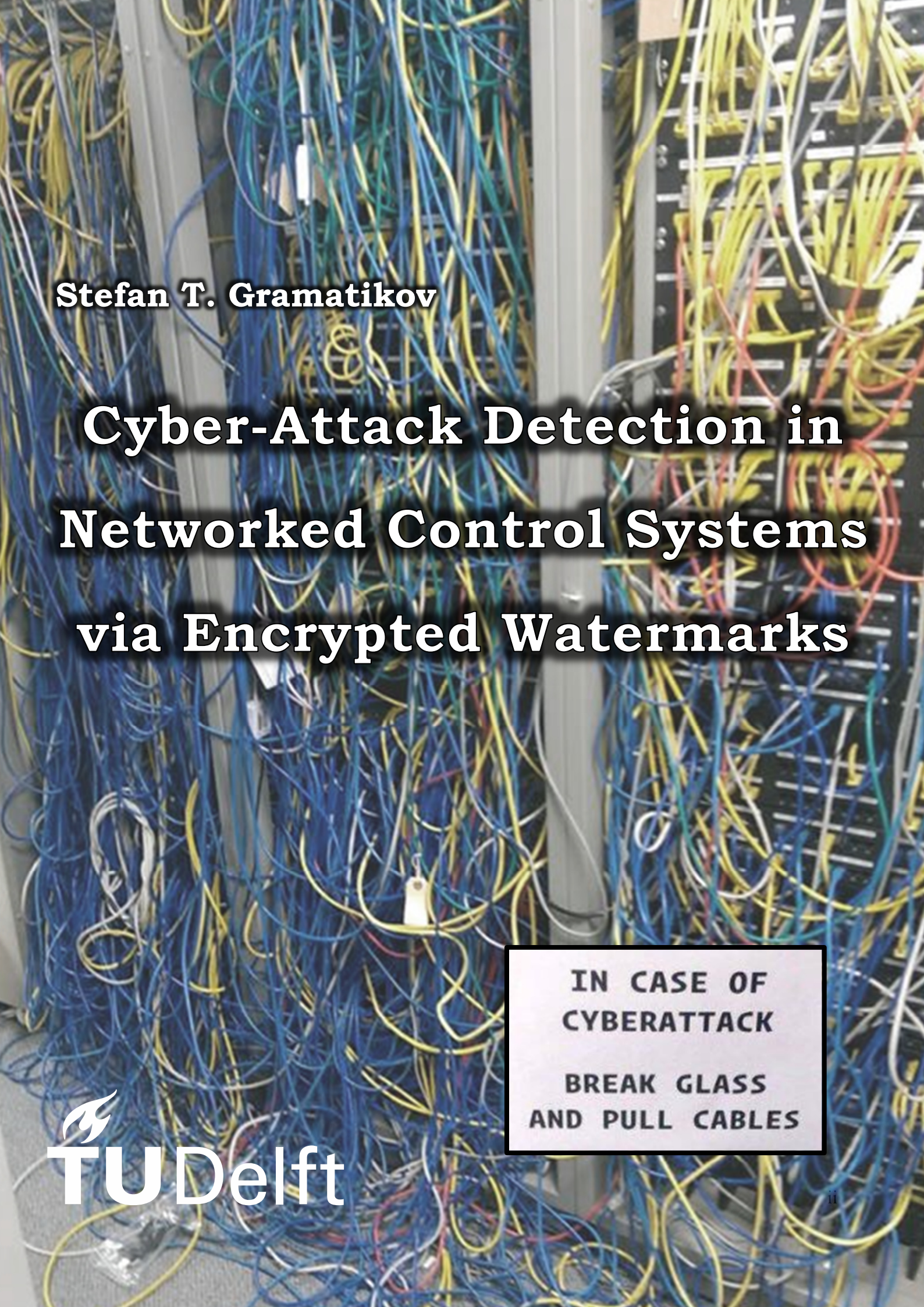
Stefan T. Gramatikov

# Cyber-Attack Detection in Networked Control Systems via Encrypted Watermarks




TU Delft

# Cyber-Attack Detection in Networked Control Systems via Encrypted Watermarks

By

Stefan T. Gramatikov

in partial fulfilment of the requirements for the degree of

**Master of Science**

in Systems & Control

at the Delft Center for Systems and Control,

Delft University of Technology,

to be defended publicly on Friday, September 25, 2020 at 14:30

| | |
|---|---|
| Supervisor: | Dr. Riccardo Ferrari |
| Thesis committee: | Dr.-Ing. Sander Wahls |
| | Jean Gonzalez Silva |



