sensors

MDPI

Article

Identification of Data Injection Attacks in Networked Control Systems Using Noise Impulse Integration †

Alan Oliveira de Sá [1,2,*,‡], António Casimiro [3,‡], Raphael C. S. Machado [4,5,‡] and Luiz F. R. da C. Carmo [2,3,‡]

1 Admiral Wandenkolk Instruction Center, Brazilian Navy, Rio de Janeiro RJ 20180-003, Brazil
2 Institute of Mathematics/NCE, Federal University of Rio de Janeiro, Rio de Janeiro RJ 21945-970, Brazil; lfrust@inmetro.gov.br
3 Department of Informatics, Faculty of Sciences of the University of Lisboa, 1749-016 Lisboa, Portugal; casim@ciencias.ulisboa.pt
4 National Institute of Metrology, Quality and Technology, Xerém RJ 25250-020, Brazil; rcmachado@inmetro.gov.br
5 Institute of Computing, Fluminense Federal University, Niterói RJ 24210-310, Brazil
* Correspondence: alan.oliveira.sa@gmail.com
† This paper is an extention version of conference paper : de Sá A.O.; Carmo, L.F.R.d.C.; Machado, R.C.S. Countermeasure for Identification of Controlled Data Injection Attacks in Networked Control Systems. In Proceedings of 2019 IEEE International Workshop on Metrology for Industry 4.0 & IoT, Naples, Italy, 4–6 June 2019.
‡ These authors contributed equally to this work.

Received: 12 November 2019; Accepted: 18 January 2020 ; Published: 31 January 2020

Abstract: The benefits of using Networked Control Systems (NCS) in the growing Industry 4.0 are numerous, including better management and operational capabilities, as well as costs reduction. However, despite these benefits, the use of NCSs can also expose physical plants to new threats originated in the cyber domain—such as data injection attacks in NCS links through which sensors and controllers transmit signals. In this sense, this work proposes a link monitoring strategy to identify linear time-invariant (LTI) functions executed during controlled data injection attacks by a Man-in-the-Middle hosted in an NCS link. The countermeasure is based on a bioinspired metaheuristic, called Backtracking Search Optimization Algorithm (BSA), and uses white Gaussian noise to excite the attack function. To increase the accuracy of this countermeasure, it is proposed the Noise Impulse Integration (NII) technique, which is developed using the radar pulse integration technique as inspiration. The results demonstrate that the proposed countermeasure is able to accurately identify LTI attack functions, here executed to impair measurements transmitted by the plant sensor, without interfering with the NCS behavior when the system is in its normal operation. Moreover, the results indicate that the NII technique can increase the accuracy of the attack identification.

Keywords: security; industrial control system; networked control system; data injection attack; countermeasure; system identification

1. Introduction

The concept of the fourth industrial revolution—Industry 4.0 [1,2]—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 sensors/actuators of physical plants connected through communication networks [3–7]—are widely used to obtain better management and operational capabilities, as well as

cost reductions [8]. In an NCS, as shown in Figure 1, a controller—i.e., a computer system—executes a control function $C(z)$ to properly drive the behavior of a physical plant, herein described by a discrete-time transfer function $P(z)$. The control signal produced by the controller is transmitted to the plant actuators through a forward stream. The signals measured by the plant sensors, in turn, are sent to the controller through a feedback stream.

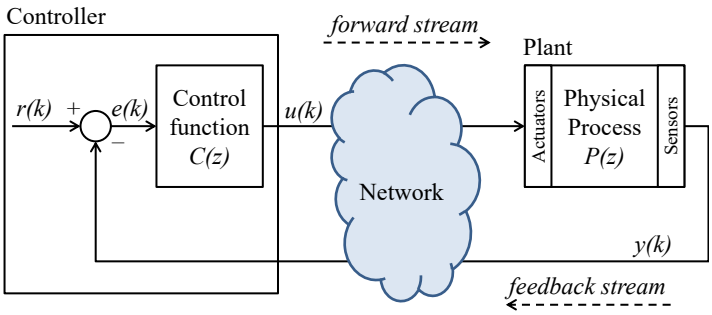


Figure 1. Networked Control Systems (NCS) [8].

The possible applications for NCSs are broad and can range from non-critical industrial plants controlled by wireless networked control systems (WNCS) [9], to critical infrastructures controlled by wired NCSs, such as nuclear reactors [6,10,11] and water canal systems [12]. 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 [8,12–16]. Indeed, the literature [7] 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 [14], whose targets were uranium enrichment centrifuges in Iran [17]. To achieve its aim, Stuxnet installed a modified control algorithm into the controller (a programmable logic controller—PLC) in order to cause subtle and harmful behaviors to the centrifuges, reducing their efficiency and causing damage [14,17,18].

Please 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 [14]. Another possible way for an attacker to negatively affect an NCS is by interfering on its communication process between controllers, sensors and actuators. 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 [9,19,20], there are protocols in industry that still lack security mechanisms [21]—such as the Profinet, MODBUS/TCP, and Ethernet/IP. The main issue of these industrial protocols is the lack of encryption and authentication [21] between devices (e.g., controllers, actuators, and sensors) used in automation and control systems. A vast collection of scientific literature about cybersecurity in Industrial Control Systems (ICS) is available, reporting security breaches in all major Real-Time Ethernet (RTE) protocols used in industry [21–28]. Therefore, considering the feasibility of occurring cyber-attacks against physical systems, as demonstrated by the real cases already reported in the literature [7,14,17], studies have been conducted aiming to characterize vulnerabilities and promote security solutions for NCSs [8,12,13,15,16,29].

In [12,15], 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 [8,13] demonstrate that Service Degradation (SD)-Controlled Data Injection attacks can be accurately built based on the NCS models previously learned through system identification attacks [8,13]. The harmful

effects that SD-Controlled Data Injection attacks can produce on physical plants motivate the research on mechanisms able to prevent them, as well as to detect/identify them when they occur.

It is possible to verify in the literature [7,8,12,13,15,16,21,22,26,29–36] that in cyber-physical systems (which includes NCSs), a relevant portion of the attack surface often lies in the communication process between sensors/actuators and controllers. For this reason, in the ICS cybersecurity landscape, significant attention has been given to the study of cyber-attacks to sensor/actuator systems [7]. Indeed, the high accuracy desired for sensors, for instance, may be useless if the integrity of sensor data is compromised by some kind of malicious manipulation in its communication process. Not by chance, still taking the scope of sensors as an example, data integrity is arising as a property as important as other typical sensor properties—e.g., accuracy, sensitivity, linearity, resolution, repeatability, etc. [32,34–40].

Aiming to improve the cybersecurity of NCSs, the authors of [29] discuss countermeasures that can be used to mitigate data injection attacks executed within the communication between sensors/actuators and controllers. These countermeasures can be systematically thought in a layered defense strategy [29] 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 [31]. Additionally, non-authorized access to data transmitted by controllers and sensors can be obtained by using security mechanisms for data confidentiality, integrity and authenticity. Such a solution is presented in [32], 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 sensors/actuators located in the plant side. However, it is noteworthy that even when NCS uses secure communication protocols and network architectures, existing security mechanisms can still be overcome. The security of the communication between sensors, controllers and actuators may be compromised, for instance, if an attacker succeed in obtaining security keys or passwords (used for encryption and authentication) through social engineering attacks [41]. In this case, as shown in [8,13], 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 linear time-invariant (LTI) transfer functions performed by a Man-in-the-Middle (MitM) during an SD-Controlled Data Injection attack [8]. The proposed countermeasure uses white gaussian noise to excite possible attack functions in the NCS, to obtain the information necessary to identify the attack. Moreover, 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 [42]. 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, 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.

Previous works [37–39] report the use of Independent and Identically Distributed (IID) noise sequence as watermark to detect data injection attacks (integrity attacks) in NCSs. More specifically, the solutions proposed in [37–39] provide a physical authentication scheme to detect replay attacks in sensors' measurements when the NCS is in steady state. In [38,39], the core idea of the detection scheme is to add an IID noise to the control signal applied to the plant and, thus, obtain a physical watermark within the plant output signal—transmitted by sensors—in a system equipped with a χ^2 failure detector. In [37], to detect counterfeit sensor signals, the authors investigate the problem of designing the optimal watermark signal in the class of stationary Gaussian processes. Their results

generalize the solution proposed in [38,39] where only IID Gaussian processes are considered in the design of watermarked control inputs. Also, the authors propose a watermark design method that bounds the control performance loss incurred by the watermark signal—note that although the cost to control performance is bounded, it is not completely eliminated. As mentioned by the authors, this drawback occurs in all solutions presented in [37–39]. The presence of the extra watermark signal in the control signal causes the control performance to not be optimal—i.e., to allow the attack detection, the control performance is sacrificed. In [40], the authors propose a multiplicative watermarking scheme to detect and isolate replay attacks on sensors measurements without interfering in the control performance. Unlike [37–39], to avoid detrimental effects on the closed loop performance, each sensor output is separately watermarked while an equalization filter is incorporated at the controller's side to reconstruct the original plant outputs.

