not desired or even feasible to modify $P(z)$ just to make it harder to be identified. Modify $P(z)$ means modify the physical process being controlled, which is not convenient. However, it is possible to design a controller so that it simultaneously meet two objectives:

O.I - Comply with the plant's control requirements considering, firstly, its stability and, secondly, other requirements such as: settling time, overshoot, etc.

O.II - Hinder the identification process, so that the model obtained by the attacker is imprecise or ambiguous, in such a way that the attacker hesitates to launch covert or model-dependent attacks against the NCS.

Note that, in the case of the Active System Identification Attack proposed in [6], the attacker does not identify $C(z)$ and $P(z)$ separately. The attacker intercepts the control loop at a single point and, from that point of interception, estimates the system's open-loop transfer function $G(z) = C(z)P(z)$, as shown in Figure 1. Assuming that it is not convenient to modify $P(z)$, as previously discussed, $C(z)$ must be designed to hinder the identification of the open-loop transfer function of the NCS. So, considering O.I and O.II, we propose the use of switching controllers as a countermeasure for the Active System Identification attack.

A Switching Controller consists of a set of control functions $C_i(z)$, $i \in \mathcal{I} = \{1, ..., N\}$, that are switched among N states by a switching rule S, to perform the control of a plant $P(z)$, as in Figure 2. When all $C_i(z)$ and $P(z)$ are linear, as the NCS herein discussed, the system is a *switched linear system* (SLS). For the sake of clarity, in this work, the switching controller is discussed with only two control functions $C_1(z)$ and $C_2(z)$. In general S considers the behavior of the plant to switch among the control functions, such as in [11]. However, in the present this work, the switching rule does not take into account the plant's behavior, to command the switchings. To make de identification more difficult, the proposed switching rule is described as a Markov chain, shown in Figure 3, where the control functions are switched at random intervals, following the probabilities $p_{11}(l)$, $p_{12}(l)$. $p_{21}(l)$ and $p_{12}(l)$, wherein l is the number of sampling intervals occurred since the last switch. The reason to switch the control functions at random intervals is that, according to [12], if the switching time is known, the identification of an SLS is straightforward. However, when the switching time is not available the identification of SLSs becomes a nontrivial task. The probabilities, $p_{12}(l)$ and $p_{21}(l)$ are taken from the probability density function (PDF) shown in Figure 4, wherein a is the minimum number of sampling intervals that the system have to remain in the same state and b is the maximum number of sampling intervals that the system can remain in the same state. Note that $p_{11}(l) = 1 - p_{12}(l)$ and $p_{22}(l) = 1 - p_{21}(l)$.

A valid strategy to achieve the stability on an SLS is by restricting the switching events, for example, by establishing a minimum dwell time – *i.e.* the time between two consecutive switches. In an SLS, the instability generated when switching among two stable subsystems is caused by the failure to absorb the energy increase, caused by the switchings [13]. Intuitively, it is reasonable to think that if the system stays at stable subsystems long enough, the system becomes able to avoid the energy increase caused by the switchings, maintaining its stability. In [14], it is proven that it is always possible to preserve the stability when all the subsystems are stable and the dwell time is sufficiently large. Actually, it is not an issue if the SLS occasionally have a smaller dwell time, provided it does not occur too frequently. It was shown in [15], [16] that if all the subsystems are exponentially stable then the SLS remains exponentially stable provided that the *average dwell time* is sufficiently large.

In the present work, $C_1(z)$ and $C_2(z)$ are separately designed based on the root-locus analysis, in order to make each subsystem stable. Then, the overall stability is achieved by adjusting the parameters a and b of the PDF shown in Figure 4, aiming an *average dwell-time* that makes the NCS stable. Besides being adjusted aiming the stability of the system, a and b are also adjusted to mitigate the system identification attack. So, concerning O.I, specifically for the sake of stability, a and b are increased as much as possible to ensure the minimum *average dwell-time* required for stability. On the other hand, regarding O.II, a and b are adjusted to make the identification attack as much imprecise as possible, which is not necessarily obtained with high dwell times. In this sense, a and b are empirically adjusted to meet both potentially conflicting objectives.

5. Results

In this section, the performance of the proposed countermeasure is analyzed in face of the Active System Identification attack proposed in [6]. Two NCSs are used for comparison: one with the proposed countermeasure (using a switching controller); and another without the proposed countermeasure (using a non-switching controller). The model of both NCSs, as well as the attack parameters, are described in Section 5.1. Section 5.2 presents the results of the switching controller as a countermeasure for the Active System Identification attack. Section 5.3 presents the performance of this countermeasure from the control perspective, in order to identify possible trade-offs that may exist between O.I and O.II (see Section 4). The simulations of Sections 5.2 and 5.3 were performed in MATLAB.

5.1. Attacked NCSs and Parameters of the Attack

The NCS without the proposed countermeasure – also referred here as a *system with vulnerable model* – is the

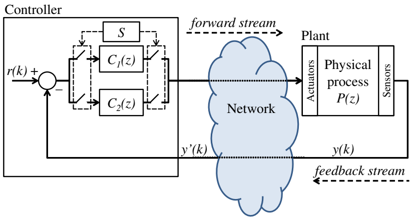


Figure 2: Switching controller in an NCS.

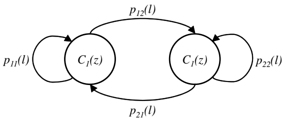


Figure 3: Markov chain switching rule.

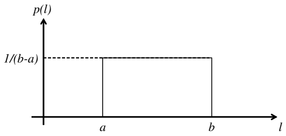


Figure 4: Probability density function of p_{12} and p_{21}.

same NCS attacked in [6], endowed with a non-switching controller. It consists of a DC motor whose rotational speed is controlled by a Proportional-Integral (PI) controller. The DC motor's transfer function $P(z)$ and the PI control function $C_1(z)$ are represented by (5) and (6), respectively:

$$P(z) = \frac{0.3379z + 0.2793}{z^2 - 1.5462z + 0.5646}, \tag{5}$$

$$C_1(z) = \frac{0.1701z - 0.1673}{z - 1}. \tag{6}$$

Thereby, the open-loop transfer function of the *system with vulnerable model* $G_1(z)$ – to be identified – is defined as (7):

$$G_1(z) = C_1(z)P(z) = \frac{g_{1,1}z^2 + g_{2,1}z + g_{3,1}}{z^3 + g_{4,1}z^2 + g_{5,1}z + g_{6,1}}, \tag{7}$$

wherein $g_{1,1} = 0.0575$, $g_{2,1} = -0.0090$, $g_{3,1} = -0.0467$, $g_{4,1} = -2.5462$, $g_{5,1} = 2.1108$ and $g_{6,1} = -0.5646$.

The NCS endowed with the proposed countermeasure – *i.e.* the switching controller – also controls a DC motor defined by the transfer function (5). The switching controller switches among two control functions: $C_1(z)$, that is the same control function (6) of the *system with vulnerable model*; and $C_2(z)$ defined as (8).

$$C_2(z) = \frac{0.1208z - 0,1167}{z - 1}. \tag{8}$$

Therefore, the NCS with the switching controller is an SLS composed by two subsystems, each having an open-loop transfer function. The open-loop transfer functions are: $G_1(z)$, that is the same open-loop transfer function (7) of the *system with vulnerable model*; and $G_2(z)$ defined by (9),

$$G_2(z) = C_2(z)P(z) = \frac{g_{1,2}z^2 + g_{2,2}z + g_{3,2}}{z^3 + g_{4,2}z^2 + g_{5,2}z + g_{6,2}}, \tag{9}$$

wherein $g_{1,2} = 0.0408$, $g_{2,2} = -0.0057$, $g_{3,2} = -0.0326$, $g_{4,2} = -2.5462$, $g_{5,2} = 2.1108$ and $g_{6,2} = -0.5646$. Note that the denominators of $G_1(z)$ and $G_2(z)$ are equal, given that just the numerators of $C_1(z)$ and $C_2(z)$ are different. Thus, $g_{4,1} = g_{4,2}$, $g_{5,1} = g_{5,2}$ and $g_{6,1} = g_{6,2}$.

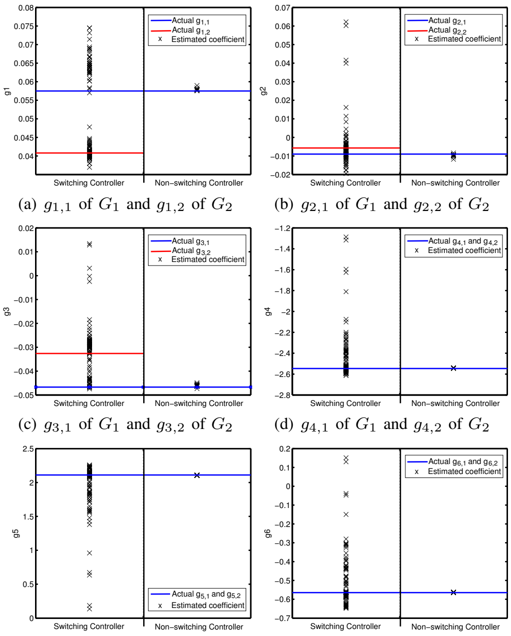
The control functions $C_1(z)$ and $C_2(z)$ are designed to make both subsystems of this SLS stable. As described in Section 4, the control functions are randomly switched based on the Markov chain shown in Figure 3, under a restricted switching policy, whose restrictions are bounded by the PDF shown in Figure 4. The parameters of the PDF were empirically adjusted to $a = 20$ and $b = 40$, in order to meet O.I and O.II, as discussed in Section 4. It is worth mentioning that, regarding O.I, the parameters a and b were empirically adjusted aiming, primarily, the overall stability of the system. However, the settling time and the overshoot of the system are also evaluated in these simulations.

The attack is implemented using the BSA, given that this metaheuristic presented the best performance in the attack simulations of [6]. The parameters of the BSA are the same as in [6]: the population has 100 individuals; the limits of each dimension of the search space are $[-10, 10]$; and η – that establishes the amplitude of the movements of the individuals of the BSA – is set to 1. In each simulation, the BSA is executed for 4500 iterations. As in [6], the attack signal shown in Figure 1 is a discrete-time unitary impulse, *i.e.* $a(k) = \delta(k - k_a)$,wherein k_a is the single sample in which the attacker interfere in the system by adding 1 to the feedback stream. In each simulation, the feedback stream is captured by the attacker during a period $T = 2s$ (100 samples), starting at sample $k_a{+}1$. In both NCSs, the sample rate is 50 samples/s and the set point $r(k)$ is an unitary step function. Network delay and packet loss are not taken into account in the simulations of this paper.

5.2. Performance as a Countermeasure

This section presents the results obtained by the Active System Identification attack, when launched in the NCSs described in Section 5.1 – one NCS using the switching controller and other using the non-switching controller. In each NCS, there were executed 100 attack simulations.

All coefficients estimated by the 100 attack simulations in each NCS are presented in Figure 5. Recall that the NCS with the non-switching controller just have one open-loop transfer function $G_1(z)$, while the NCS with the switching controller has two open-loop transfer functions $G_1(z)$ and $G_2(z)$. Note that the actual values of the coefficients $[g_{1,1}, g_{2,1}, g_{3,1}, g_{4,1}, g_{5,1}, g_{6,1}]$ and $[g_{1,2}, g_{2,2}, g_{3,2}, g_{4,2}, g_{5,2}, g_{6,2}]$ of $G_1(z)$ and $G_2(z)$, respectively, are also depicted in Figure 5. By observing Figures 5(a) to 5(f), it is possible to state that the coefficients estimated in the NCS with the non-switching controller are precise and accurate. In the NCS with non-switching controller, the Active System Identification attack provides the information and the confidence that the attacker needs to design other model-dependent attacks. On the other hand, in the NCS endowed with the proposed countermeasure, the use of the switching



(a) $g_{1,1}$ of G_1 and $g_{1,2}$ of G_2

(b) $g_{2,1}$ of G_1 and $g_{2,2}$ of G_2

(c) $g_{3,1}$ of G_1 and $g_{3,2}$ of G_2

(d) $g_{4,1}$ of G_1 and $g_{4,2}$ of G_2

(e) $g_{5,1}$ of G_1 and $g_{5,2}$ of G_2

(f) $g_{6,1}$ of G_1 and $g_{6,2}$ of G_2

Figure 5: Coefficients estimated by the attack in NCSs with and without the proposed countermeasure.

controller causes the dispersion of the estimated values, reducing the precision and the accuracy of the coefficients obtained by the attacker. As shown in Figure 5, the set of estimated values in this SLS are spread and does not accurately indicate any of the coefficients of $G_1(z)$ and $G_2(z)$.