TU Delft

# Table of Contents

## Table of Tables



## Table of Figures

## Abstract

Cyber-attacks long have been a topic reserved for sci-fi movies and books. With the advance of internet and the globalisation of technology supply chains, as well as the growing political and economic pressure around the world, cyber warfare has become the new weapon of choice for covert state operations, but also rogue organizations. In the last 2 decades multiple major industries have suffered some kind of outage – power generation, manufacturing, oil & gas, transport, and others. A typical industrial system is designed for a life span of more than 20 years, making future issues hard to protect against at the planning phase. No widespread efforts exist to identify such threats, to detect attacks and counteract them. This thesis proposes a practical approach for lower level protection of control systems based on linear watermarking as a transparent process to provide detection for any malicious activity that might significantly impact the operations of the plant.

## Acknowledgements

I would like to thank all my professors for the provided insights at the lectures and talks over the last two years. Furthermore, I am grateful for the guidance of my supervisor, prof. Riccardo Ferrari, and for the opportunity to work on this interesting topic. This accomplishment would not have been possible without the continuous, faithful support of my parents, who have always encouraged me to pursue my dreams. Finally, I am glad I have made some great new friends and met many fantastic people along this journey!

Stefan T. Gramatikov

Delft, September 2020

# 1 Introduction to Industrial Control Systems (ICS)

The advance of the First Industrial Revolution saw certain physical processes become controlled and later automated – the most recognized early example being the steam boiler and the centrifugal governor in 1877. Over the next 130 years industry steadily developed with the addition of gas, electricity, and the production line, whereas technology advanced to provide better capabilities to the ever-increasing requirements set. Electronics and telecommunication allowed for large-scale operations that span beyond the ordinary local feedback control, operators' work moved away from active control to a supervisory role, while the Internet introduced a whole new dimension of interconnectivity.

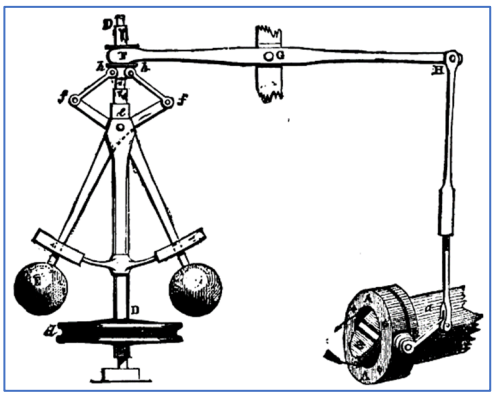


**Figure 1 Centrifugal Governor with Throttle Valve (Proportionate control) [1]**

Although ICS are viewed as a highly conservative field, more and more effort is being put to revolutionize the field through innovations from the Information Technology (IT) field, an approach which is prone to certain pitfalls, as it will be discussed in the next chapter. Contemporary ICSs can be classified [2] in two main categories: manufacturing – the transformation of raw materials into finished goods, and distribution – the delivery of materials (water, oil) or energy (electricity) to the users that require them. Regardless of the end goal both types employ a mix of Distributed Control Systems (DCS) and Supervisory Control and Data Acquisition Systems that utilize a plethora of communication channels to function.