In the present work, differently from [37–39], the proposed solution is designed not to sacrifice the system performance when it is in normal operation. Here, the white gaussian noise added in the transmitting device (e.g., a sensor) is cancelled in the receiving device (e.g., a controller), in a strategy analogous to the multiplicative watermarking scheme used in [40]. Moreover, while the watermarking schemes proposed in [37–40] aim to detect data injection attacks (specifically, replay attacks) in sensors measurements, they do not intend to identify possible LTI attack functions within the communication between sensors/actuators and controllers. In the present work, differently from [37–40], the proposed solution is intended to detect and identify (i.e., estimate the parameters of) LTI attack functions executed during data injection attacks in NCSs—precisely the class of SD-Controlled Data Injection attacks discussed in [8,13]. The identification of SD-Controlled Data Loss attacks using switching LTI attack functions is not considered in this paper.

It is worth mentioning that the proposed countermeasure is not intended to prevent the implementation of an SD-Controlled Data Injection attack, but to detect and identify it once it occurs, in order to obtain knowledge about the attack function. As in other works addressing the detection of data integrity attacks in NCS [37–39], the countermeasure herein proposed is designed to identify the attack when the plant is operating in steady-state condition – which is still a relevant system condition to be considered in cybersecurity of NCSs [37–39]. Please note that SD attacks, by definition [8], are not intended to cause immediate system failure. They are intended to degrade the efficiency of the physical process or to reduce the mean time between failure of the plant, remaining active in the system for mid/long term. Thus, the knowledge about the attack function obtained during steady operating conditions is useful to build reactive countermeasures that make the attack cease (or mitigate it) once it has started—even if the beginning of the attack was during a transient response and its identification occurs during the subsequent steady condition.

The reminder of this work is organized as follows: Section 2 briefly presents the concepts of the SD-Controlled Data Injection attack [8]. Section 3 describes the proposed countermeasure—a link monitoring mechanism—including the NII technique herein introduced to increase the accuracy of the attack identification. Section 4 shows simulation results that evaluate the performance of the proposed countermeasure when identifying an SD-Controlled Data Injection attack in the communication between a sensor and a controller. It also evaluates the ability of the NII technique in increasing the accuracy of the identification process. Finally, Section 5 brings the conclusions of this work.

2. SD-Controlled Data Injection Attack

For the sake of completeness, this section briefly describes the SD-Controlled Data Injection attack characterized in [8]. The attack 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 (e.g., an overshoot or a steady-state error), 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

$M(z)$ between the sensor and the controller, as presented in Figure 2, wherein $Y'(z) = M(z)Y(z)$, $Y'(z) = \mathcal{Z}[y'(k)]$, $Y(z) = \mathcal{Z}[y(k)]$, and $\mathcal{Z}$ represents de Z-transform operation. The function $M(z)$ is designed based on the models of the plant and the controller, both obtained through a System Identification attack [8,13]. Therefore, according to [8], the SD-Controlled Data Injection attack is implemented in two subsequent stages:

STAGE-I: The system identification stage [8,13] is executed to provide the attacker an accurate knowledge about the models of the targeted system, i.e., the plant's transfer function $P(z)$ and the controller's control function $C(z)$. This knowledge is obtained either through a Passive System Identification process [8] or through an Active System Identification process [13].

STAGE-II: The Data Injection stage is performed. The attacker, as an MitM, injects false data in the NCS control loop. To accurately change the plant physical behavior, the injected false data is computed according to $M(z)$ which, in turn, is designed based on the knowledge obtained by the attacker during STAGE-I.

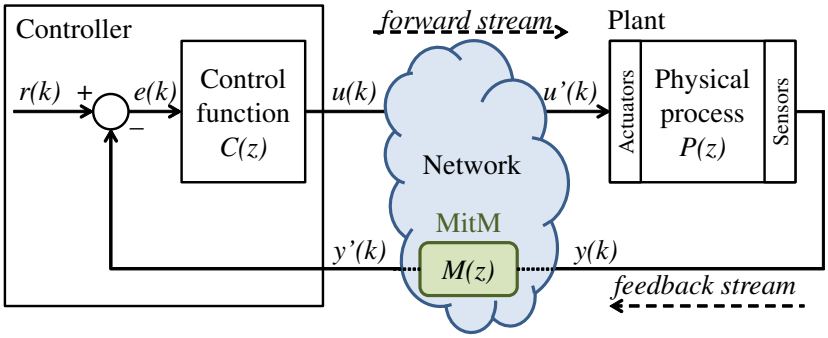


Figure 2. SD-controlled data injection attack.

3. Identification of Controlled Data Injection Attacks

This section proposes a countermeasure—a link monitoring strategy—to identify the LTI transfer function performed by a MitM during an SD-Controlled Data Injection attack (described in Section 2). Section 3.1 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. The countermeasure is designed to do not affect the plant behavior in normal operating conditions (i.e., without attack). To estimate parameters of the LTI attack function, the identification process uses a bioinspired metaheuristic called Backtracking Search Optimization Algorithm (BSA) [43]. Additionally, 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 3.2.

3.1. 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 defined in Section 2. Consider, for instance, the SD-Controlled Data Injection attack shown in Figure 3, where the attacker only has access to the measurements transmitted by the sensor to the controller through the feedback stream.

To identify the attack function, $M(z)$ must be excited by an input signal 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 [44]. 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, these variations may not occur often, which can make the identification of

$M(z)$ time consuming. Furthermore, causing arbitrary variations in such signals to identify $M(z)$ may not be convenient as it may affect the behavior of the plant.

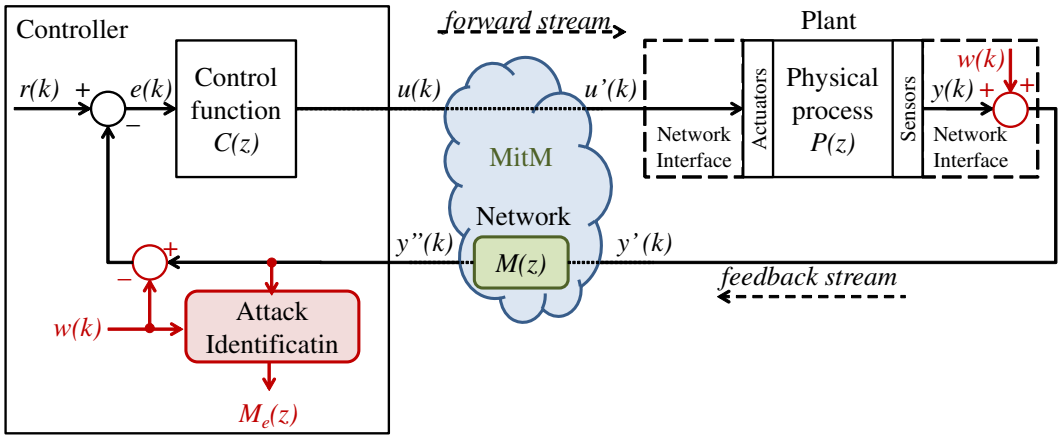


Figure 3. Identification of an SD-Controlled data injection attack [45].

The architecture shown in Figure 3 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 3, 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 3, 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 $W(z) = \mathcal{Z}[w(k)]$. Please 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 $y(k)$. 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 [46].

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 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 3, 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 conditions:

$$y''(k) = \underbrace{w(k) * \mathcal{Z}^{-1}\left[M(z)\left(\frac{1+C(z)P(z)}{1+C(z)P(z)M(z)}\right)\right]}_{y''_1(k)} + \underbrace{\rho}_{y''_2(k)}, \qquad \forall k_s < k < k_u, \tag{4}$$

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):

$$\bar{y}'' = \sum_{k_s}^{k_s+\tau} \frac{y''(k)}{\tau} = \underbrace{\sum_{k_s}^{k_s+\tau} \frac{w(k) * \mathcal{Z}^{-1}\left[M(z)\left(\frac{1+C(z)P(z)}{1+C(z)P(z)M(z)}\right)\right]}{\tau}}_{\bar{y}''_1(k)} + \underbrace{\sum_{k_s}^{k_s+\tau} \frac{\rho}{\tau}}_{\bar{y}''_2(k)}, \tag{5}$$

Considering that $w(k) \sim N(\mu, \sigma)$, wherein $\mu = 0$ as previously stated, then $\bar{y}''_1(k) \to 0$ when $\tau \to \infty$. 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 can evaluate whether $M_e(z)$ is equal/approximately $M(z)$. Please 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) [43], 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 by comparing $\hat{y}''_{1j}(k)$ with $y''_1(k)$, according to (11):

$$f_j = \frac{\sum_{k=0}^{N} (y''_1(k) - \hat{y}''_{1j}(k))^2}{N}, \tag{11}$$

wherein N is the number of samples that exist during a monitoring period T of $y''_1(k)$. Please 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 3.2), is summarized in Algorithm 1.