The impact of the switching controller in the attack performance can also be verified by comparing the global minimum values found for the fitness function (4). In the NCS endowed with the switching controller, the global minimum values of all attack simulations are within 1.81×10^{-06} and 1.96×10^{-04} (the mean is 2.50×10^{-05}). On the other hand, in the NCS with the non-switching controller, all global minimum values are within 7.82×10^{-09} and 4.46×10^{-08} (the mean is 8.75×10^{-09}). Recall that, as discussed in Section 3 the minimum value of (4) is $\min f_j = 0$ when the attacked system is perfectly identified. So, the higher order of the global minimum values caused by the switching controller also demonstrates the effectiveness of the proposed countermeasure. From the attacker point o view, these higher global minimum values may be an indicative that the Active System Identification attack was not effective in obtaining the system model. In this sense, the attacker must hesitate to launch a model-dependent attack with the information gathered by the Active System Identification attack.

The impact of the proposed countermeasure in the attack can also be verified in the pole-zero maps shown in Figure 6. Figure 6(a) shows the zeros and poles of the open-loop transfer functions estimated by 100 simulations with the non-switching controller. Figure 6(b) shows the zeros and poles of the open-loop transfer functions estimated by the simulations using the switching controller. Note that, in the simulations with the non-switching controller, the estimated zeros and poles accurately meet the actual zeros and poles of $G_1(z)$. On the other hand, Figure 6(b) shows that when the proposed countermeasure is used, the estimated zeros and poles are spread and do not concur for the actual zeros and poles of $G_1(z)$ and $G_2(z)$ – *i.e.* the open-loop transfer functions of the two subsystems of the SLS.

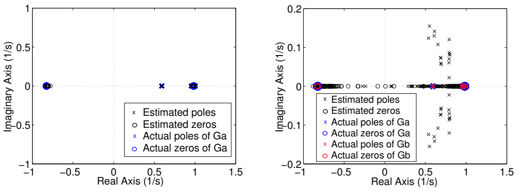
The spreading of the estimated poles and zeros in Figure 6(b), the inaccuracy of the estimated coefficients shown in Figure 5, and the higher global minimum values found by the BSA demonstrate the effectiveness of the switching controllers as a countermeasure for the Active System Identification attack of [6]. With the proposed countermeasure, it is possible to state that the model obtained by the attacker is imprecise/ambiguous in such a way that, with the obtained information, the attacker may hesitate to launch a covert/model-dependent attack. So, O.II (Section 4) is met.

5.3. Performance as a Controller

In this section, the performance of the proposed countermeasure is analyzed from the control perspective, in order to identify the possible impacts that it may produce in the control of the plant. To do so, the following aspects are evaluated: stability; overshoot; and settling time. Considering these aspects, the performance of the switching controller is compared with the performance of the non-switching controller. Given the stochastic nature of the switching controller described in Section 5.1, the mentioned aspects are evaluated through a set of 100,000 simulations.

In Figure 7, there are shown the responses of both NCSs in the time domain. The responses of the NCS with the proposed countermeasure are represented by the highlighted area. The bounds of this area are drawn based on the maximum and minimum values of the output of the plant, considering all 100,000 simulations. The non-stochastic response of the NCS using the non-switching controller is represented in Figure 7 by the red line. Note that, up to $t = 0.4s$ the responses using the switching controller are the same as the response with the non-switching controller. This is caused by the minimum dwell time of $0.4s$, set by the minimum number of sampling intervals that the system have to remain in the same state, defined as $a = 20$ samples. Based on Figure 7, it is possible to verify that the NCS with the proposed countermeasure is stable, the output of the plant converges to the set point $(1rad/s)$ without stationary error, and it does not present overshoots. In these aspects, from the control perspective, the proposed countermeasure presents the same performance as the non-switching controller.

On the other hand, due to the successive switchings, it is possible to verify in Figure 7 that the settling time of the proposed countermeasure is higher than the settling time provided by the non-switching controller. The deterministic settling time of the NCS with the non-switching controller is $2.4s$. The settling time t_s provided by the switching



(a) NCS without the countermeasure (b) NCS using the countermeasure

Figure 6: Zeros and poles estimated by the attack.

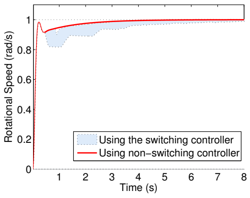


Figure 7: Response of the systems in the time domain.

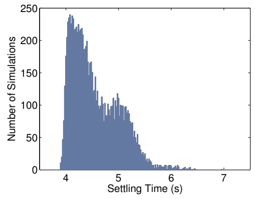


Figure 8: Settling times with the proposed countermeasure.

controller is stochastic and depends on the random sequence of dwell times occurred before t_s. The settling times of all 100,000 simulations using the switching controller are represented in the histogram of Figure 8. The minimum and maximum settling times are $3.90s$ and $6.96s$, respectively, and the mean is $4.555{\pm}0.009s$ (confidence interval of 95%).

The performance of the proposed countermeasure, from the control perspective, is satisfactory and indicates the feasibility of meeting O.I and O.II, simultaneously. In these simulations, the control provided by the switching controller presents a performance similar to the performance of the non-switching controller. The primary requirement of O.I – *i.e.* stability – is met, as well as the requirement of not causing overshoots on the plant. However, the simulations indicate an increase in the settling time of the system, which may not be an issue, but have to be analyzed depending on the specific process being controlled.

6. Conclusion

We propose the use of a randomly switching controller as a countermeasure for the Active System Identification attack, in case of other conventional countermeasures – such as encryption and network security policies – fail. It is demonstrated that this countermeasure is capable to mitigate the mentioned attack, making the model obtained by the attacker imprecise and ambiguous. At the same time, the simulations demonstrate that the performance of this countermeasure is satisfactory from the control perspective. Considering the control aspects, in general, the countermeasure presents a performance similar to the performance of a non-switching controller, with an increase in the system's settling time. Therefore, the tradeoff between hindering the identification attack and increasing the settling time – which, depending on the plant, is not necessarily a drawback – must be taken into account when deciding for using this countermeasure. As future work, we plan to assess the performance of this countermeasure against other system identification attacks/algorithms. Also, we encourage the development of an heuristic or an analytical method capable to provide control functions and switching rules that maximize the performance of the countermeasure in both mentioned objectives – *i.e.* comply with the plant's control requirements; and hinder the identification process.

Acknowledgments

Research partially supported by CNPq/FAPERJ (Brazil).

References

[1] R. Smith, "A decoupled feedback structure for covertly appropriating networked control systems," in *Proceedings of the 18th IFAC World Congress 2011*, vol. 18, no. 1. IFAC-PapersOnLine, 2011.

[2] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen, "Cyber security of water scada systems part i: analysis and experimentation of stealthy deception attacks," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1963–1970, 2013.

[3] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "A secure control framework for resource-limited adversaries," *Automatica*, vol. 51, pp. 135–148, 2015.

[4] R. S. Smith, "Covert misappropriation of networked control systems: Presenting a feedback structure," *Control Systems, IEEE*, vol. 35, no. 1, pp. 82–92, 2015.

[5] A. O. de Sa, L. F. R. da Costa Carmo, and R. C. S. Machado, "Covert attacks in cyber-physical control systems," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1641–1651, Aug 2017.

[6] ——, "Bio-inspired active attack for identification of networked control systems," in *10th EAI Int. Conference on Bio-inspired Information and Communications Technologies*. ACM, 3 2017, pp. 1–8.

[7] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn, "Nist special publication 800-82, revision 2: Guide to industrial control systems (ics) security," *Gaithersburg, MD, USA: National Institute of Standards and Technology*, 2015.

[8] Z.-H. Pang and G.-P. Liu, "Design and implementation of secure networked predictive control systems under deception attacks," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1334–1342, 2012.

[9] P. Civicioglu, "Backtracking search optimization algorithm for numerical optimization problems," *Applied Mathematics and Computation*, vol. 219, no. 15, pp. 8121–8144, 2013.

[10] R. Kennedy, J. e Eberhart, "Particle swarm optimization," in *Proceedings of 1995 IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[11] E. Skafidas, R. J. Evans, A. V. Savkin, and I. R. Petersen, "Stability results for switched controller systems," *Automatica*, vol. 35, no. 4, pp. 553–564, 1999.

[12] J. Wang, "Identification of switched linear systems," Ph.D. dissertation, University of Alberta, 2013.

[13] H. Lin and P. J. Antsaklis, "Stability and stabilizability of switched linear systems: a survey of recent results," *IEEE Transactions on Automatic control*, vol. 54, no. 2, pp. 308–322, 2009.

[14] A. S. Morse, "Supervisory control of families of linear set-point controllers-part i. exact matching," *IEEE Transactions on Automatic Control*, vol. 41, no. 10, pp. 1413–1431, 1996.

[15] J. P. Hespanha and A. S. Morse, "Stability of switched systems with average dwell-time," in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, vol. 3. IEEE, 1999, pp. 2655–2660.

[16] G. Zhai, B. Hu, K. Yasuda, and A. N. Michel, "Qualitative analysis of discrete-time switched systems," in *American Control Conference, 2002. Proceedings of the 2002*, vol. 3. IEEE, 2002, pp. 1880–1885.

APPENDIX G

Evaluation on Passive System Identification and Covert Misappropriation attacks in Large Pressurized Heavy Water Reactors

Alan Oliveira de Sá
Admiral Wandenkolk Instruction Center, Brazilian Navy; and Institute of Mathematics/NCE, Federal University of Rio de Janeiro, RJ, Brazil
Email: alan.oliveira.sa@gmail.com

Luiz Fernando Rust da C. Carmo
National Institute of Metrology, Quality and Technology; and Institute of Mathematics/NCE, Federal University of Rio de Janeiro, RJ, Brazil
Email: lfrust@inmetro.gov.br

Raphael C. Santos Machado
National Institute of Metrology, Quality and Technology; and Rio de Janeiro Federal Center for Technological Education, RJ, Brazil
Email: rcmachado@inmetro.gov.br

***Abstract*—The recent literature on nuclear science demonstrates the feasibility and the benefits of controlling large Pressurized Heavy Water Reactors (PHWR) through Networked Control Systems (NCS). However, the use of NCSs in PHWRs may also expose such critical systems to threats launched from the cyber domain. In the present paper, we propose a novel combination of two cyber-physical attacks (Passive System Identification attack and Covert Misappropriation attack) and evaluate their impact in a PHWR. The results indicate that, with this two attacks, the attacker is able to manipulate the power of the PHWR achieving, at the same time, a high degree of covertness. Moreover, the outcomes suggest the sensitivity and accuracy required for a monitoring system to detect this kind of attack, which may be considered in the development of standards and requirements for PHWR monitoring systems.**

***Index Terms*—Security, Pressurized Heavy Water Reactor, Networked Control Systems, System Identification, Covert Attack.**

1. Introduction

The use of communication networks to integrate controllers and physical plants in industry aims to reduce costs as well as improve management and operational capabilities [1]. For the same reasons, there is a research effort regarding the use of Networked Control Systems (NCS) in large Pressurized Heavy Water Reactors (PHWR) [2], [3], [4]. According to [2], the main benefits of using NCS technologies in a PHWR are: reduced wiring between sensors and controllers; greater instrumentation flexibility; better diagnostics provided by the digital connectivity, which is useful to identify erros in the system.

However, at the same time that the use of NCSs brings several advantages, the use of communication networks to integrate controllers and physical plants can also expose these systems to cyber threats [1], [5]. The most emblematic example of these cyber-physical threats is the Stuxnet worm [6], which targeted the uranium enrichment centrifuges of the Iranian nuclear program. In this context, there is a research effort to characterize vulnerabilities and solve security issues in NCSs [1], [5], [7], [8], [9], [10], [11]. In [7], [10], it is proposed a Covert Misappropriation attack, where the malicious agent uses the knowledge about the plant model to inject false data in the NCS without being noticed by the control system. Although the covertness of this attack is necessarily model dependent, the author does not describe how the attacker obtains the model of the plant. As a premise, it is just assumed that the attacker knows the plant model to design the attack. A more recent work [1] proposed the Passive System Identification attack. The aim of this attack – also classified as a Cyber-physical Intelligence attack [1] – is to estimate the NCS models and should be used to support the Covert Misappropriation attack. However, a complete attack resulting from the combination of these two attacks was not yet evaluated. Therefore, in this work, we propose and evaluate the new joint action of the two aforementioned attacks in a PHWR. The complete offensive is performed in two steps:

S-I The Passive System Identification (PSI) attack: which is performed to obtain an accurate model of the attacked system – in this case, the model to be obtained is the transfer function of a PHWR zone;

S-II The Covert Misappropriation attack: where a Man-in-the-Middle (MitM) intercepts the NCS communication links and injects false information in both forward and feedback streams. The falsa data injected in the network is computed based on the model estimated by the PSI attack, in order to make the attacker as covert as possible during the manipulation of the PHWR power.