## 1.1 Physical Processes and Control

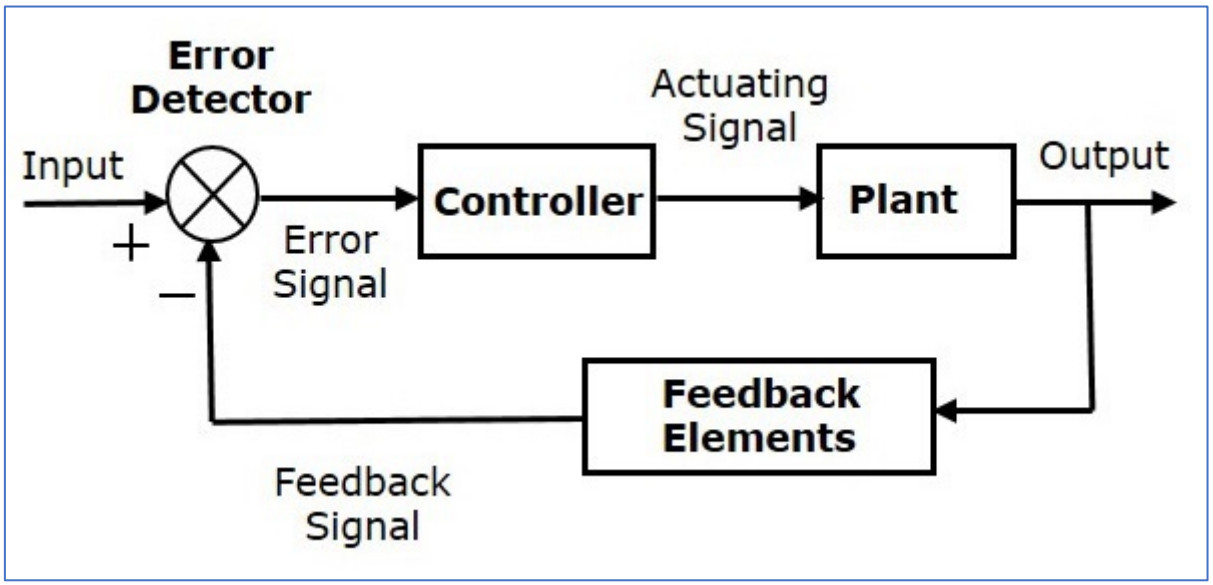



**Figure 2 A typical control loop [3]**

In the field of control a physical process is exactly what its name suggests – a dynamical system which is governed by laws of science and whose process values can be influenced by the correct input of energy to the system – the steam engine can be seen as a predecessor to the steam turbine, working in all thermal power plants around the world to convert heat to mechanical (and then electrical) power.

Control is the action of applying calculated input over time to achieve the desired output in optimal time, with resistance to disturbances or in terms of materials' cost – the generator coupled to the steam turbine's shaft has to maintain output power and angular velocity with respect to the grid, while providing emergency reserves for grid fluctuations. Each controller is tuned with regards to those requirements and the system itself, commonly known as a plant, with many possible control strategies depending on the order of performance criteria. Historically control has been carried manually by an operator, or automatically by physical mechanism, analogue electronics, and nowadays complex digital electronics.

## 1.2 Programable Logic Controllers (PLC), other controllers and periphery

Once a mathematical control algorithm is devised, a device is needed to implement it. The most common choice nowadays is a PLC, a digital microprocessor that is programmed on a relay logic language like Ladder Logic or the more code-like Structured Text using specialized software. PLCs periodically execute their code, acting in a discrete manner.

An alternative to professional PLCs is regular microcontrollers, supported by the contemporary Commercial off-the-shelf movement that calls to use widely available components to reduce costs. They are built basically with the same components but differ in customization possibilities and can be programmed on different levels, unlocking the full capabilities of the hardware. Such systems are welcome in hobbyist settings or various non-demanding projects but are generally frowned upon when high security, safety or reliability are required. This approach has been improved by the introduction of industrial Programable Automation Controllers (PAC).