Algorithm 1: Attack Identification without the NII technique.

```
begin
    if y''(k) is in steady state regarding r(k) then
        Record w(k) and y''(k) during T seconds;
        Compute ȳ'' according to (5);
        Compute y''_1(k) according to (7);
        Execute BSA, using w(k) and y''_1(k) to find M_e(z) based on (9), (10) and (11).
    end if
end
```

3.2. Integrating Impulses of Noise

This section presents the Noise Impulse Integration (NII) technique, which is added to the attack identification process described in Section 3.1 to improve its accuracy. This technique is inspired by the Pulse Integration process [42], used in pulse radar systems to improve the probability of detection and reduce the probability of false alarms in 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. Therefore, first, Section 3.2.1 provides an explanation on the radar pulse integration process. Then, Section 3.2.2 introduces the NII technique.

3.2.1. Radar Pulse Integration

In a pulse radar system, the radar transmits electromagnetic pulses to the environment 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 only one pulse during the detection process. Instead, as depicted in Figure 4, the radar transmits a series of pulses, one at each pulse repetition interval T_R. Also, as shown in Figure 4, between two consecutive transmissions there is a silence period T_L in which the radar remains listening the echoes 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.

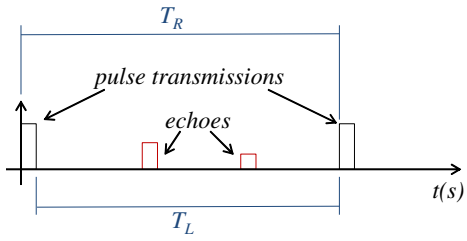


Figure 4. Pulse transmissions.

Please 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 5. Naturally, for each pulse p transmitted from a given 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 6, the received signal also contains uncorrelated signal fluctuations (noise), whose amplitude follows a gaussian distribution with zero mean [47,48].

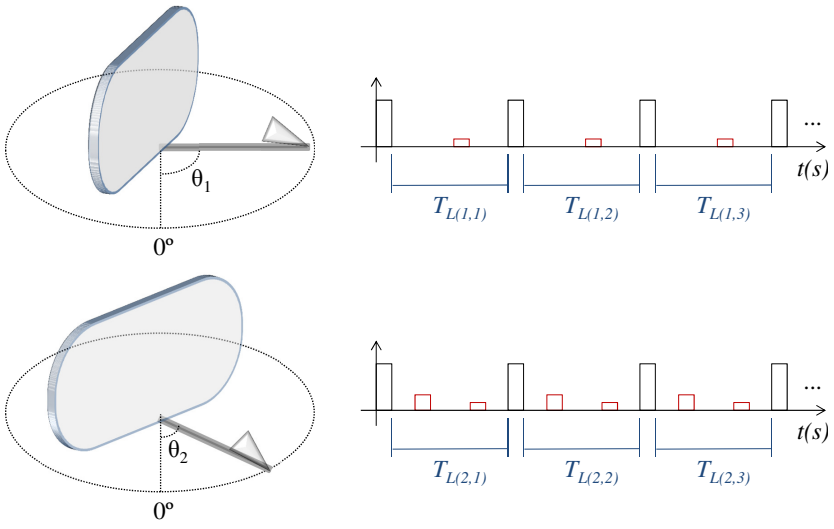


Figure 5. Radar scan process, in which a sequence of pulses is transmitted for each antenna pointing angle.

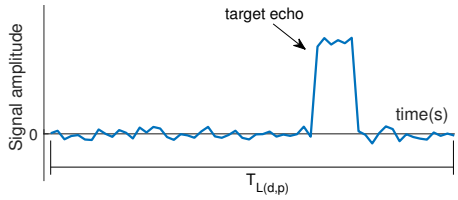


Figure 6. Noisy signal typically received during a given listening period $T_{L(d,p)}$.

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 (12):

$$I(t) = \frac{\sum_{p=1}^{h} S_{(d,p)}(t)}{h}, \tag{12}$$

wherein $I(t)$ is the integrated signal and h is the number of signals buffered in a sequence of listening periods. A representation of this computation is shown in Figure 7, where the signals received in a sequence of four listening periods (i.e., $h = 4$) are buffered and integrated according to (12). Please 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.

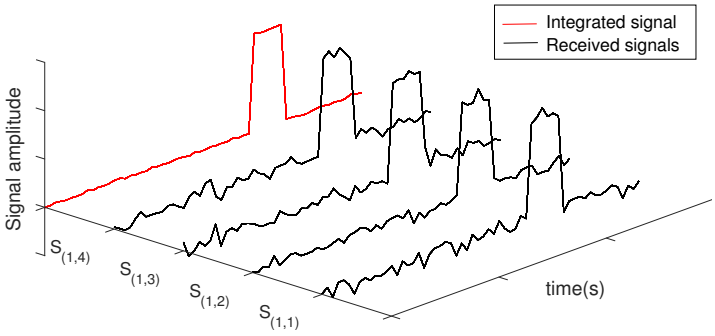


Figure 7. Radar pulse integration.

3.2.2. Noise Impulse Integration Technique

The NII technique described in this section works similarly to the RPI process described in Section 3.2.1. 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 differences 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 3.2.1, 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 7, 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 other 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) as shown in (13):

$$w(k) = \sum_{i=-\infty}^{\infty} \omega(i)\delta(k-i), \tag{13}$$

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 (14):

$$\begin{aligned} y_i(k) &= \omega(i)\delta(k-i) * h(k) \\ &= \omega(i)h(k-i). \end{aligned} \tag{14}$$

Please 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 (15):

$$y(k) = w(k) * h(k). \tag{15}$$

Considering (13), Equation (15) can be rewritten as (16) and (17):

$$y(k) = \sum_{i=-\infty}^{\infty} \omega(i)\delta(k-i) * h(k) \tag{16}$$

$$y(k) = \sum_{i=-\infty}^{-1} \omega(i)\delta(k-i) * h(k) + \omega(0)\delta(k) * h(k) + \sum_{i=1}^{\infty} \omega(i)\delta(k-i) * h(k). \tag{17}$$

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 (17), it is possible to verify that $y(k)$ could result in a weighted impulse response of $h(k)$ if conditions (18) and (19) were met:

$$\omega(0) \neq 0 \tag{18}$$

$$\omega(i) = 0, \quad \forall i \neq 0, \tag{19}$$

which would make it straightforward to reveal $h(k)$ by measuring $y(k)$. However, although condition (18) is possible, condition (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 (19). Its goal is to produce a signal derived from $y(k)$ that can reveal $h(k)$ in the same way as if conditions (18) and (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 by other weighted time-shifted impulse responses of $h(k)$ contained in $y(k)$. Therefore, 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) ≥ Ω then
            j ← 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. Please 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 (17), $y_j(k)$ can be written as (20):

$$y_j(k) = \sum_{i=-\infty}^{-1} \omega_j(i)\delta(k-i) * h(k) + \omega_j(0)\delta(k) * h(k) + \sum_{i=1}^{\infty} \omega_j(i)\delta(k-i) * h(k) \tag{20}$$

wherein $\omega_j(i)$, defined according to (21), are the advanced (left-shifted) amplitudes of the white gaussian noise (13):

$$\omega_j(i) = \omega(i+j). \tag{21}$$

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 $j \in J$. Therefore, analogously to the RPI process, the mean $\Upsilon(k)$ of all $y_j(k)$ is computed according to (22):

$$\Upsilon(k) = \frac{\sum\limits_{j \in J} y_j(k)}{|J|}, \tag{22}$$

Thus, considering (20), Equation (22) can be rewritten as (23):

$$\Upsilon(k) = \underbrace{\frac{\sum\limits_{j \in J}\left[\sum\limits_{i=-\infty}^{-1} \omega_j(i)\delta(k-i) * h(k)\right]}{|J|}}_{\Upsilon_1(k)} + \underbrace{\frac{\sum\limits_{j \in J} \omega_j(0)\delta(k) * h(k)}{|J|}}_{\Upsilon_2(k)} + \underbrace{\frac{\sum\limits_{j \in J}\left[\sum\limits_{i=1}^{\infty} \omega_j(i)\delta(k-i) * h(k)\right]}{|J|}}_{\Upsilon_3(k)} \tag{23}$$