It is worth mentioning that the purpose of this work is not to facilitate cyber attacks in PHWRs. The aim of the present paper is to show how stealth a Covert Misappropriation attack can be when designed based on the information provided by a PSI attack. With this study, we aim to encourage the research for techniques capable of effectively measuring and identifying this kind of attack in PHWRs – and other critical cyber-physical systems. From the PHWR owner perspective, it is worth knowing the possible impacts and what should be expected as evidence if such complete

978-1-5386-2497-5/18/$31.00 ©2018 IEEE

and sophisticated attack occurs. This information may contribute with the development of standards and requirements for PHWR monitoring systems.

The next sections of this work are organized as follows. Section 2 provides a brief explanation on the attacked PHWR. Section 3 describes the PSI attack. Section 4 explains the Covert Misappropriation attack. Section 5 shows the results of a set of simulations performed to evaluate the impact of the complete attack in a PHWR zone. Finally, Section 6 brings the conclusions of this work.

2. Networked PHWR

A PHWR is a type of nuclear reactor whose fuel is natural Uranium and, therefore, uses Heavy Water (D_2O, or ^2H_2O) as coolant and moderator. It is known that, in nuclear power plants, the power produced by the reactor is controlled through adjustments in its reactivity. Depending on the type of the reactor, such reactivity adjustment may be implemented, for instance, using light water, control rods or liquid poisons. In the PHWR considered in this work, the reactivity input is controlled using light water. According to [12], in a nuclear reactor, the control system computes the signals that drive the reactivity control devices – *i.e.* the actuators – used to change the reactivity input to the reactor. When the reactivity is increased, the neutron flux also increases and so does the burn-up of fissile material. On the other hand, when the reactivity input is reduced, the burn-up of fissile material also reduces.

The literature on nuclear science [2], [3], [4] demonstrates the feasibility of controlling a large PHWR through an NCS. In [2], the satisfactory control of an example of large PHWR is achieved using a state-feedback controller and a 100 Mbps Ethernet LAN, using UDP/IP. More recently, in [3], [4], the authors demonstrate the feasibility of using PID algorithms to control a large PHWR through an UDP/IP Ethernet communication.

As in [2], [3], [4], the model of the reactor considered in this work belongs to an Indian PHWR of 540 MWe. According to the analysis provided in [2] this 540 MWe PHWR is constituted by 14 zones which can be modeled as 14 independent Single-Input-Single-Output (SISO) Systems. For the sake of simplicity, we choose one of these 14 zones to assess the performance and impact of the joint operation of the PSI attack and the Covert Misappropriation attack. The model of the attacked zone and its PID controller, both obtained from [4], are defined by (1) and (2), respectively:

$$G(z) = \frac{0.0001889z}{z^2 - 1.289z + 0.2891}, \tag{1}$$

$$C(z) = k_p + T_s k_i \left(\frac{z}{z-1}\right) + \frac{k_d}{T_s}\left(\frac{z-1}{z}\right), \tag{2}$$

wherein the sample time is $T_s = 500ms$, $k_p = 348.52$, $k_i = 17.25$ and $k_d = 10.79$. The transfer function (1) of this PHWR zone is obtained based on practical plant data [3], [4]. In a PHWR, the power of a specific zone is controlled through a control valve that is used to fill and drain water from a compartment. Therefore, the transfer function (1) describes the relationship that exists between the power P and the valve input v in zone 6 of the PHWR reported in [3], [4], which has a full power of $132.75MWt$. As in [4], the signal applied to the controller's set point is a ramp increased by $0.66MWt/s$ – *i.e.* 0.5% of the full power – for $10s$ and held constant after that. The ramp increase rate herein used is the maximum permitted for this PHWR class.

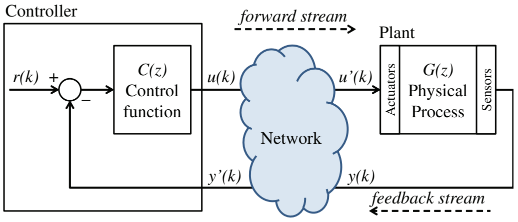


Figure 1: Networked Control System [1]

3. Passive System Identification Attack

This section describes the PSI attack [1], in order to present the basic concepts necessary to study the potential impacts of the joint operation of this attack and the Covert Misappropriation attack in a large PHWR. The goal of the PSI attack is to obtain an accurate estimate of the coefficients of the PHWR model, $G(z)$, using the data gathered from both forward and feedback streams of the NCS presented in Figure 1. To this end, the attacker executes two steps:

- STEP-I: The plant's input $u'(k)$ as well as its output $y(k)$ are eavesdropped during a monitoring period T.
- STEP-II: Knowing $y(k)$ and $u'(k)$, the plant model $G(z)$ is estimated. The attacker applies $u'(k)$ to the input of an estimated model $G_e(z)$ and adjusts its coefficients until the output $\hat{y}(k)$ of $G_e(z)$ converges to $y(k)$. In [1], the coefficients of $G_e(z)$ are adjusted using a bio-inspired metaheuristic, namely: Backtracking Search Algorithm (BSA) [13].

Therefore, to estimate $G(z)$ in STEP-II, the attacker applies $u'(k)$ into the estimated transfer function represented by (3):

$$G_e(z) = \frac{\mathcal{Z}[\hat{y}(k)]}{\mathcal{Z}[u'(k)]} = \frac{\alpha_n z^n + \alpha_{n-1} z^{n-1} + ... + \alpha_1 z^1 + \alpha_0}{z^m + \beta_{m-1} z^{m-1} + ... + \beta_1 z^1 + \beta_0}, \tag{3}$$

wherein $\hat{y}(k)$ is the output provided by the estimated model $G_e(z)$, and $\mathcal{Z}$ represents the Z-transform operation. Note that, $[\alpha_n, \alpha_{n-1}, ..., \alpha_1, \alpha_0, \beta_{m-1}, \beta_{m-2}, ...\beta_1, \beta_0]$ is the set of coefficients of $G(z)$ that the PSI attack aims to discover, wherein n defines the numerator's order and m the denominator's order. Therefore, to obtain the model of the actual plant $G(z)$, the parameters of $G_e(z)$ are modified and adapted until $\hat{y}(k)$ converges to $y(k)$. To do so, the numerical optimization process performed by the BSA iteratively adjusts the parameters of $G_e(z)$ in order to minimize a fitness function f, until $G_e(z)$ meets $G(z)$. The coordinates $x_j = [\alpha_{n,j}, \alpha_{n-1,j}, ...\alpha_{1,j}, \alpha_{0,j}, \beta_{m-1,j}, \beta_{m-2,j}, ...\beta_{1,j}, \beta_{0,j}]$ of each individual j of the BSA are assigned as the coefficients of an estimated model $G_e(z)$. Each individual j is, therefore, evaluated by computing its fitness f_j according to (4):

$$f_j = \frac{\sum_{k=0}^{S} [y(k) - \hat{y}_j(k)]^2}{S}, \tag{4}$$

wherein S is the total amount of samples captured by the attacker during the monitoring period T of STEP-I. The signal $\hat{y}_j(k)$ is the output of $G_e(z)$ (3) when its coefficients are defined as x_j. From (4) it is possible to see that $\min f_j = 0$ if $y(k) = \hat{y}_j(k)$. This result is achieved whenever $[\alpha_{n,j}, \alpha_{n-1,j}, \ldots, \alpha_{1,j}, \alpha_{0,j}, \beta_{m-1,j}, \beta_{m-2,j}, \ldots, \beta_{1,j}, \beta_{0,j}] = [\alpha_n, \alpha_{n-1}, \ldots, \alpha_1, \alpha_0, \beta_{m-1}, \beta_{m-2}, \ldots, \beta_1, \beta_0]$ or, in other words, when $G_e(z) = G(z)$.

4. Covert Misappropriation attack

The attack for covert misappropriation of NCSs is presented in [10]. The aim of this attack is to allow a Man-in-the-Middle (MitM) to perform malicious control actions in a plant without being perceived from the perspective of the signals arriving at the original networked controller. Figure 2 shows an implementation of such covert misappropriation attack, based on the attack architecture proposed in [10], wherein $A(z)$ is the covert controller and $G'(z)$ is an estimated model of the plant, which the attacker is supposed to know. The input $\delta(k)$ drives the attacker's feedback loop and allows the MitM to lead the actual plant output to the desired offset.

Note in Figure 2 that, in the forward stream, the MitM performs a data injection attack in which the input of the plant is given by (5):

$$u'(k) = \psi(k) + u(k), \tag{5}$$

wherein $\psi(k)$, referred to as attack signal, is defined by (6):

$$\psi(k) = \delta(k) * \mathcal{Z}^{-1}\left[\frac{A(z)}{G'(z)A(z)+1}\right], \tag{6}$$

wherein $\mathcal{Z}$ represents the Z-transform operation. Therefore, considering this data injection, the output of the plant $Y(z) = \mathcal{Z}[y(k)]$ is given by (7):

$$Y(z) = \mathcal{Z}[\psi(k) + u(k)]\, G(z). \tag{7}$$

Yet, from Figure 2, it is possible to see that in the feedback stream the MitM also implements a data injection attack in order to manipulate the controller's input signal $Y'(z) = \mathcal{Z}[y'(k)]$. With this manipulation, considering (7), the signal that arrives at the controller is defined as (8):

$$\begin{aligned} Y'(z) &= Y(z) - \mathcal{Z}[\psi(k)]\, G'(z) \\ &= \mathcal{Z}[u(k) + \psi(k)]\, G(z) - \mathcal{Z}[\psi(k)]\, G'(z). \end{aligned} \tag{8}$$

In this sense, if the attacker perfectly knows the model of the actual plant – *i.e.* if $G'(z) = G(z)$ –, then (8) can be rewritten as (9):

$$\begin{aligned} Y'(z) &= G(z)\mathcal{Z}[u(k) + \psi(k)] - G(z)\mathcal{Z}[\psi(k)] \\ &= G(z)\mathcal{Z}[u(k) + \psi(k) - \psi(k)] \\ &= G(z)\mathcal{Z}[u(k)] \end{aligned} \tag{9}$$

which, from the perspective of the controller, means that the plant is behaving as in a normal operation, where $Y(z) = \mathcal{Z}[u(k)]\, G(z)$. In other words, by analyzing $y'(k)$, one should assume that $y'(k) = y(k)$, $u'(k) = u(k)$ and, therefore, there is no data injection attack in the NCS.

5. Results

This section presents an evaluation on the performance of the attacks described in Sections 3 and 4, when launched together against the PHWR zone specified in Section 2. The results of both attacks are obtained through simulations using MATLAB/SIMULINK. First, the PSI attack is performed, in order provide the attacker with an estimate of the model $G'(z)$ of the attacked PHWR zone. After that, the Covert Misappropriation attack is carried out using the data that the PSI attack provided.

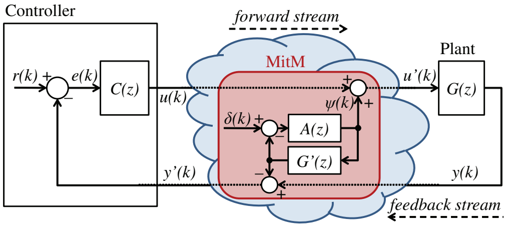


Figure 2: Covert Misappropriation attack.

As previously discussed, the PSI attack aims to estimate the coefficients of the attacked plant which, according to (1), are: $\alpha_1 = 1.889 \times 10^{-4}$, $\beta_1 = -1.289$ and $\beta_0 = 0.2891$. The monitoring time of the attack is $T = 200s$, starting when the power P of the attacked zone begins to increase – *i.e.* when the ramp setpoint specified in Section 2 starts. The BSA configurations used in the simulations of this work are the same as those used in [1]: the lower and upper limits of each search space dimension are -10 and 10, respectively; the number of individuals in the BSA population is 100; and $\eta = 1$ (in the BSA, η is used to define the amplitude of the displacement of the individuals). The accuracy of the PSI attack is analyzed considering three different numbers of iterations of the BSA: 200, 400 and 600. For each number of iterations, there were executed 100 attack simulations.