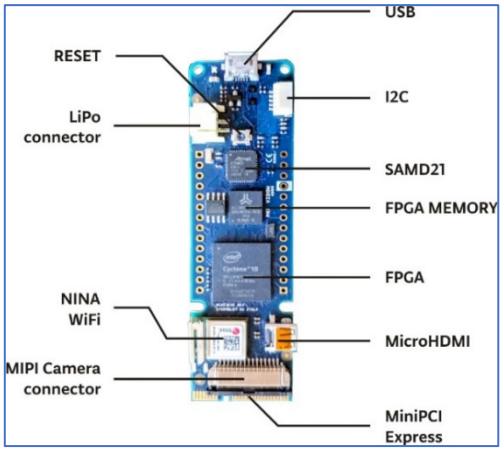



**Figure 3 Arduino microcontroller with built-in FPGA capabilities** [4]

Another option for fast-acting systems is the Field Programmable Gate Array (FPGA) – a device which "hardwires" the code to be executed down to individual logical gates on the chip itself, increasing reaction speed and reliability – qualities which come useful when protecting the transistors inside an electric motor's invertor from short circuit for example. An interesting hybrid is the hobby-grade Arduino MKR Vidor 4000 (Figure 3), which incorporates an FPGA processor in addition to its regular instruction based microprocessor, allowing for greater functionality in the high-frequency switching spectrum.

Regardless of the choice of controller, the control loop requires sensors and actuators to operate – and those come in all shapes and sizes, literally. Actuators can be powered by electricity, pneumatics or hydraulics and control power of the same or different type to the system, like an electrically operated valve controlling steam pressure. Sensors measure a physical quantity using a known physical effect to convert it to a more easily measurable value, for example a tensoresistor that allows for the measurement of microscopic changes in size through a simple electric circuit. Both sensors and actuators can be extremely simple or considerably complex (smart) with capabilities to communicate digitally, verify their operation, adjust to the environment and include built-in protection.

## 1.3 Distributed Control Systems (DCS) and Supervisory Control and Data Acquisition (SCADA) Systems

Any practical system such as an industrial plant can have hundreds of loops which are physically and operationally dependent and thus have to act in coordination with each other. This concept has been evolving over the decades as the DCS – multiple controllers on the functional Level 1 (displayed on Figure 4) exchange information in real time through equipment on Level 2 and perform in unison to meet the requirements. Such systems are often working with continuous processes and have high reliability requirements. Another notion is the automated integration of multiple, not necessarily connected, even geographically spread systems and the historical data collection that is performed by SCADA systems.