Please note that $\omega_j(i)$ has the same probability distribution function of $\omega(i)$ (i.e., $\omega_j(i) \sim N(\mu, \sigma)$) since, according to (21), $\omega_j(i)$ consists of the same amplitudes of $\omega(i)$, but 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, $\mathrm{Y}_2(k) \neq 0$ since that the mean of $\omega_j(0)$, among all $j \in J$, is different from zero. Please note that according to (21) $\omega_j(0) = \omega(j)$. From Algorithm 2, $w(j) \geq \Omega$ which, according to (13), means that $\omega(j) \geq \Omega$. Therefore, $\omega_j(0) \geq \Omega, \forall j$. This reasoning demonstrates that the mean of all $\omega_j(0)$ is greater than Ω and, therefore, $\mathrm{Y}_2(k) \neq 0$. 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, (23) can be simplified as (24):

$$\mathrm{Y}(k) = \bar{\omega}_j(0)\delta(k) * h(k), \tag{24}$$

wherein $\bar{\omega}_j(0)$ is the mean of all $\omega_j(0)$, according to (25):

$$\bar{\omega}_j(0) = \frac{\sum\limits_{j \in J} \omega_j(0)}{|J|}. \tag{25}$$

An example of the computation performed by the NII technique is represented in Figures 8 and 9. In this example, the transfer function $H(z)$ (26):

$$H(z) = \mathcal{Z}[h(k)] = \frac{z^3 - 2.546z^2 + 2.111z - 0.5646}{z^3 - 2.489z^2 + 2.102z - 0.6113}, \tag{26}$$

is excited by the white gaussian noise $w(k)$ (13), with $\omega(i) \sim N(0, 0.005)$, thereby providing the output $y(k) = w(k) * h(k)$. The amplitude threshold Ω of Algorithm 2, used to obtain all $y_j(k)$ from $y(k)$, is $\Omega = 0.01$. Figure 8 shows all $y_j(k)$ aligned side by side to be integrated (similarly to the representation shown in Figure 7 for the RPI process). This figure—when compared to Figure 7—depicts the analogy between the NII and the RPI techniques, showing signals whose uncorrelated noise can be cancelled through (22) in order to obtain the desired information—which here, in the NII technique, is the weighted impulse response $\mathrm{Y}(k)$ of $H(z)$.

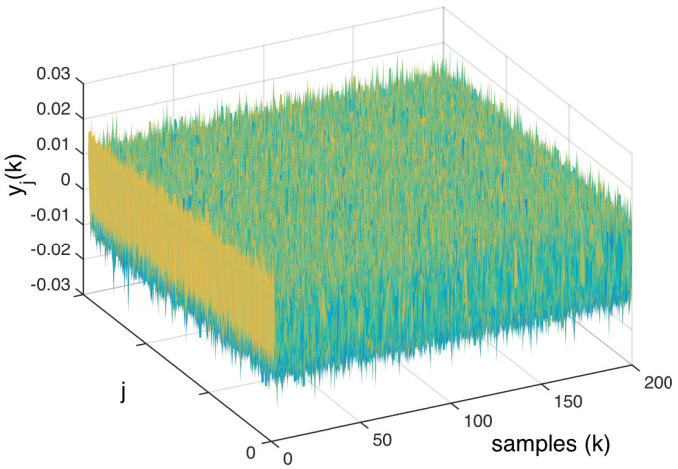


Figure 8. Signals $y_j(k)$ aligned to be integrated.

Figure 9 shows the signal $\mathrm{Y}(k)$ produced by the computation of (22) using the set of signals represented in Figure 8. The signal $\mathrm{Y}(k)$, highlighted in red, is the result of the integration of all $y_j(k)$ which, according to (24), reveals the impulse response of the system as it was excited by the impulse $\bar{\omega}_j(0)\delta(k)$. To graphically compare the magnitude of $\mathrm{Y}(k)$ with the noise magnitude, Figure 9 also shows all $y_j(k)$ of overlapped in black—as a front view of Figure 8. Therefore, in Figure 9, $\mathrm{Y}(k)$ (in red)

is the signal of interest, which is extracted from all noisy signals $y_j(k)$ (overlapped in black) when the uncorrelated noise of all $y_j(k)$ is cancelled by computing (22).

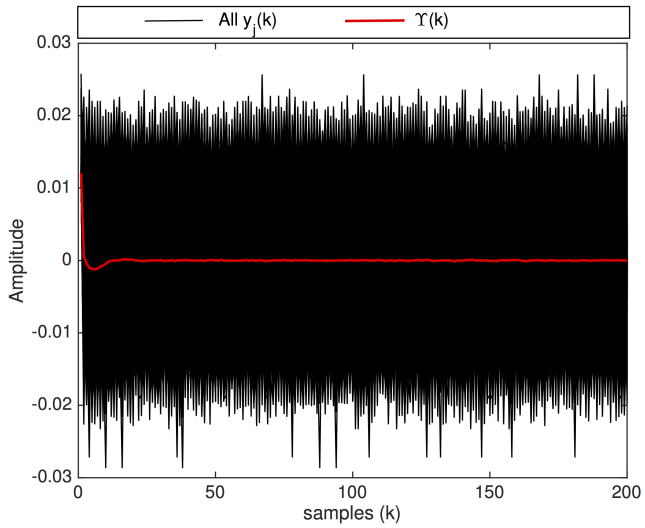


Figure 9. The impulse response $\Upsilon(k)$ (in red) produced by the NII technique after the integration of a set of signals $y_j(k)$ (shown overlapped in black).

As previously discussed, the NII technique is herein used to complement the attack identification strategy described in Section 3.1 to improve its accuracy. To do so, let us 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 3.1 and indicated in Figure 3;
- $h(k)$ is the transfer function between $w(k)$ and $y_1''(k)$ which, according to (8), is defined as (27):

$$h(k) = \mathcal{Z}^{-1}\left[M(z)\left(\frac{1+C(z)P(z)}{1+C(z)P(z)M(z)}\right)\right]. \tag{27}$$

Doing so, (24) can be rewritten as (28):

$$\Upsilon(k) = \bar{\omega}_j(0)\delta(k) * \mathcal{Z}^{-1}\left[M(z)\left(\frac{1+C(z)P(z)}{1+C(z)P(z)M(z)}\right)\right], \tag{28}$$

which can now be used to estimate $M(z)$ in the same way as in Section 3.1 for equation (8). Please note that the differences between (8) and (28) are:

- the input of (8) is a white gaussian noise and its output is a white gaussian noise filtered by $h(k)$;
- the input of (28) is a weighted impulse signal and its output is a weighted impulse response of $h(k)$.

Now, given (28), the attack function $M(z)$ can be estimated by an optimization algorithm (e.g., the BSA), such as described in Section 3.1. In this case, if $C(z)$ and $P(z)$ are known (which is feasible for the NCS owner), $M(z)$ can be estimated by applying $\bar{\omega}_j(0)\delta(k)$ in an estimated system, defined by (29):

$$\hat{\Upsilon}(k) = \bar{\omega}_j(0)\delta(k) * \mathcal{Z}^{-1}\left[M_e(z)\left(\frac{1+C(z)P(z)}{1+C(z)P(z)M_e(z)}\right)\right], \tag{29}$$

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 (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 $\hat{Y}(k)$ with $Y(k)$, the Attack Identification process can evaluate whether $M_e(z)$ is equal to/approximately $M(z)$.

In the same way as in Section 3.1, to discover $M(z)$, the coefficients of $M_e(z)$ are adjusted by the BSA until the estimated output $\hat{Y}(k)$ converges to $Y(k)$ (the latter obtained by the NII technique from measurements of $y''(k)$ and $w(k)$ in the real NCS). Let $\hat{Y}j(k)$ be the output of the estimated model (29) in face of the input $\bar{\omega}_j(0)\delta(k)$, when the coefficients of $M_e(z)$ (10) 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 $\hat{Y}j(k)$ with $Y(k)$, according to (30):

$$f_j = \frac{\sum_{k=0}^{\mathcal{N}} (Y(k) - \hat{Y}j(k))^2}{\mathcal{N}}, \tag{30}$$

wherein $\mathcal{N}$ is the number of samples that exist in $Y(k)$. As already discussed in Section 3.1, 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. Please 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 $Y(k)$ as input signals to the BSA-based identification.

Algorithm 3: Attack Identification with the NII technique.

```
begin
    if y''(k) is in steady state regarding r(k) then
        Record w(k) and y''(k) during T seconds;
        Compute ȳ'' according to (5);
        Compute y''_1(k) according to (7);
        NII stage:
            Obtain a set of y_j(k) signals from y''_1(k) and w(k) using Algorithm 2;
            Compute Y(k) according to equation (22);
            Compute ω̄_j(0) according to equation (25);
        end
        Execute BSA, using ω̄_j(0)δ(k) and Y(k) to find M_e(z) based on (10), (29) and (30).
    end if
end
```

4. Results

This section analyses the performance of the attack identification strategy proposed in section 3 when identifying the Controlled Data Injection attack characterized in Section 2. The evaluation on the accuracy of the countermeasure is based on results obtained through simulations using MATLAB/SIMULINK. First, Section 4.1 describes the attacked NCS and the attack parameters. Then, Section 4.2 presents the results obtained by the proposed countermeasure in the scenario described in Section 4.1.