Figure 3 shows the values of α_0, β_1 and β_0 estimated by 100 attack simulations for each of the mentioned numbers of BSA iterations. It is possible to verify that the accuracy achieved by the PSI attack increases as the BSA iterations rises. It is noticeable that, with more BSA iterations the estimated values of α_0, β_1 and β_0 become more concentrated and close to their actual values. Additionally, Table 1 shows the statistics of this PSI attack in the PHWR zone. From this table, it is also possible to see that, when the attacker increases the number of BSA iterations, he/she improves the performance of the attack – the mean estimated coefficients become closer to their actual values and the standard deviation decreases. Note that, a high level of accuracy is achieved when the attacker runs the BSA for 600 iterations.

TABLE 1: Statistics of the PSI attack in the PHWR zone

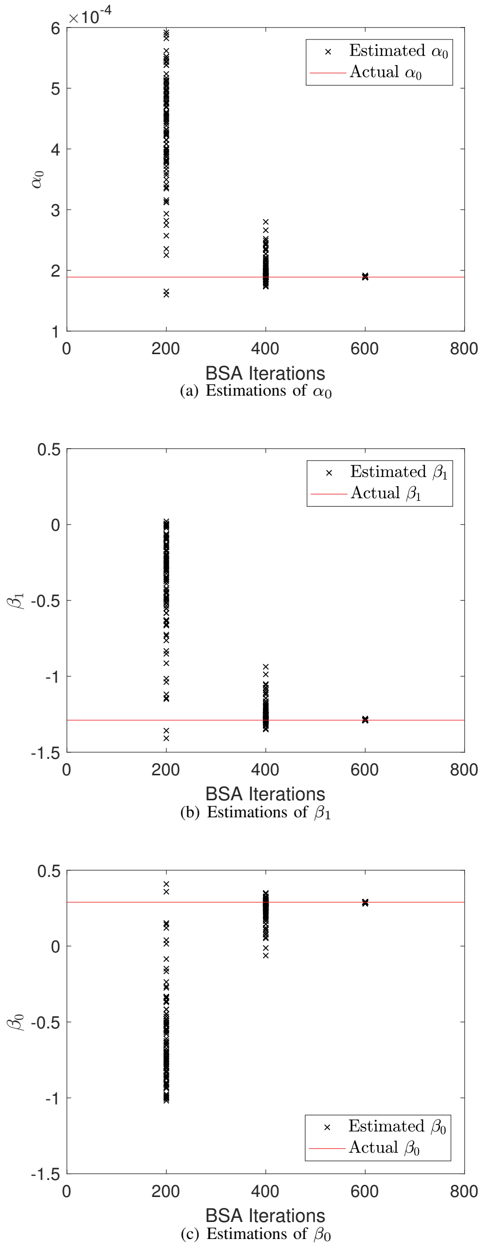
	Mean			Standard Deviation		
BSA Iterations	α_1 $(\times 10^{-4})$	β_1	β_0	α_1 $(\times 10^{-6})$	β_1 $(\times 10^{-2})$	β_0 $(\times 10^{-2})$
200	4.331	-0.383	-0.617	87.64	31.18	31.18
400	2.013	-1.242	0.242	19.55	7.51	7.51
600	1.890	-1.289	0.288	0.40	0.15	0.15

To evaluate how the accuracy provided by the PSI attack may contribute for the covertness of the misappropriation attack, $G'(z)$ is configured using the mean coefficients presented in Table 1. The aim of this covert misappropriation attack is to reduce 1MWt of attacked zone power, modifying as less as possible the controller input signal $y'(k)$ (comparing with a normal operation scenario). The input $\delta(k)$ of the MitM is a ramp signal that starts at $30s$, decreases at the rate of -0.2 during $5s$ and then is kept steady. The covert controller $A(z)$ computes the same PID function defined in (2), however, using the following configuration: $k_p = 310$, $k_i = 40$ and $k_d = 10$.

Figure 4 shows the responses of the PHWR zone with and without the influence of the Covert Misappropriation attack, considering the worst estimated model – *i.e.* when $G'(z)$ is estimated through 200 BSA iterations. The time when the covert misappropriation begins is indicated by the dotted line, placed at $30s$. It is possible to see that the MitM is capable of making the output of the plant $y(k)$ converges to a power $1MWt$ lower than in its normal operation (*i.e.* without the misappropriation attack). Additionally, by comparing the controller input signals $y'(k)$ with and without the attack, it is possible to verify that both are quite similar. It indicates the high degree of covertness achieved using the model estimated by the PSI attack – even executing only 200 BSA iterations. When $G'(z)$ is estimated using 400 and 600 iterations, the covertness of the misappropriation attack is better than the covertness obtained with 200 iterations. The difference between $y'(k)$ with attack and $y'(k)$ without attack decreases as the number of BSA iterations increases. It is difficult to perceive the differences of covertness if the three cases (using 200, 400 and 600 iterations) are represented as in Figure 4. Thus, to compare the covertness of these three attack conditions, we compute $\xi(k)$ (10):

$$\xi(k) = y'_A(k) - y'_N(k). \quad (10)$$

wherein $y'_A(k)$ and $y'_N(k)$ are the controller input signal $y'(k)$ with and without the misappropriation attack, respectively. Figure 5 shows the differences $\xi(k)$ in the controller



(a) Estimations of α_0

(b) Estimations of β_1

(c) Estimations of β_0

Figure 3: Estimations of α_0, β_1 and β_0 with different numbers of BSA iterations.

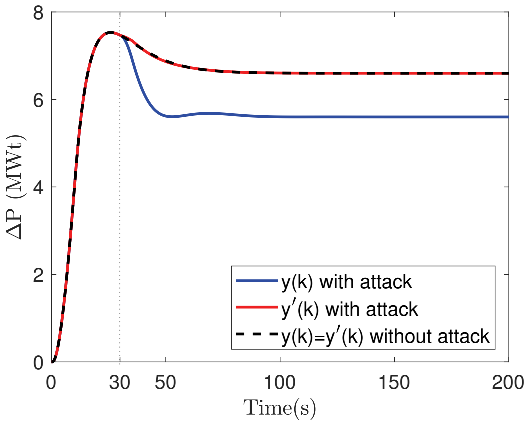


Figure 4: PHWR responses with and without the Covert Misappropriation attack ($G'(z)$ estimated in 200 iterations).

input, considering misappropriation attacks where $G'(z)$ is estimated through 200, 400 and 600 BSA iterations.

Note in Figure 5 that the highest amplitude of $\xi(k)$ is obtained when $G'(z)$ is estimated with 200 iterations. With 200 iterations $\max|\xi(k)| = 3.9 \times 10^{-2} MWt$ (during the transient regime of the attack), while with 400 iterations $\max|\xi(k)| = 3.8 \times 10^{-3} MWt$. From the attacker perspective, the best covertness occurs when $G'(z)$ is estimated with 600 iterations. In this case, $\max|\xi(k)| = 2.9 \times 10^{-5} MWt$, which is a quite small deviation in the controller input, considering the magnitude of the zone power.

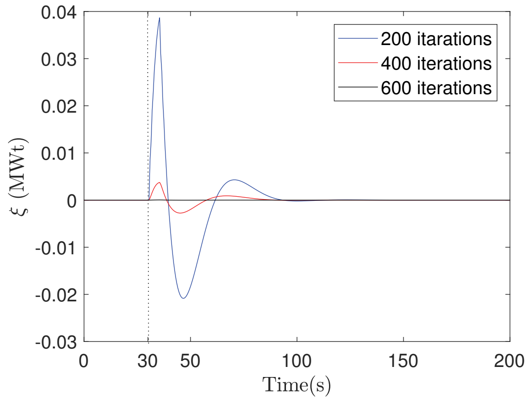


Figure 5: Differences in the controller's input signal.

These results provide an idea on how covert and harmful may be this joint attack in a PHWR. The attacker is able to achieve his/her goal, reducing 1MWt of attacked zone power, while causing low levels of $\xi(k)$ – especially when the PSI attack is performed with 600 iterations. These low levels of $\xi(k)$ may be considered in the development of standards and requirements for PHWR monitoring systems.

6. Conclusion

This paper brings the novel joint operation of the PSI attack and the Covert Misappropriation attack against a PHWR. The results show that the PSI attack can be considered a powerful tool for the design of such covert misappropriation attack. A high degree of covertness is achieved when the PSI attack is performed with 600 iterations. In this case, the results demonstrate that the attacker is able to reduce 1MWt of attacked zone power, causing an interference $\leq 2.9 \times 10^{-5} MWt$ in the controller input. Moreover, the outcomes of $\xi(k)$ provide a quantitative assessment on the accuracy and sensitivity required for a monitoring system to detect such covert and harmful attack in a PHWR. As future work, we encourage the evaluation of techniques to measure and identify the occurrence of this kind of attack, as well as develop countermeasures to mitigate it in situations where the attacker has access to the information that is transmitted through the NCS links.

Acknowledgments

The present work is partially supported by the following Brazilian research agencies: FAPESP, FAPERJ and CNPq.

References

[1] A. O. de Sa, L. F. R. da Costa Carmo, and R. C. S. Machado, "Covert attacks in cyber-physical control systems," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1641–1651, Aug 2017.

[2] M. Das, R. Ghosh, B. Goswami, A. Gupta, A. Tiwari, R. Balasubrmanian, and A. Chandra, "Network control system applied to a large pressurized heavy water reactor," *IEEE Transactions on Nuclear Science*, vol. 53, no. 5, pp. 2948–2956, 2006.

[3] S. Dasgupta, A. Routh, S. Banerjee, K. Agilageswari, R. Balasubramanian, S. Bhandarkar, S. Chattopadhyay, M. Kumar, and A. Gupta, "Networked control of a large pressurized heavy water reactor (phwr) with discrete proportional-integral-derivative (pid) controllers," *IEEE Transactions on Nuclear Science*, vol. 60, no. 5, pp. 3879–3888, 2013.

[4] S. Dasgupta, K. Halder, S. Banerjee, and A. Gupta, "Stability of networked control system (ncs) with discrete time-driven pid controllers," *Control Engineering Practice*, vol. 42, pp. 41–49, 2015.

[5] A. O. de Sa, L. F. R. da Costa Carmo, and R. C. S. Machado, "Bio-inspired active system identification: a cyber-physical intelligence attack in networked control systems," *Mobile Networks and Applications*, pp. 1–14, 2017.

[6] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *Security & Privacy, IEEE*, vol. 9, no. 3, pp. 49–51, 2011.

[7] R. Smith, "A decoupled feedback structure for covertly appropriating networked control systems," in *Proceedings of the 18th IFAC World Congress 2011*, vol. 18, no. 1. IFAC-PapersOnLine, 2011.

[8] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen, "Cyber security of water scada systems part i: analysis and experimentation of stealthy deception attacks," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1963–1970, 2013.

[9] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "A secure control framework for resource-limited adversaries," *Automatica*, vol. 51, pp. 135–148, 2015.

[10] R. S. Smith, "Covert misappropriation of networked control systems: Presenting a feedback structure," *Control Systems, IEEE*, vol. 35, no. 1, pp. 82–92, 2015.

[11] A. O. de Sa, L. F. R. da Costa Carmo, and R. C. S. Machado, "A controller design for mitigation of passive system identification attacks in networked control systems," *Journal of Internet Services and Applications*, vol. 9, no. 1, pp. 1–19, Feb 2018.

[12] S. Banerjee, K. Halder, S. Dasgupta, S. Mukhopadhyay, K. Ghosh, and A. Gupta, "An interval approach for robust control of a large phwr with pid controllers," *IEEE Transactions on Nuclear Science*, vol. 62, no. 1, pp. 281–292, 2015.

[13] P. Civicioglu, "Backtracking search optimization algorithm for numerical optimization problems," *Applied Mathematics and Computation*, vol. 219, no. 15, pp. 8121–8144, 2013.