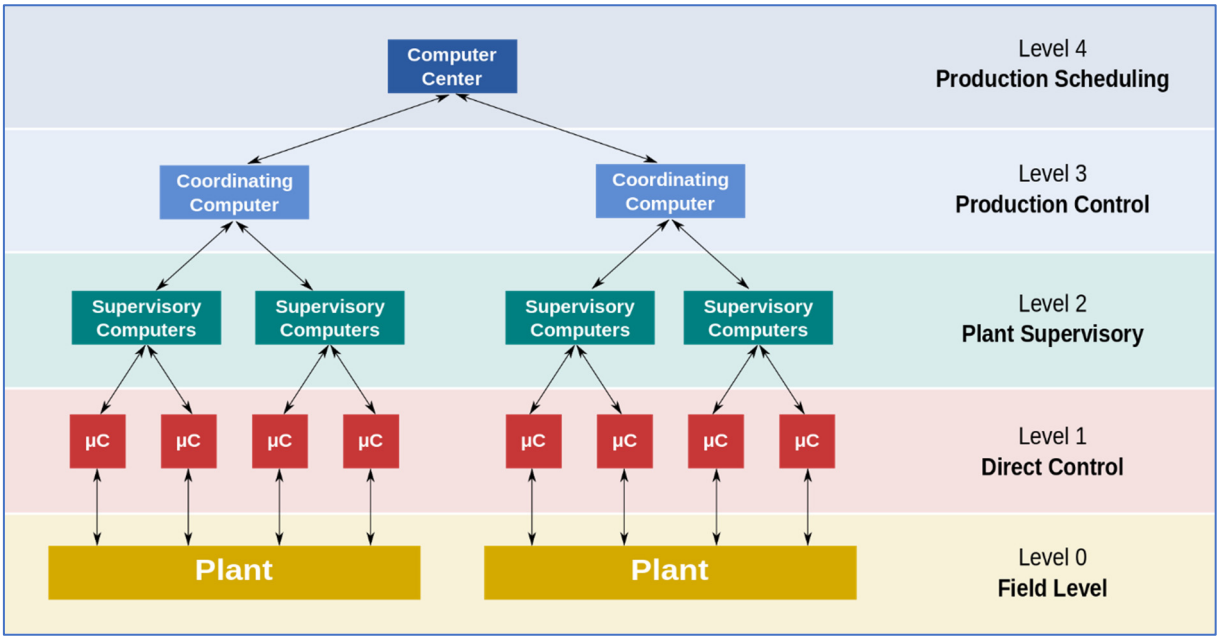



**Figure 4 Functional Levels in a Control System** [5]

## 1.4 Communication Channels

A main component in all mentioned systems is the ability to transfer information through digital channels. The original method of analogue single value transfer through a copper pair based on voltage or current is still widely recognized (an example on Figure 5) and used in the industry due to ease of use and reliability but is being shifted to lower levels due to poor scalability and information throughput. Its most common replacement is the fieldbus – a common communication medium for a set of devices, usually with a designated master device to control the flow of information.

Some notable examples are PROFIBUS, MODBUS and EtherCAT. Such protocols can be used to send data between a controller and its sensors or actuators if they have digital capabilities, or between a controller and an IO board which controls the analogue devices.

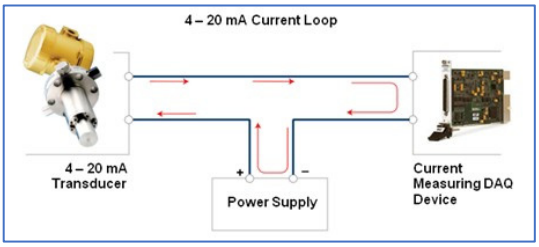



**Figure 5 An analog current loop** [6]

On a higher level, SCADA systems and DCS require connections between controllers and supervising equipment. Although this can also be achieved with the mentioned protocols, due to the relaxed

timing constraints and the star/mesh topology a switched-packet Ethernet network can be utilized. More so, such networks find usage even as a fieldbus, when the standard switch is replaced by specialized hardware with realtime guarantee.

For remote geographical access like water systems or train signalling exist various solutions: When wired telephone access is available, a dial-up connection through the public telephone network can be established. Otherwise a private radio network is required, which nowadays has evolved into the GSM(3G/4G/5G) public cell phone network with good coverage and increased bandwidth when needed. Furthermore, ICS are commonly interconnected, or at least maintenance access is provided, through the Internet.

As illustrated in Figure 6, many various communication protocols have been developed over the past 50 years. These networks are increasingly being interconnected, and as such should be considered public, due to the fact they use shared communication channels. Appropriate security measures must be taken for authentication and authorization, as will be discussed in Paragraph 1.1.