4.1. 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 3. The system consists of Proportional-Integral (PI) controller that controls the rotational speed of a DC motor – which has broad applications in industry and real-world systems, and has been widely used in previous works about NCS [8,49–52]. The control function $C(z)$ and the plant transfer function $P(z)$ are the same as in [8,50], which are represented by (31):

$$C(z) = \frac{0.1701z - 0.1673}{z - 1} \quad P(z) = \frac{0.3379z + 0.2793}{z^2 - 1.5462z + 0.5646} \tag{31}$$

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

As discussed in [8], one way to degrade the service of a plant is by causing overshoots during its transient response, which, indeed, can cause stress and possibly damage a variety of physical systems [53]. 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. To achieve this goal, a MitM located in the feedback link runs the attack function represented by (32), wherein $\alpha_0 = 0.25$ and $\beta_0 = -0.75$:

$$M(z) = \frac{\alpha_0}{z + \beta_0}. \tag{32}$$

4.2. Performance of the Attack Identification

Section 3 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 4.1. 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 10 shows examples of the system output (the motor speed) with and without the attack. Please 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 (1) when $M(z) = 1$).

As previously discussed, the present attack identification scheme aims to estimate the coefficients of $M(z)$, which according to (32) are α_0 and β_0. The BSA settings in both Algorithms 1 and 3 are the same as those used in [8,45]: the lower and upper limits of each search space dimension are -10 and 10, respectively; the BSA population has 100 individuals; and $\eta = 1$ (in the BSA, η is used to define the amplitude of the displacement of the individuals).

The BSA is executed for 600 iterations.

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 (9) and (11), respectively, is $N = 100$ samples. For the execution of Algorithm 3 the signals $w(k)$ and $y''(k)$ are recorded during $0,5Msamples$, also starting when the system achieves its steady state regarding to $r(k)$. Recall that in 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 (29) and (30), 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 $w(k)$.

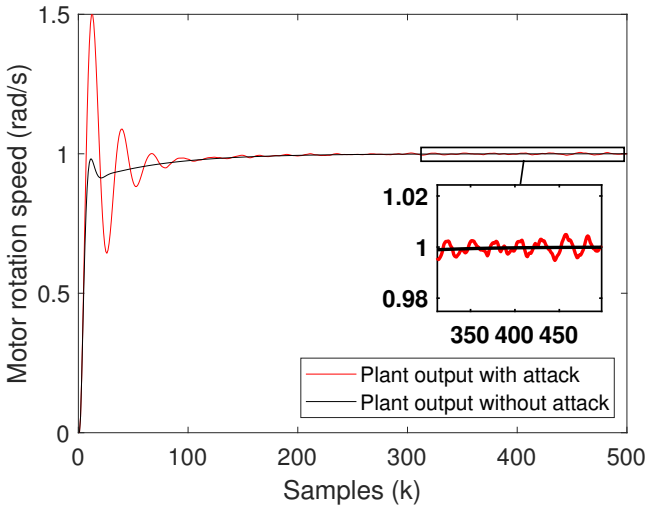
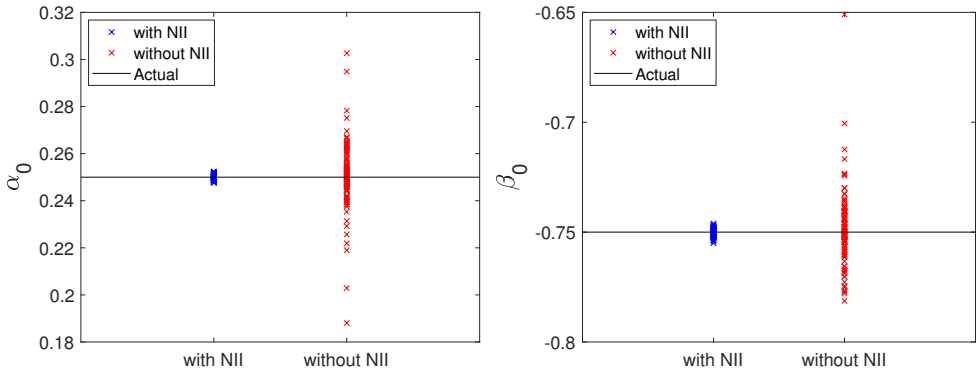


Figure 10. Motor speed with and without attack.

Figure 11 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 1 shows the statistics of the results presented in Figure 11. From Figure 11 and Table 1, 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 11 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. Please note that the statistics shown in Table 1 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.

Table 1. Statistics of the attack identification process.

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



(**a**) Estimations of α_0 (**b**) Estimations of β_0

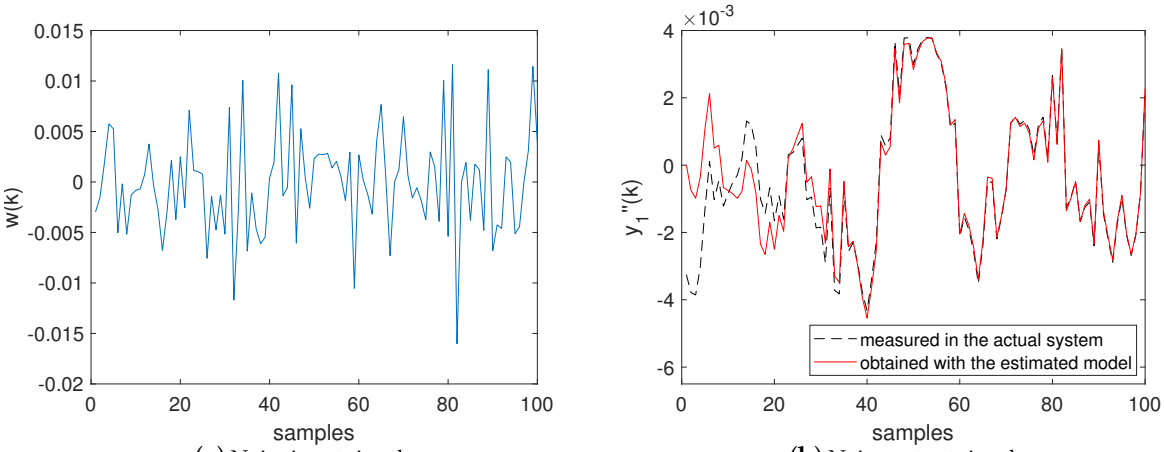
Figure 11. Estimations of α_0 and β_0 with and without the NII stage.

Figure 12 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 12a shows the noise $w(k)$ recorded in the actual system and used by the BSA as input for the model defined by (8). Figure 12b 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 (8). Additionally, Figure 12b shows in red line the signal $\hat{y}_1''(k)$ produced by the estimated model—i.e., the model (9) containing the estimated attack function—when excited by the noise input shown in Figure 12a. In Figure 12b, it is possible to see that the output $\hat{y}_1''(k)$ 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 11 and Table 1, the lower accuracy of Algorithm 1 when identifying $M(z)$.

Figure 13, 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 13a 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 (28). Figure 13b shows:

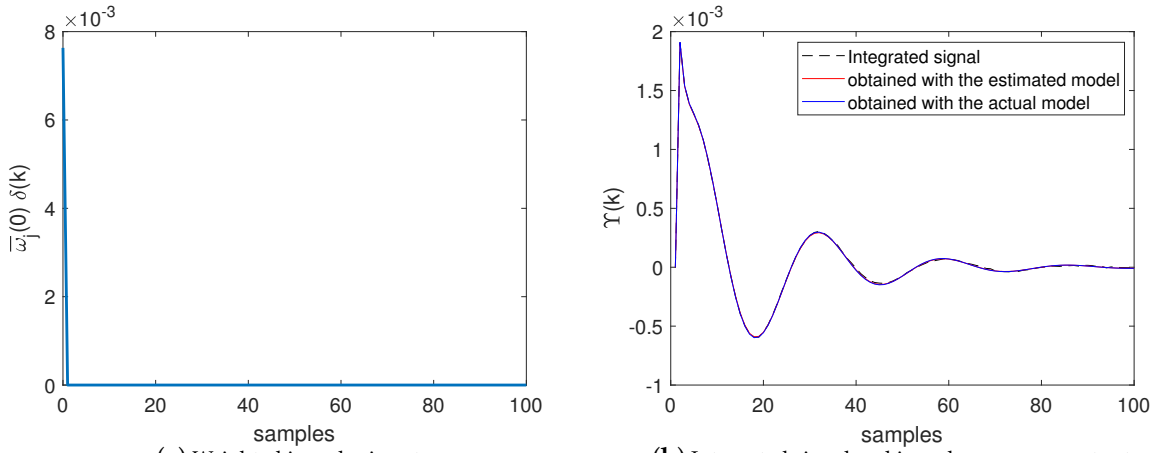
- In black dashed line: the integrated signal $\mathrm{Y}(k)$ produced by the NII stage (based on measurements in the actual system) and used by the BSA as the reference output for the model defined in (28);
- In blue line: the impulse response produced when the weighted impulse $\bar{\omega}_j(0)\delta(k)$, shown Figure 13a, is applied to the system defined in (28) containing the actual attack function;
- In red line: the impulse response produced when the weighted impulse $\bar{\omega}_j(0)\delta(k)$, shown Figure 13a, is applied to the system defined in (29) containing the estimated attack function.