APPENDIX H

Bio-inspired System Identification Attacks in Noisy Networked Control Systems

Alan Oliveira de Sá[1,2],
António Casimiro[3],
Raphael Carlos Santos Machado[4,5], and
Luiz Fernando Rust da Costa Carmo[2,4]

[1] Admiral Wandenkolk Instruction Center, Brazilian Navy, RJ, Brazil
[2] Institute of Mathematics/NCE, Federal University of Rio de Janeiro, RJ, Brazil
[3] Department of Informatics, Faculty of Sciences of the University of Lisboa, Portugal.
[4] National Institute of Metrology, Quality and Technology, RJ, Brazil
[5] Rio de Janeiro Federal Center for Technological Education, RJ, Brazil
alan.oliveira.sa@gmail.com, casim@ciencias.ulisboa.pt,
{rcmachado,lfrust}@inmetro.gov.br

Abstract. The possibility of cyberattacks in Networked Control Systems (NCS), along with the growing use of networked controllers in industry and critical infrastructures, is motivating studies about the cybersecurity of these systems. The literature on cybersecurity of NCSs indicates that accurate and covert model-based attacks require high level of knowledge about the models of the attacked system. In this sense, recent works recognize that Bio-inspired System Identification (BiSI) attacks can be considered an effective tool to provide the attacker with the required system models. However, while BiSI attacks have obtained sufficiently accurate models to support the design of model-based attacks, they have demonstrated loss of accuracy in the presence of noisy signals. In this work, a noise processing technique is proposed to improve the accuracy of BiSI attacks in noisy NCSs. The technique is implemented along with a bio-inspired metaheuristic that was previously used in other BiSI attacks: the Backtracking Search Optimization Algorithm (BSA). The results indicate that, with the proposed approach, the accuracy of the estimated models improves. With the proposed noise processing technique, the attacker is able to obtain the model of an NCS by exploiting the noise as a useful information, instead of having it as a negative factor for the performance of the identification process.

Keywords: Security· Networked Control Systems· Cyber-Physical Systems· System Identification· Backtracking Search Algorithm· Bio-inspired Algorithm

1 Introduction

The use of communication networks to integrate controllers and physical processes in a Networked Control Systems (NCS), such as shown in Figure 1, aims to improve management and operational capabilities, as well as reduce costs [10]. However, this integration also exposes the physical plants to new threats originated in the cyber domain.

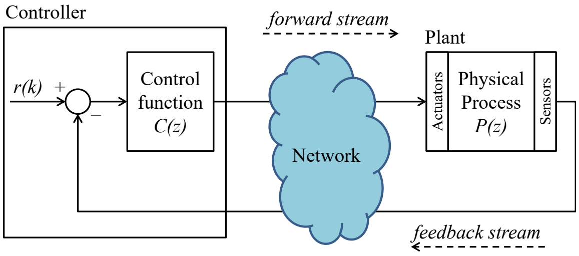


Fig. 1. Networked Control System.

The possibility of sophisticated and large impact attacks in Networked Control Systems (NCS) became unprecedentedly concrete after the launch of the Stuxnet worm [6]. The example of such cyber-physical attack – which is not unique –, along with the growing use of networked controllers in industry and critical infrastructures, has been motivating studies about the cybersecurity of NCSs. In this context, there is a research effort to characterize vulnerabilities, understand attack strategies, and propose security solutions for NCS [1, 3, 7–14].

The literature on cybersecurity of NCSs [1, 9–12, 14] indicates that accurate and covert offensives require high level of knowledge about the models of the attacked system. Exemples of covert attacks that agree with this statement are provided in [11, 12]. In these works the attacks are performed by a man-in-the-middle (MitM), where the attacker needs to know the model of the attacked plant to covertly manipulate the system by injecting false data in both forward and feedback streams. The covertness of the attacks shown in [11, 12] is analyzed from the perspective of the signals arriving to the controller, and depends on the difference between the actual model of the plant and the model known by the attacker. In [1], the authors demonstrate another covert offensive where the attacker, aware of the system's model, injects an attack signal in the NCS to steal water from a canal system located in Southern France.

However, in [1, 11, 12, 14], where the attacks intrinsically require knowledge about the NCS models, it is not described how such knowledge is obtained by the attacker. It is just stated that a model is previously known to subsidize the design of those attacks. More recently, in [9, 10], the authors propose two Bio-inspired System Identification (BiSI) attacks to fill this gap. They demonstrate how the data required to design Denial-of-Service (DoS) or Service Degradation (SD) attacks may be obtained using bio-inspired metaheuristics. Specifically, the attacks proposed in [9, 10] are used to obtain the linear time-invariant (LTI) transfer functions of NCS devices – be it a controller [10], a plant [10], or both in a open loop transfer function [9].

While BiSI attacks have obtained sufficiently accurate models to support the design of model-based attacks, they have demonstrated loss of accuracy in the presence of noisy signals [9]. To overcome this constraint, this work proposes a noise processing technique to improve the accuracy of BiSI attacks in noisy NCSs. With the proposed strategy, an attacker is able to obtain the model of an NCS by exploiting the noise as a useful information, instead of having it as a negative factor for the performance of the identification process. In this paper, the BiSI attack is implemented using the bio-inspired metaheuristic called Backtracking Search Optimization Algorithm (BSA) [2]. It is worth mentioning that the purpose of this work is not to facilitate cyber-attacks in NCSs. With this study, we aim to encourage the research for techniques capable to enhance the security of NCSs against advanced attacks. Moreover, from the NCS owner perspective, it is worth knowing how an attacker can obtain valuable information about the NCS in case of a lack of confidentiality.

The next sections of this work are organized as follows. Section 2 provides a brief description about the BSA. Section 3 explains the novel noise processing strategy for BiSI attacks. Section 4 shows the results obtained when the noise processing strategy herein proposed is used to support a BiSI attack. Finally, Section 5 brings the conclusions of this work.

2 Backtracking Search Algorithm

This section describes the basic concepts of the BSA, in order to provide a clear understanding about the algorithm parameters that are adjusted when implementing a BSA-based BiSI attack. The BSA is a bio-inspired metaheuristic that searches for solutions of optimization problems using the information obtained by past generations [2] – or iterations. According to [2], its search process is metaphorically analogous to the behavior of a social group of animals that, at random intervals returns to hunting areas previously visited for food foraging. The general, evolutionary like, concept of the BSA is shown in Algorithm 1.

Algorithm 1: BSA

```
begin
    Initialization;
    repeat
        Selection-I;
        Generate new population
            Mutation;
            Crossover;
        end
        Selection-II;
    until Stopping Condition;
end
```

At the Initialization stage, the algorithm generates and evaluates the initial population $\mathcal{P}_0$ and sets the historical population $\mathcal{P}_{hist}$. The latter acts as the memory of the BSA.

In the first selection stage (Selection-I), the algorithm randomly determines, based on an uniform distribution U, whether the current population $\mathcal{P}$ should be kept as the new historical population and, thus, replace $\mathcal{P}_{hist}$ (*i.e.* if $a < b \mid a, b \sim U(0,1)$, then $P_{hist} = P$). After that, it shuffles the individuals of $\mathcal{P}_{hist}$.

The mutation operator creates $\mathcal{P}_{mod}$, which is the preliminary version of the new population $\mathcal{P}_{new}$. The computation of $\mathcal{P}_{mod}$ is performed according to (1):

$$\mathcal{P}_{mod} = \mathcal{P} + \eta \cdot \Gamma(\mathcal{P}_{hist} - \mathcal{P}), \tag{1}$$

wherein η is empirically adjusted through simulations and $\Gamma \sim N(0,1)$, with N being a normal standard distribution. Thus, $\mathcal{P}_{mod}$ is the result of the movement of $\mathcal{P}$'s individuals in the directions established by vector $(\mathcal{P}_{hist} - \mathcal{P})$. In order to create the final version of $\mathcal{P}_{new}$, the crossover operator randomly combines individuals from $\mathcal{P}_{mod}$ and $\mathcal{P}$, also following a uniform distribution.

In the second selection stage (Selection-II), the algorithm evaluates the elements of $\mathcal{P}_{new}$ using a fitness function f, selects the elements of $\mathcal{P}_{new}$ with better fitness than the ones in $\mathcal{P}$, and replaces them in $\mathcal{P}$. Hence, $\mathcal{P}$ includes only new individuals that have evolved. The algorithm iterates until the stopping condition is met. When it occurs, the BSA returns the best solution found.

Note that the algorithm has two parameters that are empirically adjusted: the size $|\mathcal{P}|$ of its population $\mathcal{P}$; and η, that establishes the amplitude of the movements of the individuals of $\mathcal{P}$. The parameter η must be adjusted to assign to the algorithm both good exploration and exploitation capabilities. With this parameters set, the BSA is used to search for the global minimum of the fitness function f described in Section 3.

3 Noise Processing Technique for BiSI attacks

The purpose of the technique presented in this section is to use the white gaussian noise that may be present in an NCS – such as in [9] – in favor of a BiSI attack. With this technique, an attacker is able to accurately estimate the models of an NCS by exploiting the noise as a useful information, instead of having it as a negative factor for the performance of the identification process – which happened in previous implementations of BiSI attacks [9].

The first step of the attack is to eavesdrop the input $i(k)$ and output $o(k)$ signals of the device to be identified, represented in Figure 2. The device can be a controller or a plant. The signals are captured during a monitoring period containing T samples.

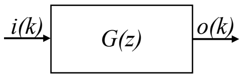


Fig. 2. Device to be identified.

After that, the attacker selects every sample of the eavesdropped input signal $i(k)$ that exceeds a predefined threshold Ω, *i.e.* if (2) is satisfied:

$$i(k) > \Omega, \tag{2}$$

Each sample selected from $i(k)$ according to (2) is referred to as i_n, wherein $n \in \mathbb{Z}_+^*$ is a sequential index number for each selected sample, as exemplified in Figure 3. Additionally, every time that (2) is satisfied, the attacker also stores a portion $o_n(k)$ of the output signal $o(k)$. As represented in Figure 3, each portion $o_n(k)$ selected from $o(k)$ starts when its respective i_n occurs. Each portion $o_n(k)$ encompasses a sequence of τ samples.

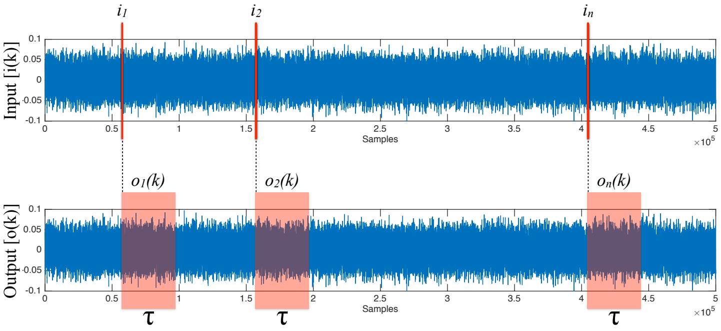


Fig. 3. Selection of noise portions.