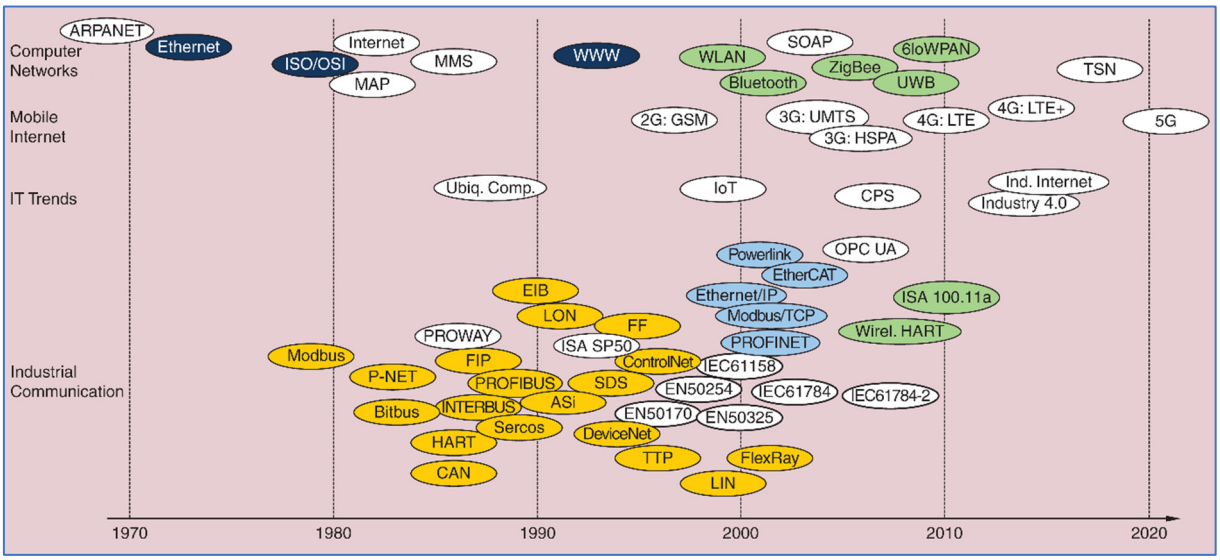



**Figure 6 Developments in Communication** [7]

# 2 Cyber Security Vulnerabilities and Requirements

Every system is designed to operate properly when its components are available and performing to specifications. Operation with present faults is an important research field which has fruited multiple methodologies to detect, predict and mitigate such events. However, none can detect any malicious actions or whether faults are caused by an external event – a relatively new topic in industry.

## 2.1 Physical Security of Equipment

The first important step towards security has been practiced virtually forever – all important components must be locked down and access to it limited. People without authorization should not be able to influence the process or interfere with the control equipment. This is most often observed as physical barriers around the plant with live security, CCTV control, ID checks, access card verification and proper record keeping.

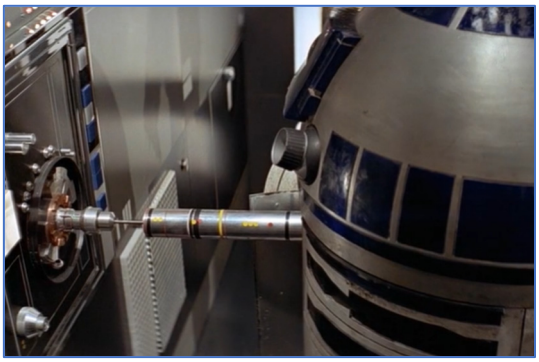


**Figure 7 A popular fictional example of bad physical port security** [8]

Any network is as secure as its weakest point – a well-known paradigm in IT security, where due to the capabilities of the hardware a single point of entry could potentially provide access to the whole site infrastructure. Measures such as port disabling and locking, authentication between devices frequent traffic scanning, as well as layered security, can be borrowed into the ICS field for network protection.

## 2.2 Outside Connectivity

Long gone are the days when a facility would be physically isolated from the outside world for security reasons – nowadays this is probably only true for Nuclear Power Plants which due to the nature of production have all required personnel and equipment available on site. At other systems the benefits have outweighed the risks and broadband connectivity has been enabled for various reasons. The classic one is the connection with the corporate network for automatic monitoring, statistics or production streamlining across business levels. Other initiatives include bridging together remote systems or vendors providing realtime maintenance and predictive diagnostics, as illustrated in Figure 8.