From Figure 13b, 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 can accurately reveal the impulse response of the system based on the signals produced by the white gaussian noise injected in the NCS. Additionally, Figure 13b 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 11 and Table 1.



(**a**) Noise input signal (**b**) Noisy output signals

Figure 12. Input and output signals used by the BSA in Algorithm 1 to estimate $M(z)$ considering the model defined in (8).



(**a**) Weighted impulse input (**b**) Integrated signal and impulse response outputs

Figure 13. Input and output signals used by the BSA in Algorithm 3 to estimate $M(z)$ considering the model defined in (28).

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. Please 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 (23), where $\Upsilon_1(k) \rightarrow 0$ when all $y_j(k)$ are integrated among all $j \in J$, as demonstrated in Section 3.2.2. 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 (28), given that $\bar{\omega}_j(0)\delta(k) = 0, \forall -\infty \leq k < 0$.

Additionally, the performance of the proposed countermeasure is evaluated in scenarios where the reference signal is slowly changing during the execution of Algorithm 3. For this purpose, the reference signal of the system described in Section 4.1 is changed to (33):

$$r(k) = \begin{cases} 0 & k < 0; \\ 1 + A\ sin(0.01k) & k \geq 0. \end{cases} \tag{33}$$

where three different amplitudes A are considered: 0.001, 0.01, and 0.1 (i.e., 0.1%, 1% and 10% of the unitary step function setpoint, respectively). Please note that the reference signal of the original system described in Section 4.1 corresponds to the case where $A = 0$. For statistical analysis, 100 different simulations are provided for each amplitude A. Figure 14 shows examples of output signals $y'(k)$ obtained in simulations using different amplitudes A in (33).

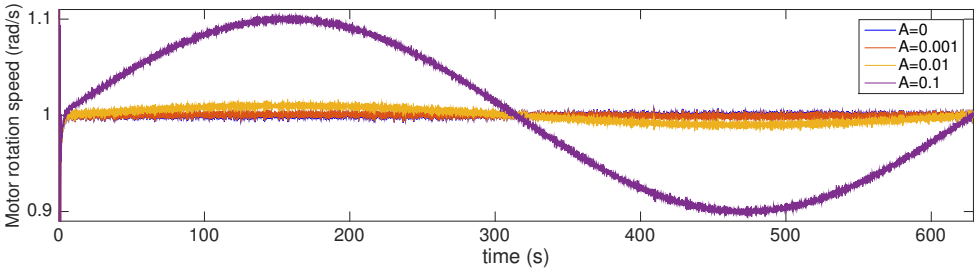


Figure 14. Examples of sensor outputs $y'(k)$ in simulations with different amplitudes A.

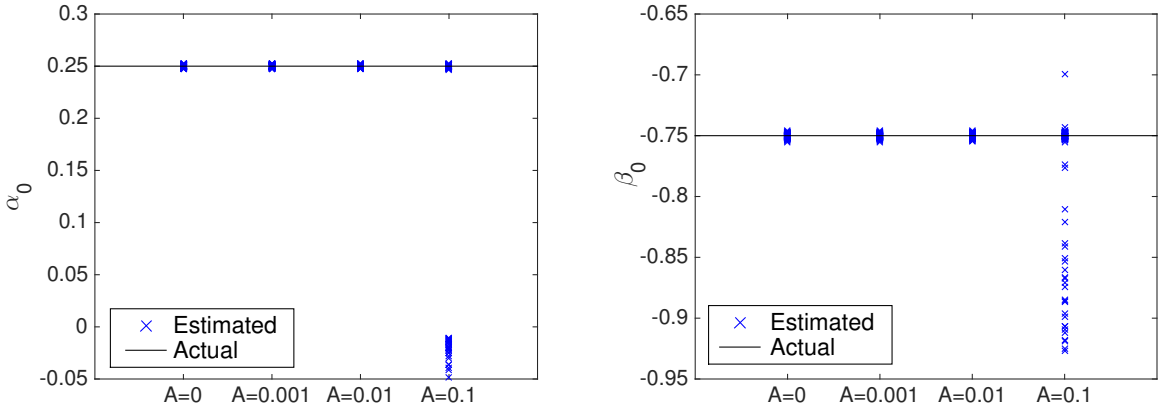
Figure 15 compares the coefficients estimated by Algorithm 3 when the reference signal is constant ($A = 0$) and when it slowly varies using different amplitude values A. Additionally, Table 2 shows the statistics of the results presented in Figure 15. From Figure 15 and Table 2, it is possible to verify that the accuracy of the attack identification algorithm is not affected by variations in the reference signal when $A = 0.001$ and $A = 0.01$. In these cases, as shown in Figure 15, the estimated coefficients are quite close to their actual values and are as accurate as when $A = 0$ (when there are no variations in $r(k)$). The statistics shown in Table 2 ratifies that for $A = 0.001$ and $A = 0.01$, the algorithm presents the same performance as when $r(k)$ is not varying—the means and the standard deviations are practically the same as when $A = 0$.

Lower performance is verified when the amplitude is increased to $A = 0.1$. In this case, 29% of the estimated coefficients have their accuracy affected—these outliers can be seen in Figure 15 (specially in Figure 15a) far from the coefficients' actual values. According to Table 2, when these outliers are taken into account, the means of the estimated coefficients diverge from the actual values and the standard deviations increase. However, it is worth mentioning that when $A = 0.1$, even with the reduced performance, 71% of the results are not affected by the variations in $r(k)$ and the estimated coefficients are as accurate as when $A = 0$. Figure 15 shows these 71% of estimated coefficients close to their actual values. Moreover, Table 2 shows that if the outliers are not taken into account, the means and the standard deviations are practically the same as when $A = 0$. These results suggest that even when the reference signal is (slowly) varying $\pm 10\%$, the proposed algorithm can provide satisfactory performance in most cases (71%).

For the sake of comparison with Figure 13b, Figure 16 brings examples of results obtained in scenarios with different amplitudes A. Through these examples, it is possible to visualize the impact of the different amplitudes A in the integrated signal $\Upsilon(k)$ produced by the NII stage, and in the weighted impulse response of the system containing the estimated attack function.

Note in Figure 16 that even with variations in the reference signal, the NII technique is able to accurately reveal the impulse response of the system under attack (represented in black dashed line). However, it is possible to see the presence of an offset between the integrated signal and the impulse response of the actual system. This offset tends to increase as A becomes higher, and is caused by the different levels of $y''(k)$ (due the variations in $r(k)$) at the time when $w(k) \geq \Omega$ is satisfied and each $y_j(k)$ is obtained according to Algorithm 2. Still, even when such offset is higher (as in Figure 16c, when $A = 0.1$), the optimization metaheuristic in most cases (71%) is able to accurately find the coefficients of $M(z)$ by minimizing the difference between the impulse response of the system with the estimated model (represented in red) and the integrated signal (represented in black dashed line), making them parallel. In all examples shown in Figure 16, the impulse response of the system with

the estimated model (represented in red) converges to the impulse response of the system with the actual model (represented in blue), illustrating the good accuracy obtained by the countermeasure even when $r(k)$ is varying.



(a) Estimations of α_0 **(b)** Estimations of β_0

Figure 15. Estimations of α_0 and β_0 using the NII technique, with and without variations in the reference signal $r(k)$.

Table 2. Statistics of the attack identification process using the NII technique, with and without variations in the reference signal $r(k)$.

Coefficient	A	Mean	Standard Deviation
α_0	0 *	0.2500	0.0011
	0.001	0.2500	0.0011
	0.01	0.2500	0.0011
	0.1	0.1712	0.1241
	0.1 **	0.2500	0.0012
β_0	0 *	−0.7502	0.0017
	0.001	−0.7502	0.0017
	0.01	−0.7501	0.0017
	0.1	−0.7812	0.0586
	0.1 **	−0.7500	0.0020

* The reference signal $r(k)$ does not vary. ** Not taking into account the outliers verified in Figure 15. (These statistics represent the remaining 71% of the results.