After selecting all i_n and $o_n(k)$ from the eavesdropped signals, the attacker computes $\mathcal{I}$ and $\mathcal{O}(k)$ according to (3) and (4), respectively:

$$\mathcal{I} = \frac{\sum_{n=1}^{\mathcal{N}} i_n}{\mathcal{N}}, \tag{3}$$

$$\mathcal{O}(k) = \frac{\sum_{n=1}^{\mathcal{N}} o_n(k)}{\mathcal{N}}, \tag{4}$$

wherein $\mathcal{N}$ is the index number of the last sample i_n obtained from $i(k)$ based on (2). In the present approach, $\mathcal{I}$ corresponds to the amplitude of an impulse signal $\mathcal{I}(k)$ (5) that, when applied to $G(z)$, produces $\mathcal{O}(k)$ – the impulse response function of $G(z)$.

$$\mathcal{I}(k) = \mathcal{I}\delta(k). \tag{5}$$

Now, to estimate $G(z)$, the attacker applies $\mathcal{I}(k)$ to the input of an estimated model $G_e(z)$ defined by (6):

$$G_e(z) = \frac{\mathcal{Z}[\hat{\mathcal{O}}(k)]}{\mathcal{Z}[\mathcal{I}(k)]} = \frac{\alpha_p z^p + \alpha_{p-1} z^{p-1} + ... + \alpha_1 z^1 + \alpha_0}{z^q + \beta_{q-1} z^{q-1} + ... + \beta_1 z^1 + \beta_0}, \tag{6}$$

wherein $\hat{\mathcal{O}}(k)$ is the output provided by the estimated model $G_e(z)$, and $\mathcal{Z}$ represents the Z-transform operation. See that, $[\alpha_p, \alpha_{p-1}, ..., \alpha_1, \alpha_0, \beta_{q-1}, \beta_{q-2}, ...\beta_1, \beta_0]$ is the set of coefficients of $G(z)$ that the BiSI attack aims to discover, wherein p and q represent the order of the numerator and denominator, respectively. Therefore, to obtain the model of the actual device $G(z)$, the parameters of the estimated model $G_e(z)$ are modified and adapted until the output $\hat{\mathcal{O}}(k)$ of $G_e(z)$ converges to $\mathcal{O}(k)$. To do so, the BSA iteratively adjusts the parameters of $G_e(z)$ by minimizing a fitness function f, until $G_e(z)$ meets $G(z)$. The coordinates $x_j = [\alpha_{p,j}, \alpha_{p-1,j}, ...\alpha_{1,j}, \alpha_{0,j}, \beta_{q-1,j}, \beta_{q-2,j}, ...\beta_{1,j}, \beta_{0,j}]$ of each individual j of the BSA are assigned as the coefficients of an estimated model $G_e(z)$. The fitness f_j of each individual j of the BSA is computed according to (7):

$$f_j = \frac{\sum_{k=1}^{\tau} \left[\mathcal{O}(k) - \hat{\mathcal{O}}_j(k)\right]^2}{\tau}. \tag{7}$$

Recall, from Figure 3, that τ is the number of samples contained in each portion $o_n(k)$ of $o(k)$, and, therefore, is also the number of samples contained in $\mathcal{O}(k)$ and $\hat{\mathcal{O}}_j(k)$. The signal $\hat{\mathcal{O}}_j(k)$ is the output of $G_e(z)$ (6) when its coefficients are defined as x_j. From (7) it is possible to see that $\min f_j = 0$ if $\mathcal{O}(k) = \hat{\mathcal{O}}_j(k)$. This result is achieved whenever $[\alpha_{p,j}, \alpha_{p-1,j}, \ldots, \alpha_{1,j}, \alpha_{0,j}, \beta_{q-1,j}, \beta_{q-2,j}, \ldots, \beta_{1,j}, \beta_{0,j}]$ $= [\alpha_p, \alpha_{p-1}, \ldots, \alpha_1, \alpha_0, \beta_{q-1}, \beta_{q-2}, \ldots, \beta_1, \beta_0]$ or, in other words, when $G_e(z) = G(z)$.

Algorithm 2: BiSI attack with the noise processing strategy

```
begin
    Eavesdrop $i(k)$ and $o(k)$ during $T$ samples;
    Noise Processing
        Select all $i_n$ and the respective $o_n(k)$, $\forall i(k) > \Omega$;
        Compute $\mathcal{I}(k)$ and $\mathcal{O}(k)$ according to (3), (4) and (5);
    end
    Execute BSA, using $\mathcal{I}(k)$ and $\mathcal{O}(k)$ to find $G(z)$.
end
```

The Algorithm 2 briefly describes the complete BiSI attack with the proposed noise processing strategy. Albeit the BiSI attack herein proposed uses the same bio-inspired metaheuristic used in [9] (*i.e.*, the BSA, concisely described in Section 2 as in [9]), its is worth mentioning the differences from the present attack and the BiSI attack of [9]:

- In [9] the attacker injects an attack signal in the system to identify its transfer function. In that approach, the presence of noise affects the ability of the attack to learn the system model from the outputs caused by the attack signal. On the other hand, in the present work, the attacker does not injects an attack signal in the system. Conversely, the attacker passively collects the noisy signals and use them to estimate the system transfer function.

- The approach presented in [9] does not use the Noise Processing technique herein proposed.

4 Results

This section presents an evaluation on the performance of the BiSI attack with the noise processing strategy presented in Section 3. The model of the attacked device – *i.e.*, the device to be identified – is represented by (8). In practice, such second order transfer function can represent, for instance, a DC motor [4] or a lighting system [5] (among other systems). However, it is worth mentioning that, depending on the system characteristics, the coefficients of such plants can be different from the example defined by (8).

$$G(z) = \frac{\mathcal{Z}[o(k)]}{\mathcal{Z}[i(k)]} = \frac{2}{z - 0.9}. \tag{8}$$

The sample rate is 50 samples/s, and the noise measured in the input of $G(z)$ is a white gaussian noise $w(k) \sim N(\mu, \sigma)$, wherein N is a normal distribution with mean $\mu = 0$ and standard deviation $\sigma = 0.005$. This way, 95% of the amplitudes of $w(k)$ are within ± 0.01 (2σ).

The results of this section were obtained through simulations using MATLAB/SIMULINK. The evaluate the benefits – in terms of accuracy – provided by the noise processing technique described in Section 3, two BiSI attacks are implemented for comparison:

(I) a BiSI attack using the noise processing technique along with the BSA optimization process, such as described in Section 3;

(II) a BiSI attack using only the BSA optimization process (*i.e.*, without the noise processing stage). In this case, the eavesdropped signals $i(k)$ and $o(k)$ are directly used – without treatment – by the BSA to estimate the parameters of $G_e(z)$. To do so, equations (6) and (7) – used to compute the fitness of BSA individuals – are rewritten as (9) and (10), and the BiSI attack is simply represented by Algorithm 3.

$$G_e(z) = \frac{\mathcal{Z}[\hat{o}(k)]}{\mathcal{Z}[i(k)]} = \frac{\alpha_p z^p + \alpha_{p-1} z^{p-1} + ... + \alpha_1 z^1 + \alpha_0}{z^q + \beta_{q-1} z^{q-1} + ... + \beta_1 z^1 + \beta_0}, \tag{9}$$

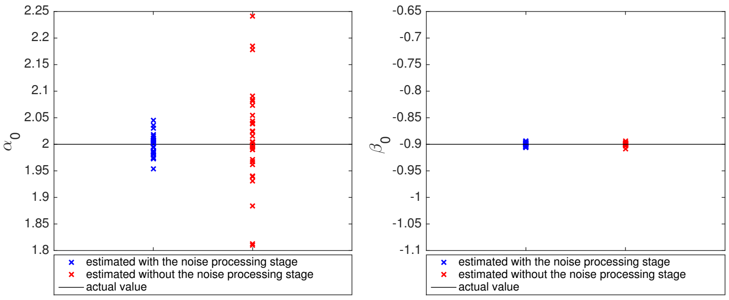
$$f_j = \frac{\sum_{k=1}^{\tau} \left[o(k) - \hat{o}_j(k)\right]^2}{\tau}. \tag{10}$$

Algorithm 3: BiSI attack without the noise processing strategy

```
begin
    Eavesdrop $i(k)$ and $o(k)$ during $\tau$ samples;
    Execute BSA, using $i(k)$ and $o(k)$ to find $G(z)$.
end
```

As previously discussed, the BiSI attack aims to estimate the coefficients of the LTI transfer function of an NCS device. Therefore, in the present simulations, the parameters to be identified – according to (8) – are $\alpha_0 = 2$ and $\beta_0 = 0.9$. The BSA configurations in this paper are the same as those used in [9, 10]: the lower and upper limits of each search space dimension are -10 and 10, respectively; the number of individuals in the BSA population is 100; $\eta = 1$; and the stopping criteria is 600 iterations. Moreover, $T = 0,5Msamples$, $\tau = 100samples$ and $\mathit{\Omega} = 0.01$.

Each of the BiSI attack implementations – (I) and (II) – are evaluated through 31 different simulations. Each simulation uses a different white gaussian noise signal, randomly generated. Figure 4 shows the 31 values of α_0 and β_0 estimated by the two BiSI attack implementations (*i.e.*, with and without the noise processing stage). Additionally, Table 1 shows the statistics of the results presented in Figure 4. From Figure 4 and Table 1, it is possible to verify that the accuracy of the BiSI attack with the noise processing stage is better than the accuracy of the BiSI attack without the proposed technique. Figure 4(b) indicates that the two implementations have similar performance when estimating β_0. In both implementations, all estimated β_0 are close to the actual β_0 and, according to Table 1, the standard deviations are similarly low. On the other hand, Figure 4(a) demonstrates that implementation (I) has better performance than implementation (II) when estimating α_0. With the noise processing stage, the estimated values of α_0 are closer to the actual α_0 – *i.e.*, less spread than without the noise processing stage. The statistics shown in Table 1 ratifies the better performance provided by the noise processing stage when the BiSI attack estimates α_0. In this case, the mean of the estimated values is closer to the actual α_0, with lower standard deviation.



(a) Estimations of α_0 (b) Estimations of β_0

Fig. 4. Estimations of α_0 and β_0 with and without the noise processing stage.

Table 1. Statistics of the BiSI attacks

Coefficient	BiSI attack Implementation	Mean	Standard Deviation
α_0	(I)	1.9997	0.0189
	(II)	2.0119	0.0911
β_0	(I)	-0.8999	0.0034
	(II)	-0.8998	0.0024

Figure 5, obtained from one example of BiSI attack using implementation (I), compares the impulse response function $\mathcal{O}(k)$ of $G(z)$ – computed by the noise processing stage – with the impulse response function $\hat{\mathcal{O}}(k)$ of the estimated model $G_e(z)$. Note that, this figure demonstrates the product of the work done by the noise processing stage: a clear impulse response function, extracted from a white gaussian noise, that is better handled by the bio-inspired identification process performed by the BSA. It is possible to see how close $\hat{\mathcal{O}}(k)$ is from $\mathcal{O}(k)$, which demonstrates the high accuracy of the estimated model $G_e(z)$ when the BSA-based identification uses the signals provided by the proposed noise processing stage.

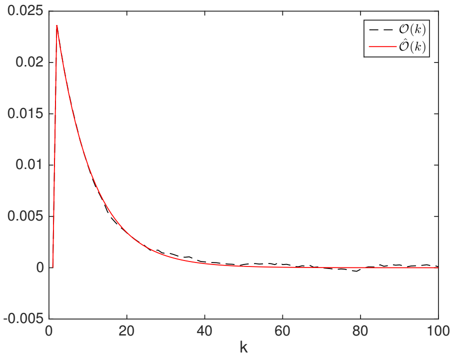


Fig. 5. Evaluation of the performance of the identification process – comparison between $\mathcal{O}(k)$ and $\hat{\mathcal{O}}(k)$.

5 Conclusion

In this work we propose a noise processing technique to improve the accuracy of bio-inspired system identification algorithms. The simulation results indicate that when the proposed technique is performed prior to the BSA-based system identification process, the accuracy of the estimated model increases. Therefore, the present technique represent a useful tool to make BiSI attacks effective in noisy NCSs. The proposed technique overcomes the constraint presented in other implementations of BiSI attacks, where the accuracy of estimated models used to be degraded by noise. The outcomes indicates that, with this approach, noise may not be a problem for a BiSI attack. Instead, noise can represent a meaningful and useful information for an attacker if he/she uses the approach described in this paper.

For future work, we plan to investigate techniques to mitigate BiSI attacks, by hindering the identification process in situations where an attacker has access to the data flowing in the NCS. Moreover, we plan to investigate the use of the proposed algorithm as a defense tool to identify possible model-based attacks in noisy NCSs. In this sense, we believe that this algorithm can be used to provide the NCS with information regarding the model of an eventual attack, in order to allow the autonomous reconfiguration of the control function to compensate the presence of the attack.

Acknowledgements

This work was partially sponsored by the EU-BR SecureCloud project (MCTI/ RNP 3rd Coordinated Call), by the Coordination for the Improvement of Higher Education Personnel (CAPES), grant 99999.008512/2014-0, and by FCT through project LaSIGE (UID/CEC/00408/2013).