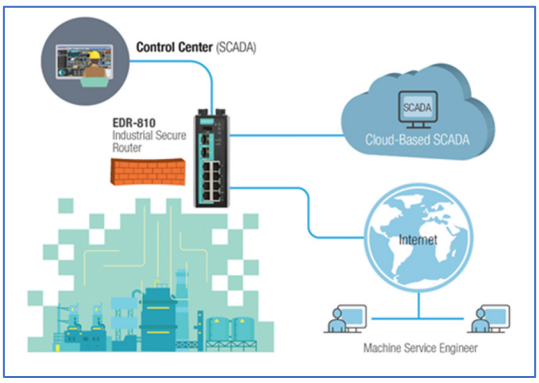



**Figure 8 The modern maintenance approach** [9]

Like physical ports, each possible connection from the outside world to a system is a vector for an attack. Measures must be taken to monitor such channels with firewalls for any unauthorized traffic and to isolate the access through such channels only to the intended target device. The network should be segmented, with a layered structure, as discussed below, recommended to shield the critical components accordingly.

## 2.3 The Layer Defence concept

Paranoid people are commonly portrayed in popular media as having a front door with an impractical number of locks, all from different vendors. While a good approach to make the door harder to lockpick, this does not protect against battering attacks or entrance through the window.

A proper security layout would include a security perimeter outside, bars on the windows, multiple doors, an active security system, a safe for valuables and even a panic room for personal protection – which represents the layered security concept.

Important assets (information or access) should be hidden behind multiple layers, each representing a different challenge to an intruder. No security is impenetrable; however, attempts can be slowed down or trigger alarms. One such idea is the honeypot trap, where unused fake assets are set up with compromised security – if the asset is accessed, it is a signal that somebody without proper knowledge is present, perhaps an intruder doing a reconnaissance attack.

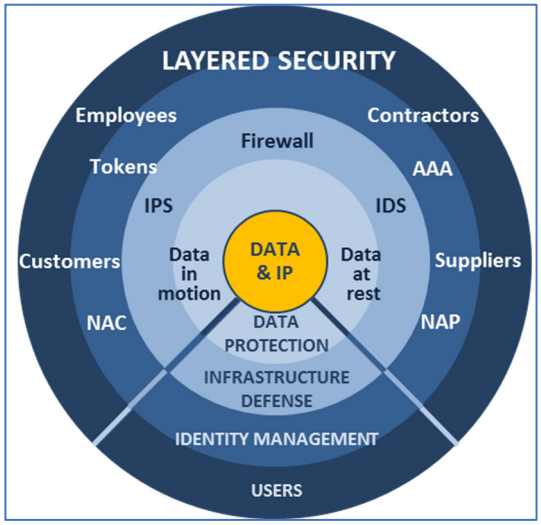



**Figure 9 The Layered Security Concept [10]**

Early knowledge of a possible attack can mean the difference between the successful protection of the system and total disaster when some time latter the full attack is carried out – with sufficient real-world examples present to highlight the dangers. System security is always a game of cat and mice, with the roles frequently reversing. Often no reaction is possible in reasonable time once the attack begins, therefore design in the key element in security.

## 2.4 Security Requirements

Overall, a system should operate with the minimum necessary access provided to resources; authorization and authentication must be present and properly carried out at each important step of an operation; any possible red flags should be logged and examined for the early signs of an intrusion. Operators and other personnel must be aware of the system and be properly trained in terms of cybersecurity. Management must have scenarios prepared for various events; for key facilities, a business continuity plan together with disaster recovery would be required.

# 3 Revised Attack Scenarios and Test Bench

The IT industry recognizes, among others, the 3 most important properties of data: Confidentiality – the notion for protecting data against unauthorized access, Integrity – the verification that the data read or received is identical to what the author created, and Availability – that the system, service or specific information can be accessed upon request within reasonable time [11].

Confidentiality is rarely sought after in control systems – most of the data transferred loses its importance after the control action is taken and carries little to no useful information outside the control loop. Exceptions of course exist when corporate espionage is considered, to obtain data regarding a rival company's status and internal doings. Should such be the case, full scale encryption is always recommended as in the IT sector, but otherwise, no actions are taken for this issue.