The results of this section indicate the effectiveness and accuracy of the proposed countermeasure when identifying SD-Controlled Data Injection attacks in NCSs, especially when the NII technique is used. The performance of the countermeasure, designed for scenarios where the reference signal remains constant, is evaluated considering $r(k)$ as a step function. Additionally, the countermeasure is also evaluated in scenarios where the reference signal slowly changes, causing oscillations in the monitored signals (which in the present work are sensor measurements). In both cases, accurate results are obtained, with a performance reduction when the oscillation amplitude is increased to $A = 0.1$. The results indicate that if the offsets shown in Figure 16 (introduced by variations in $r(k)$) are reduced, or taken into account in the BSA optimization process, the performance of the countermeasure can be further improved for reference signals varying with higher amplitudes—encouraging research in this direction.

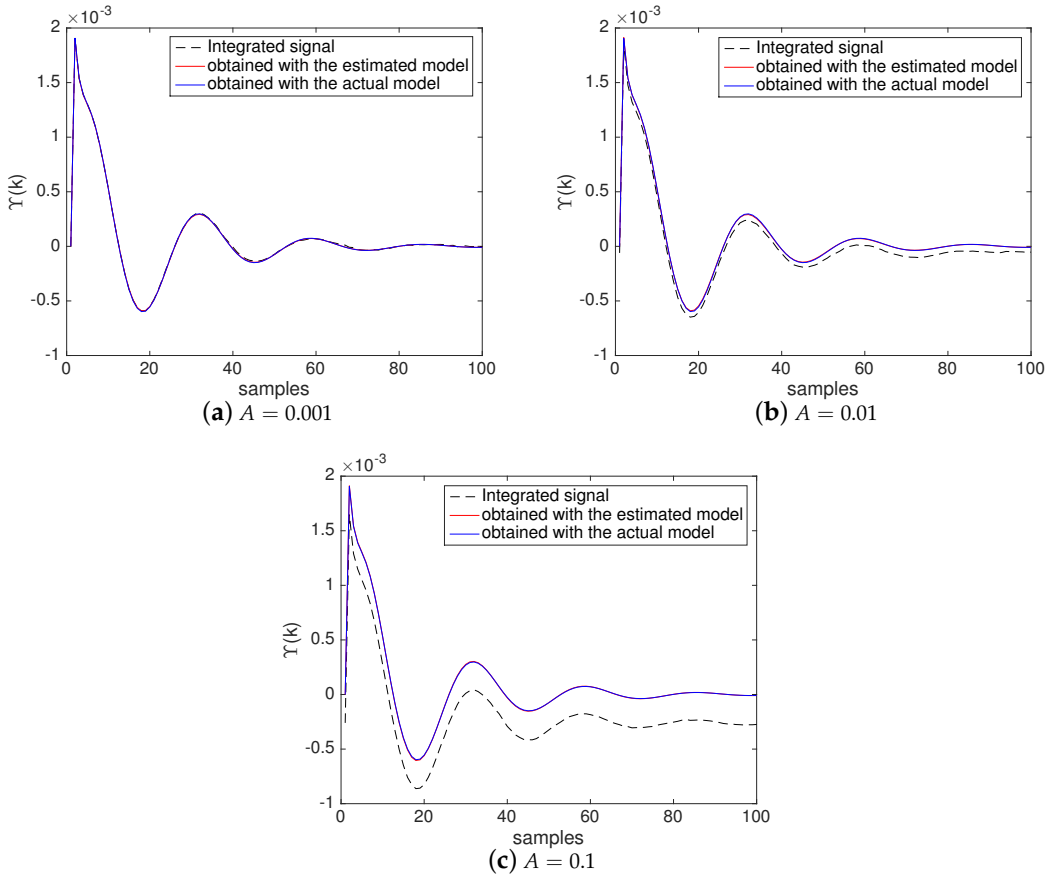


Figure 16. Examples of integrated signals and weighted impulse responses obtained through Algorithm 3 in scenarios with different amplitudes A.

Moreover, it is worth mentioning that 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.

5. Conclusions

This paper proposes a BSA-based countermeasure to identify LTI attack functions executed during SD-Controlled Data Injection attacks in NCSs. 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. The proposed solution is evaluated through simulations where the attacker aims to manipulate the measurements transmitted by the plant sensor. It 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 when an attack is occurring. In this case, the presence of noise may not necessarily be a drawback, but the possible impacts of such noise (in case of attack) must be evaluated according to the requirements of each specific plant.

Additionally, to increase the accuracy of the proposed countermeasure, this paper introduces the NII technique which is developed using the radar pulse integration process as inspiration. It is proven that the NII technique can accurately reveal the impulse response of the system under attack based on the signals produced by the white gaussian noise injected in the NCS. The results indicate that the NII

technique indeed increases the accuracy of the attack identification, eliminating the need to estimate the initial conditions caused by the noise injected into the NCS.

Although the proposed countermeasure is designed for scenarios where the reference signal remains constant, we also evaluate it in scenarios where the reference signal slowly changes. In both cases, accurate results can be obtained, with a performance reduction when the oscillation of the reference signal increases. With this regard, the results indicate that if the integrated signal offset (introduced by variations in $r(k)$) is reduced, or taken into account in the BSA optimization process, the performance of the countermeasure can be further improved for reference signals varying with higher amplitudes. Therefore, we encourage future research in this direction.

As future work we plan to investigate the possibility of generalizing the proposed countermeasure, to be able to run the identification task when the system is either in steady or transient operating conditions. Moreover, we plan to evaluate the performance of the proposed countermeasure in identifying switching LTI attack functions.

Also, we consider to investigate the use of the NII technique as an attack tool for System Identification attacks [8,13] in scenarios with high data loss. In this paper, the technique is used to enhance the performance of a countermeasure in scenarios where the monitored signals are not impaired by data loss. However, we consider 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 [8,13] in scenarios with extreme data loss—as in the case of an attacker far from WNCS transmitters, with poor connectivity, trying to identify the WNCS models.

Author Contributions: All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by the Brazilian research agencies CNPq and FAPERJ, by the SHCDCiber project, 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).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Lasi, H.; Fettke, P.; Kemper, H.G.; Feld, T.; Hoffmann, M. Industry 4.0. *Bus. Inf. Syst. Eng.* **2014**, *6*, 239–242. [CrossRef]
2. Jazdi, N. Cyber physical systems in the context of Industry 4.0. In Proceedings of the 2014 IEEE International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, Romania, 22–24 May 2014; pp. 1–4.
3. Latrech, C.; Chaibet, A.; Boukhnifer, M.; Glaser, S. Integrated longitudinal and lateral networked control system design for vehicle platooning. *Sensors* **2018**, *18*, 3085. [CrossRef] [PubMed]
4. Ju, H.H.; Long, Y.; Wang, H. Reliable Finite Frequency Filter Design for Networked Control Systems with Sensor Faults. *Sensors* **2012**, *12*, 7975–7993. [CrossRef] [PubMed]
5. Santos, C.; Martínez-Rey, M.; Espinosa, F.; Gardel, A.; Santiso, E. Event-based sensing and control for remote robot guidance: An experimental case. *Sensors* **2017**, *17*, 2034. [CrossRef]
6. Dasgupta, S.; Halder, K.; Banerjee, S.; Gupta, A. Stability of Networked Control System (NCS) with discrete time-driven PID controllers. *Control Eng. Pract.* **2015**, *42*, 41–49. [CrossRef]
7. McLaughlin, S.; Konstantinou, C.; Wang, X.; Davi, L.; Sadeghi, A.R.; Maniatakos, M.; Karri, R. The cybersecurity landscape in industrial control systems. *Proc. IEEE* **2016**, *104*, 1039–1057. [CrossRef]
8. de Sa, A.O.; da Costa Carmo, L.F.R.; Machado, R.C.S. Covert Attacks in Cyber-Physical Control Systems. *IEEE Trans. Ind. Inf.* **2017**, *13*, 1641–1651. [CrossRef]
9. Ferrari, P.; Flammini, A.; Rizzi, M.; Sisinni, E. Improving simulation of wireless networked control systems based on WirelessHART. *Comput. Stand. Interfac.* **2013**, *35*, 605–615. [CrossRef]
10. Das, M.; Ghosh, R.; Goswami, B.; Gupta, A.; Tiwari, A.; Balasubrmanian, R.; Chandra, A. Network control system applied to a large pressurized heavy water reactor. *IEEE Trans. Nucl. Sci.* **2006**, *53*, 2948–2956. [CrossRef]