References

1. Amin, S., Litrico, X., Sastry, S., Bayen, A.M.: Cyber security of water scada systems part i: analysis and experimentation of stealthy deception attacks. IEEE Transactions on Control Systems Technology **21**(5), 1963–1970 (2013)
2. Civicioglu, P.: Backtracking search optimization algorithm for numerical optimization problems. Applied Mathematics and Computation **219**(15), 8121–8144 (2013)
3. Farooqui, A.A., Zaidi, S.S.H., Memon, A.Y., Qazi, S.: Cyber security backdrop: A scada testbed. In: Computing, Communications and IT Applications Conference (ComComAp), 2014 IEEE. pp. 98–103. IEEE (2014)
4. Ferrari, P., Flammini, A., Rizzi, M., Sisinni, E.: Improving simulation of wireless networked control systems based on wirelesshart. Computer Standards & Interfaces **35**(6), 605–615 (2013)
5. Ji, K., Wei, D.: Resilient control for wireless networked control systems. International Journal of Control, Automation and Systems **9**(2), 285–293 (2011)
6. Langner, R.: Stuxnet: Dissecting a cyberwarfare weapon. Security & Privacy, IEEE **9**(3), 49–51 (2011)
7. Long, M., Wu, C.H., Hung, J.Y.: Denial of service attacks on network-based control systems: impact and mitigation. Industrial Informatics, IEEE Transactions on **1**(2), 85–96 (2005)
8. de Sá, A.O., da Costa Carmo, L.F., Machado, R.C.: A controller design for mitigation of passive system identification attacks in networked control systems. Journal of Internet Services and Applications **9**(1), 2 (2018)

9. de Sá, A.O., Carmo, L.F.d.C., Machado, R.C.: Bio-inspired active system identification: a cyber-physical intelligence attack in networked control systems. Mobile Networks and Applications pp. 1–14 (2017)
10. de Sá, A.O., da Costa Carmo, L.F.R., Machado, R.C.: Covert attacks in cyber-physical control systems. IEEE Transactions on Industrial Informatics **13**(4), 1641–1651 (2017)
11. Smith, R.: A decoupled feedback structure for covertly appropriating networked control systems. In: Proceedings of the 18th IFAC World Congress 2011. vol. 18. IFAC-PapersOnLine (2011)
12. Smith, R.S.: Covert misappropriation of networked control systems: Presenting a feedback structure. Control Systems, IEEE **35**(1), 82–92 (2015)
13. Snoeren, A.C., Partridge, C., Sanchez, L.A., Jones, C.E., Tchakountio, F., Schwartz, B., Kent, S.T., Strayer, W.T.: Single-packet ip traceback. IEEE/ACM Transactions on Networking (ToN) **10**(6), 721–734 (2002)
14. Teixeira, A., Shames, I., Sandberg, H., Johansson, K.H.: A secure control framework for resource-limited adversaries. Automatica **51**, 135–148 (2015)

APPENDIX I

Countermeasure for Identification of Controlled Data Injection Attacks in Networked Control Systems

Alan Oliveira de Sá
Institute of Mathematics/NCE, Federal University of Rio de Janeiro, RJ, Brazil
Email: alan.oliveira.sa@gmail.com

Luiz Fernando Rust da C. Carmo
National Institute of Metrology, Quality and Technology, RJ, Brazil
Email: lfrust@inmetro.gov.br

Raphael C. Santos Machado
National Institute of Metrology, Quality and Technology, RJ, Brazil
Email: rcmachado@inmetro.gov.br

***Abstract*—Networked Control Systems (NCS) are widely used in Industry 4.0 to obtain better management and operational capabilities, as well as to reduce costs. However, despite the benefits provided by NCSs, the integration of communication networks with physical plants can also expose these systems to cyber threats. This work proposes a link monitoring strategy to identify linear time-invariant transfer functions performed by a Man-in-the-Middle during controlled data injection attacks in NCSs. The results demonstrate that the proposed identification scheme provides adequate accuracy when estimating the attack function, and does not interfere in the plant behavior when the system is not under attack.**

***Index Terms*—Security, Networked Control System, Data Injection Attack, Countermeasure, System Identification.**

1. Introduction

The concept of the fourth industrial revolution – the Industry 4.0 – arises with the development and use of cyber-physical systems, which promote the computerization of manufacturing and integrate communication networks to physical processes. In this scenario, Networked Control Systems (NCS) – *i.e.*, controllers and physical plants connected through communication networks – are widely used to obtain better management and operational capabilities, as well as cost reductions [1]. The possible applications for NCSs are broad and can range from non-critical industrial plants controlled by wireless networked control systems (WNCS) [2], to critical infrastructures controlled by wired NCSs, such as nuclear reactors [3] and water canal systems [4]. However, despite the several benefits provided by NCSs, the use of communication networks to integrate controllers and physical plants can also expose these systems to cyber threats [1], [4], [5], [6], [7], [8]. In this context, the literature on NCSs shows a research effort to characterize vulnerabilities and promote security solutions for this kind of system [1], [4], [5], [7], [8], [9].

In [4], [7], it is proposed a covert misappropriation attack, where a malicious agent uses the knowledge about the plant model to inject false data in the NCS. The author assumes that the attacker knows the plant model, but does not describe how the model is obtained. More recent works [1], [5] demonstrate that Service Degradation (SD)-Controlled Data Injection attacks, produced to cause harmful effects on physical plants, can be accurately built based on data gathered by system identification attacks. In [9] the authors discuss countermeasures that can be used to prevent data injection attacks in NCSs. These countermeasures can be systematically thought in a layered defense strategy [9] to avoid access to the control loop and data.

Non authorized access to the NCS control loop can be obtained, for instance, by using network segmentation, demilitarized zones (DMZ), firewall policies and implementing specific network architectures, such as described in [10]. Additionally, non authorized access to data flowing in the NCS can be obtained by using security mechanisms for data confidentiality, integrity and authenticity. Such a solution is presented in [11], where the authors propose a countermeasure that integrates a symmetric-key encryption algorithm, a hash algorithm and a timestamp strategy to form a secure transmission mechanism between the controller side and the plant side. However, despite the security solutions that the literature offers to protect NCSs, it is necessary to consider that an attacker can still overcome security mechanisms for data confidentiality, integrity and authenticity. Indeed, it is possible to use alternative methods, such as social engineering, to obtain the security keys necessary to manipulate the data transmitted in the NCS links. In this case, as shown in [1], [5], an attacker can have the conditions required to implement an SD-Controlled Data Injection attack. Therefore, it is important to develop countermeasures able to detect and identify SD-Controlled Data Injection attacks in NCSs.

In this sense, this work proposes a link monitoring strategy to identify the linear time-invariant (LTI) transfer function performed by a Man-in-the-Middle (MitM) during an SD-Controlled Data Injection attack [1]. From the NCS owner perspective, the knowledge about the attack function may be useful, for instance, to:

- provide information for an autonomous process intended to redesign the NCS control function, in order to mitigate the attack effects in the plant behavior;
- reveal the attacker intentions, for forensic purposes, helping to estimate the possible impacts of the attack on the plant and its services.

The reminder of this work is organized as follows: Section 2 briefly presents the concepts of the SD-Controlled Data Injection attack [1]. Section 3 describes the proposed attack identification strategy – a link monitoring technique –, which uses white gaussian noise to excite the attack function and obtain the information necessary to identify the attack. Section 4 shows simulation results that evaluate the ability of the proposed strategy in identifying an SD-Controlled Data Injection attack. Finally, Section 5 brings our conclusions.

2. SD-Controlled Data Injection attack

For the sake of completeness, this section briefly describes the SD-Controlled Data Injection attack characterized in [1]. Its purpose is to reduce the mean time between failure (MTBF) of the plant and/or reduce the efficiency of the physical process that the plant performs, by inserting false data in the NCS communication links.

In the SD-Controlled Data Injection attack, to cause a harmful behavior on the plant, the attacker interfere in the NCS's links by injecting false data into the system in a controlled way. To do so, the attacker act as a MitM that executes an LTI attack function $M(z)$, presented in Figure 1, wherein $Y''(z) = M(z)Y'(z)$, $Y'(z) = \mathcal{Z}[y'(k)]$ and $Y''(z) = \mathcal{Z}[y''(k)]$. The function $M(z)$ is designed based on the models of the plant and the controller, both obtained through a System Identification attack [1], [5].

3. Attack Identification Strategy

This section describes a link monitoring strategy to identify the LTI attack functions used by a MitM during the SD-Controlled Data Injection attack described in Section 2. Consider, for instance, the SD-Controlled Data Injection attack shown in Figure 1, where the attacker only has access to the data flowing in the feedback stream.

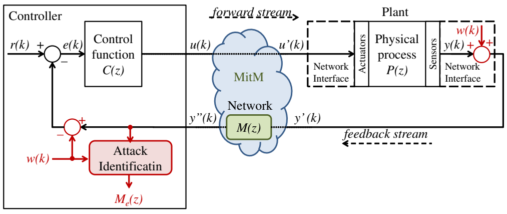


Figure 1: Attack identification strategy.

To identify the attack function, $M(z)$ has to be excited by an input signal in order to produce meaningful information for the identification process. If the system is in steady operating conditions, for instance, the information content of measured signals is often insufficient for identification purposes [12]. Considering this, one possible strategy to identify an attack function is to use typical variations in the NCS signals – such as a variation caused by a change in the setpoint $r(k)$ – to estimate $M(z)$. However, depending on the system, this variations may not occur often, which can make the identification of $M(z)$ time consuming. Furthermore, causing arbitrary variations in such signals in order to identify $M(z)$ may not be convenient as it may affect the behavior of the plant.

The architecture shown in Figure 1 is proposed as a solution that can be used to excite $M(z)$ at any time, without affecting the plant behavior when the system is working in normal conditions – *i.e.*, without attack. To do so, as shown in Figure 1, a white gaussian noise $w(k)$ is injected (added) in the signal to be transmitted to through the monitored link. To avoid interfering in the controlled plant when the system in not under attack, the same noise signal $w(k)$ is subtracted from the monitored NCS signal at the other end of the link. In Figure 1, where the feedback link is the one being monitored, $w(k)$ is injected at the sensor's network interface, and subtracted at the controller input. In this system, the NCS output $Y(z) = \mathcal{Z}[y(k)]$ is defined as (1):

$$Y(z) = \frac{C(z)P(z)}{1 + C(z)P(z)M(z)} \left[R(z) + W(z)\left(1 - M(z)\right)\right], \tag{1}$$

wherein $R(z) = \mathcal{Z}[r(k)]$ and $W(z) = \mathcal{Z}[w(k)]$. Note that, if $w(k)$ is exactly the same signal at both ends of the monitored link and the system is not under attack (*i.e.*, $M(z) = 1$), then the injection of $w(k)$ is cancelled and does not influence in $y(k)$. In this case, based on (1), the plant output $Y(z)$ is defined as (2):

$$Y(z) = \frac{C(z)P(z)}{1 + C(z)P(z)}R(z). \tag{2}$$

The white gaussian noise $w(k)$ is chosen to excite the attack function due to its unpredictability, which makes it harder for an attacker to estimate the noise that will be added to the link at any given moment. The white gaussian noise $w(k)$ is obtained from a normal distribution, such that $w(k) \sim N(\mu, \sigma)$, wherein $\mu = 0$ is the mean and σ is the standard deviation. To have the same noise signal $w(k)$ at both ends of the monitored link, it is considered that these two sources of noise are synchronized and both signals are produced based on the same seed. Moreover, to avoid an attacker to predict the noise values, the seed is exchanged among both devices – *i.e.*, the transmitter and receiver – using a secure key exchange method, such as the Diffie-Hellman algorithm [13].

Now, if the system is under attack (*i.e.*, $M(z) \neq 1$), then, according to (1), the noise is not cancelled. In this case, the the signal observed at the controller input $y''(k)$ is given by (3):

$$y''(k) = \underbrace{w(k) * \mathcal{Z}^{-1}\left[M(z)\left(\frac{1 + C(z)P(z)}{1 + C(z)P(z)M(z)}\right)\right]}_{y_1''(k)} + \underbrace{r(k) * \mathcal{Z}^{-1}\left[\frac{C(z)P(z)M(z)}{1 + C(z)P(z)M(z)}\right]}_{y_2''(k)}. \tag{3}$$

In the present countermeasure, the identification of $M(z)$ is performed by observing the variations produced by $w(k)$ in $y''(k)$ when $M(z) \neq 1$. Note, in Figure 1, that both $w(k)$ and $y''(k)$ are provided to the Attack Identification process. The effect of $w(k)$ in $y''(k)$ is specifically indicated in (3) as $y_1''(k)$. To have the identification relying on $y_1''(k)$, and independent from variations in $y_2''(k)$, it is executed when the system is in steady state with regard to $r(k)$. In other words, the identification occurs when $y_2''(k)$ – driven by the setpoint $r(k)$ – converges to a constant value ρ. In this case, considering the time window defined by $k_s < k < k_u$

in which $y_2''(k)$ is in its steady state, (3) can be rewritten as (4) without initial conditons:

$$y''(k) = \underbrace{w(k) * \mathcal{Z}^{-1}\left[M(z)\left(\frac{1+C(z)P(z)}{1+C(z)P(z)M(z)}\right)\right]}_{y_1''(k)} + \underbrace{\rho}_{y_2''(k)}, \qquad \forall k_s < k < k_u, \tag{4}$$

wherein ρ can be estimated by computing the average $\bar{y}''$ of $y''(k)$ during a certain amount of samples $\tau \leq (k_u - k_s)$ starting at k_s, as indicated in (5):

$$\begin{aligned}\bar{y}'' &= \sum_{k_s}^{k_s+\tau} \frac{y''(k)}{\tau} \\ &= \underbrace{\sum_{k_s}^{k_s+\tau} \frac{w(k) * \mathcal{Z}^{-1}\left[M(z)\left(\frac{1+C(z)P(z)}{1+C(z)P(z)M(z)}\right)\right]}{\tau}}_{\bar{y}_1''(k)} + \underbrace{\sum_{k_s}^{k_s+\tau} \frac{\rho}{\tau}}_{\bar{y}_2''(k)},\end{aligned} \tag{5}$$

Considering that $w(k) \sim N(\mu, \sigma)$, wherein $\mu = 0$, as previously stated, then $\bar{y}_1''(k) \rightarrow 0$ when $\tau \rightarrow \infty$. In this case, for a sufficiently large τ, (5) can be simplified to (6):

$$\bar{y}'' \approx \rho, \tag{6}$$

Thus, by applying (6) in (4), we may define (7):

$$y_1''(k) \approx y''(k) - \bar{y}'', \qquad \forall k_s < k < k_u, \tag{7}$$

wherein $y_1''(k)$ – obtained through measurements of $y''(k)$ – is the output of the model defined by (8) when the noise $w(k)$ is applied to its input:

$$y_1''(k) = w(k) * \mathcal{Z}^{-1}\left[M(z)\left(\frac{1+C(z)P(z)}{1+C(z)P(z)M(z)}\right)\right]. \tag{8}$$

Based on (8), if $C(z)$ and $P(z)$ are known, the Attack Identification process can estimate $M(z)$ by applying $w(k)$ in an estimated system, defined by (9):

$$\hat{y}_1''(k) = w(k) * \mathcal{Z}^{-1}\left[M_e(z)\left(\frac{1+C(z)P(z)}{1+C(z)P(z)M_e(z)}\right)\right], \tag{9}$$

wherein $M_e(z)$ is the estimation of $M(z)$ and $\hat{y}_1''(k)$ is the output of the estimated system in face of $M_e(z)$. By comparing $\hat{y}_1''(k)$ with $y_1''(k)$, the Attack Identification process is able to evaluate whether $M_e(z)$ is equal/approximately $M(z)$. Note that $M_e(z)$ is a generic LTI attack function represented by (10):

$$M_e(z) = \frac{\alpha_n z^n + \alpha_{n-1} z^{n-1} + ... + \alpha_1 z^1 + \alpha_0}{z^m + \beta_{m-1} z^{m-1} + ... + \beta_1 z^1 + \beta_0}, \tag{10}$$

wherein n and m are the order of the numerator and denominator, respectively, while $[\alpha_n, \alpha_{n-1}, ...\alpha_1, \alpha_0]$ and $[\beta_{m-1}, \beta_{m-2}, ...\beta_1, \beta_0]$ are the coefficients of the numerator and denominator, respectively, that are intended to be found by Attack Identification algorithm. Therefore, to find $M(z)$, the coefficients of $M_e(z)$ are adjusted until the estimated output $\hat{y}_1''(k)$ converges to $y_1''(k)$ – obtained from measurements of $y''(k)$ in the real NCS.

In this work, the Backtracking Search Optimization algorithm (BSA) [14], is used to iteratively adjust the coefficients of $M_e(z)$, by minimizing a specific fitness function until $M_e(z)$ converges to the actual $M(z)$. To compute the fitness of the BSA individuals, the noise $w(k)$ – recorded while $y''(k)$ was being captured – is applied on the estimated system defined by (9) and (10), where the coefficients of $M_e(z)$ are the coordinates $x_j = [\alpha_{n,j}, \alpha_{n-1,j}, ...\alpha_{1,j}, \alpha_{0,j}, \beta_{m-1,j}, \beta_{m-2,j}, ...\beta_{1,j}, \beta_{0,j}]$ of an individual j of the BSA. Let $\hat{y}_{1j}''(k)$ be the output of the estimated model (9) (10) in face of $w(k)$, when the coefficients of $M_e(z)$ are x_j. Then, the fitness f_j of each individual j is obtained comparing $\hat{y}_{1j}''(k)$ with $y_1''(k)$, according to (11):

$$f_j = \frac{\sum\limits_{k=0}^{N} (y_1''(k) - \hat{y}_{1j}''(k))^2}{N}, \tag{11}$$

wherein N is the number of samples that exist during a monitoring period T of $y_1''(k)$. Note that, $\min f_j$ occurs when $[\alpha_{n,j}, \alpha_{n-1,j}, ...\alpha_{1,j}, \alpha_{0,j}, \beta_{m-1,j}, \beta_{m-2,j}, ... \beta_{1,j}, \beta_{0,j}] \rightarrow [\alpha_n, \alpha_{n-1}, ...\alpha_1, \alpha_0, \beta_{m-1}, \beta_{m-2}, ...\beta_1, \beta_0]$, *i.e.* when the estimated $M_e(z)$ converges to $M(z)$.

4. Results

This section evaluates the performance of the countermeasure proposed in Section 3 when identifying an SD-Controlled Data Injection attack (characterized in Section 2). The results of the attack identification are obtained through simulations using MATLAB/SIMULINK. The attacked system consists of a DC motor controlled by a proportional-integral (PI) controller. The plant transfer function $P(z)$ and the control function $C(z)$ are represented by (12):

$$P(z) = \frac{0.3379z + 0.2793}{z^2 - 1.5462z + 0.5646} \quad C(z) = \frac{0.1701z - 0.1673}{z - 1} \tag{12}$$

The sample rate of the system is 50 samples/s and the set point $r(k)$ is a unitary step function.

As discussed in [1], one way to degrade a physical service of a plant is by causing overshoots during its transient response, which, indeed, can cause stress and possibly damage in a variety of physical systems [15]. Additionally, once the overshoot occur in a short period of time, they are difficult to be noticed by a human observer. Thus, in this work, an attack function $M(z)$ is designed to degrade the plant service by causing 50% of overshoot in the motor speed. The attack function implemented in the present SD-Controlled Data Injection attack is represented by (13):

$$M(z) = \frac{\alpha_0}{z + \beta_0}, \tag{13}$$

wherein $\alpha_0 = 0.25$ and $\beta_0 = -0.75$. In the present simulations, the parameters of the noise $w(k) \sim N(\mu, \sigma)$ are $\mu = 0$ and $\sigma = 0.005$.

Figure 2 shows the system output – *i.e.* the motor speed – with and without the attack. Note that, when the attack is executed, the motor speed has an overshoot of 50%, and a portion of noise is present in the system output. However, in a normal condition – *i.e.*, without attack – the noise is cancelled and does not appear in the plant output, as expected.

As previously discussed, the present attack identification strategy aims to estimate the coefficients of $M(z)$, which according to (13) are α_0 and β_0. The signals $w(k)$ and $y''(k)$, used by the identification algorithm, are recorded during is $T = 2s$ (100 samples), starting when the system achieves it steady state regarding to $r(k)$. The BSA configurations used in the simulations of this work are the same as those used in [1]: the lower and upper limits of each search space dimension are -10 and 10, respectively; the number of individuals in the BSA population is 100; and $\eta = 1$ (in the BSA, η is used to define the amplitude of the displacement of the individuals).

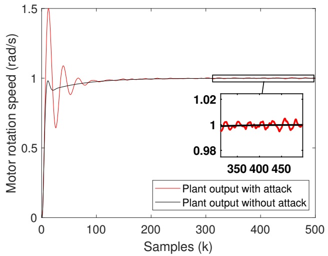


Figure 2: Motor speed with and without attack.

Figure 3 shows the values of α_0 and β_0 estimated in 100 attack simulations. Additionally, Table 1 shows the statistics of these results. In both Figure 3 and Table 1, it is possible to verify the accuracy achieved by the attack identification strategy proposed in this work. Note that, in Figure 3, all estimated values of α_0 and β_0 converge to their actual values. Moreover, in Table 1, the mean of the estimated coefficients are close to their real values with small standard deviation, which ratifies the accuracy verified in Figure 3 and indicates the effectiveness of the proposed countermeasure when identifying SD-Controlled Data Injection attacks.

TABLE 1: Statistics of the attack identification.

Coefficient	Mean	Standard Deviation
α_0	0.2506	0.0147
β_0	-0.7485	0.0172

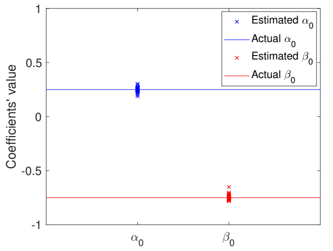


Figure 3: Estimations of α_0 and β_0.

5. Conclusion

This paper proposes an attack identification mechanism to estimate LTI transfer functions that are implemented during SD-Controlled Data Injection attacks. The identification is performed by injecting white gaussian noise in the monitored NCS link. The results demonstrate that the proposed identification scheme provides adequate accuracy when estimating the attack function. Also, the results testify that the identification scheme does not interfere in the plant behavior when the system is not under attack. The problem formulation does not take into account residual initial conditions that may exist when signals are collected in the NCS. We consider that the use of the BSA optimization – which is able to find near optimum solutions – provides satisfactory accuracy in the attack identification, even not taking into account such initial conditions. In a future work, we plan to include the NCS initial conditions in the problem formulation, in order to evaluate how does it affect the attack identification process in terms of accuracy and computational cost.

References

[1] A. O. de Sa, L. F. R. da Costa Carmo, and R. C. S. Machado, "Covert attacks in cyber-physical control systems," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1641–1651, Aug 2017.

[2] P. Ferrari, A. Flammini, M. Rizzi, and E. Sisinni, "Improving simulation of wireless networked control systems based on wirelesshart," *Computer Standards & Interfaces*, vol. 35, no. 6, pp. 605–615, 2013.

[3] M. Das, R. Ghosh, B. Goswami, A. Gupta, A. Tiwari, R. Balasubrmanian, and A. Chandra, "Network control system applied to a large pressurized heavy water reactor," *IEEE Transactions on Nuclear Science*, vol. 53, no. 5, pp. 2948–2956, 2006.

[4] R. S. Smith, "Covert misappropriation of networked control systems: Presenting a feedback structure," *Control Systems, IEEE*, vol. 35, no. 1, pp. 82–92, 2015.

[5] A. O. de Sa, L. F. R. da Costa Carmo, and R. C. S. Machado, "Bio-inspired active system identification: a cyber-physical intelligence attack in networked control systems," *Mobile Networks and Applications*, pp. 1–14, 2017.

[6] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *Security & Privacy, IEEE*, vol. 9, no. 3, pp. 49–51, 2011.

[7] R. Smith, "A decoupled feedback structure for covertly appropriating networked control systems," in *Proceedings of the 18th IFAC World Congress 2011*, vol. 18, no. 1. IFAC-PapersOnLine, 2011.

[8] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "A secure control framework for resource-limited adversaries," *Automatica*, vol. 51, pp. 135–148, 2015.

[9] A. O. de Sa, L. F. R. da Costa Carmo, and R. C. S. Machado, "A controller design for mitigation of passive system identification attacks in networked control systems," *Journal of Internet Services and Applications*, vol. 9, no. 1, pp. 1–19, Feb 2018.

[10] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn, "Nist special publication 800-82, revision 2: Guide to industrial control systems (ics) security," *Gaithersburg, MD, USA: National Institute of Standards and Technology*, 2015.

[11] Z.-H. Pang and G.-P. Liu, "Design and implementation of secure networked predictive control systems under deception attacks," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 5, pp. 1334–1342, 2012.

[12] H. J. Tulleken, "Generalized binary noise test-signal concept for improved identification-experiment design," *Automatica*, vol. 26, no. 1, pp. 37–49, 1990.

[13] W. Stallings, *Cryptography and network security: principles and practices*. Pearson Education India, 2006.

[14] R. Kennedy, J. e Eberhart, "Particle swarm optimization," in *Proceedings of 1995 IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[15] T. Tran, Q. P. Ha, and H. T. Nguyen, "Robust non-overshoot time responses using cascade sliding mode-pid control," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 2007.