11. 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*, 3879–3888. [CrossRef]
12. Smith, R.S. Covert Misappropriation of Networked Control Systems: Presenting a Feedback Structure. *Control Syst. IEEE* **2015**, *35*, 82–92.
13. De Sa, A.O.; da Costa Carmo, L.F.R.; Machado, R.C.S. Bio-inspired Active System Identification: A Cyber-Physical Intelligence Attack in Networked Control Systems. *Mob. Netw. Appl.* **2017**, 1–14. [CrossRef]
14. Langner, R. Stuxnet: Dissecting a cyberwarfare weapon. *Secur. Priv. IEEE* **2011**, *9*, 49–51. [CrossRef]
15. Smith, R. A decoupled feedback structure for covertly appropriating networked control systems. In Proceedings of the 18th IFAC World Congress 2011, IFAC-PapersOnLine, Milano, Italy, 28 August–2 September 2011.
16. Teixeira, A.; Shames, I.; Sandberg, H.; Johansson, K.H. A secure control framework for resource-limited adversaries. *Automatica* **2015**, *51*, 135–148. [CrossRef]
17. Zetter, K. *Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon*; Crown: New York, NY, USA, 2014.
18. Falliere, N.; Murchu, L.O.; Chien, E. W32. stuxnet dossier. *White Pap. Symantec Corp. Secur. Response* **2011**, *5*, 29.
19. Muller, I.; Netto, J.C.; Pereira, C.E. WirelessHART field devices. *IEEE Instrum. Meas. Mag.* **2011**, *14*, 20–25. [CrossRef]
20. Petersen, S.; Carlsen, S. WirelessHART Versus ISA100. 11a: The Format War Hits the Factory Floor. *IEEE Ind. Electron. Mag.* **2011**, *4*, 23–34. [CrossRef]
21. Collantes, M.H.; Padilla, A.L. *Protocols and Network Security in ICS Infrastructures*; Technical Report; Spanish National Institute for Cyber-Security (INCIBE): León, Spain, 2015.
22. Peschke, J.; Reinelt, D.; Yumin, W.; Treytl, A. Security in industrial ethernet. In Proceedings of the 11th IEEE International Conference on Emerging Technologies and Factory Automation, Prague, Czech Republic, 20–22 September 2006; pp. 1214–1221.
23. Granat, A.; HÖFKEN, H.; Schuba, M. Intrusion Detection of the ICS Protocol EtherCAT. In Proceedings of the 2nd International Conference on Computer, Network Security and Communication Engineering (CNSCE 2017), Bangkok, Thailand, 26–27 March 2017; pp. 113–117.
24. Ovaz Akpinar, K.; Ozcelik, I. Development of the ECAT Preprocessor with the Trust Communication Approach. *Secur. Commun. Netw.* **2018**, *2018*, 2639750. [CrossRef]
25. Yung, J.; Debar, H.; Granboulan, L. Security Issues and Mitigation in Ethernet POWERLINK. In Proceedings of the Conference on Security of Industrial-Control-and Cyber-Physical Systems, Crete, Greece, 26–30 September 2016; pp. 87–102.
26. Mathur, A.P.; Tippenhauer, N.O. SWaT: a water treatment testbed for research and training on ICS security. In Proceedings of the 2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater), Vienna, Austria, 11 April 2016; pp. 31–36.
27. Pfrang, S.; Meier, D. On the Detection of Replay Attacks in Industrial Automation Networks Operated with Profinet IO. In Proceedings of the ICISSP, Porto, Portugal, 9–21 February 2017; pp. 683–693.
28. Akerberg, J.; Bjorkman, M. Exploring security in PROFINET IO. In Proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications Conference, Seattle, WA, USA, 20–24 July 2009; Volume 1, pp. 406–412.
29. de Sa, A.O.; da Costa Carmo, L.F.R.; Machado, R.C.S. A controller design for mitigation of passive system identification attacks in networked control systems. *J. Int. Serv. Appl.* **2018**, *9*, 1–19. [CrossRef]
30. Rubio-Hernan, J.; Rodolfo-Mejias, J.; Garcia-Alfaro, J. Security of cyber-physical systems. In Proceedings of the International Workshop on the Security of Industrial Control Systems and Cyber-Physical Systems, Crete, Greece, 26–30 September 2016; pp. 3–18.
31. Stouffer, K.; Pillitteri, V.; Lightman, S.; Abrams, M.; Hahn, A. *NIST Special Publication 800-82, Revision 2: Guide to Industrial Control Systems (ICS) Security*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2015.
32. Pang, Z.H.; Liu, G.P. Design and implementation of secure networked predictive control systems under deception attacks. *IEEE Trans. Control Syst. Technol.* **2012**, *20*, 1334–1342. [CrossRef]

33. Gerard, B.; Rebaï, S.B.; Voos, H.; Darouach, M. Cyber security and vulnerability analysis of networked control system subject to false-data injection. In Proceedings of the 2018 Annual American Control Conference (ACC), Milwaukee, WI, USA, 27–29 June 2018; pp. 992–997.
34. Miao, F.; Zhu, Q.; Pajic, M.; Pappas, G.J. Coding sensor outputs for injection attacks detection. In Proceedings of the 53rd IEEE Conference on Decision and Control, Los Angeles, CA, USA, 15–17 December, 2014; pp. 5776–5781.
35. Dhunna, G.S.; Al-Anbagi, I. A Low Power WSNs Attack Detection and Isolation Mechanism for Critical Smart Grid Applications. *IEEE Sens. J.* **2019**, *19*, 5315–5324. [CrossRef]
36. Rigatos, G.; Serpanos, D.; Zervos, N. Detection of attacks against power grid sensors using Kalman filter and statistical decision making. *IEEE Sens. J.* **2017**, *17*, 7641–7648. [CrossRef]
37. Mo, Y.; Weerakkody, S.; Sinopoli, B. Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs. *IEEE Control Syst. Mag.* **2015**, *35*, 93–109.
38. Mo, Y.; Sinopoli, B. Secure control against replay attacks. In Proceedings of the 2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, VA, USA, 30 September 2009; pp. 911–918.
39. Mo, Y.; Chabukswar, R.; Sinopoli, B. Detecting integrity attacks on SCADA systems. *IEEE Trans. Control Syst. Technol.* **2014**, *22*, 1396–1407.
40. Ferrari, R.M.; Teixeira, A.M. Detection and isolation of replay attacks through sensor watermarking. *IFAC-PapersOnLine* **2017**, *50*, 7363–7368. [CrossRef]
41. Krombholz, K.; Hobel, H.; Huber, M.; Weippl, E. Advanced social engineering attacks. *J. Inf. Secur. Appl.* **2015**, *22*, 113–122. [CrossRef]
42. Skolnik, M.I. *Radar Handbook*; Electronic Engineering Series; McGraw-Hill: New York, NY, USA, 1990.
43. Civicioglu, P. Backtracking search optimization algorithm for numerical optimization problems. *Appl. Math. Comput.* **2013**, *219*, 8121–8144. [CrossRef]
44. Tulleken, H.J. Generalized binary noise test-signal concept for improved identification-experiment design. *Automatica* **1990**, *26*, 37–49. [CrossRef]
45. de Sá, A.O.; Carmo, L.F.R.d.C.; Machado, R.C.S. Countermeasure for Identification of Controlled Data Injection Attacks in Networked Control Systems. In Proceedings of the 2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4. 0&IoT), Naples, Italy, 4–6 June 2019; pp. 455–459.
46. Stallings, W. *Cryptography and Network Security: Principles and Practices*; Pearson Education India: Upper Saddle River, NJ, USA, 2006.
47. Ahmed, S. Novel noncoherent radar pulse integration to combat noise jamming. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *51*, 2350–2359. [CrossRef]
48. Schwartz, M. Effects of signal fluctuation on the detection of pulse signals in noise. *IRE Trans. Inf. Theory* **1956**, *2*, 66–71. [CrossRef]
49. Chen, X.; Song, Y.; Yu, J. Network-in-the-Loop Simulation Platform for Control System. In *AsiaSim 2012*; Springer: Shanghai, China, 27 October 2012; pp. 54–62.
50. Long, M.; Wu, C.H.; Hung, J.Y. Denial of service attacks on network-based control systems: impact and mitigation. *Ind. Inf. IEEE Trans.* **2005**, *1*, 85–96. [CrossRef]
51. Shi, Y.; Huang, J.; Yu, B. Robust tracking control of networked control systems: application to a networked DC motor. *IEEE Trans. Ind. Electron.* **2013**, *60*, 5864–5874. [CrossRef]
52. Si, M.L.; Li, H.X.; Chen, X.F.; Wang, G.H. Study on Sample Rate and Performance of a Networked Control System by Simulation. *Adv. Mater. Res. Trans. Tech. Publ.* **2010**, *139*, 2225–2228.
53. Tran, T.; Ha, Q.P.; Nguyen, H.T. Robust non-overshoot time responses using cascade sliding mode-pid control. *J. Adv. Comput. Intell. Intell. Inf.* **2007**, *11*, 1224–1231. [CrossRef]