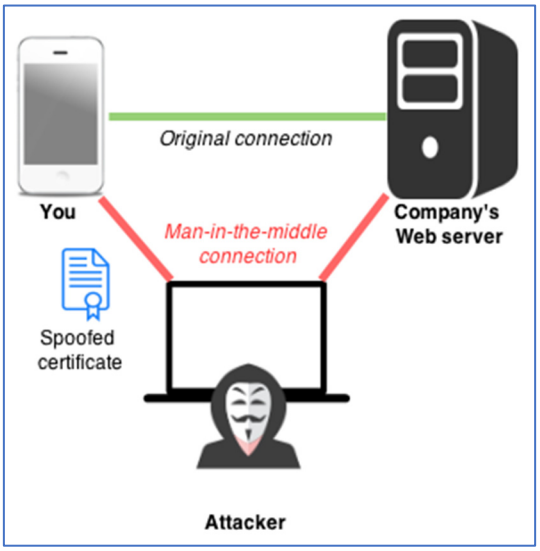



**Figure 10 A web attack on both Confidentiality and Integrity [12]**

Integrity is the main concern of control systems – due to the specific nature of control, transmitting correct values is of upmost importance as they dictate the actions performed by the system, which might be with fast dynamics, unstable in nature, or dangerous to the environment, including personnel and the general population. Attacks on integrity will further be examined in the next paragraphs, but in general any deviation from the real values is dangerous and systems are only designed against hardware malfunctions, not maleficent actions.

Availability is a relatively new concern in ICS due to the introduction of switched networks, complicated control algorithms and networked control on the lower functional levels of a plant. The system should be able to operate with various services unavailable, commonly implemented by switching to auxiliary algorithms, and critical components should be protected or designed with redundancy to guarantee performance.

## 3.1 Classic Control System attacks

Control systems are traditionally analyzed as data flowing in a closed loop, subject to plant and controller equations. As such, three types of viable attacks are considered: Reroute (Equation (3.1)), where similar channels are swapped or one duplicated over the other to hide malicious activity; Replay (Equation (3.2)) – old recorded data is sent as new for a selected period of time; Inject (Equation (3.3)) – a value is modified in transit and delivered instead of the original one.

$$\{x_1(k) \quad x_2(k)\} = \{x_2(k) \quad x_1(k)\} \tag{3.1}$$

$$x(k) = x(k-d) \tag{3.2}$$

$$x(k) = x(k) + \eta \tag{3.3}$$

Those attacks are used in numerous research papers to test performance of system validators [13], failure detectors [14] and in general system operation under fault-like conditions. The described

attacks, however, can easily be countered by introducing proper metadata management: As laid out in Table 1, each attack utilizes a specific vulnerability, which, if addressed on a higher level, would neutralize the attack, as long as the ID, timing and checksum data are protected against the tampering expected on the practical data – otherwise an attacker could modify those as well and circumvent the introduced protection measures. This implies the usage of a cipher for encryption and/or hashing – a topic for Chapter 4.

| | **Attack** | | |
|---|---|---|---|
| **Measures** | **Reroute** | **Replay** | **Inject** |
| **Channel ID** | Blocked | Allowed | Allowed |
| **Time Data** | Allowed | Blocked | Allowed |
| **Checksum** | Allowed | Allowed | Blocked |

**Table 1 Classic attacks and countermeasures**

## 3.2 Revised Attacks

As mentioned, metadata must be protected and transmitted together with the data to avoid any attempts at counterfeiting. In general, the scheme could be described as:

$$X_M = \{X, E_S(M(T, I, X))\} \tag{3.4}$$

Where X is the data, T – time, I – id of channel/author is metadata, bundled together with X by the encompassing function M and further protected by the mechanism E with possible secret key S.

A contemporary smart attack is proposed that follows modern cryptography standards: the attacker is aware of the existence of E and M, with only S unknown. Since all protection relies on S and the security of the algorithms, a successful smart attack would attempt to circumvent those measures. This concept will further be elaborated in the following chapters.

# 4 Watermarking, Steganography and Encryption

In the field of Information and especially Information Security, three major areas provide resources and tools for proper management – Watermarking, Steganography and Encryption. With different goals in mind they perform unique actions. The next chapters will elaborate on the concepts and existing methods and what follows will be an attempt to combine methods for optimal performance.

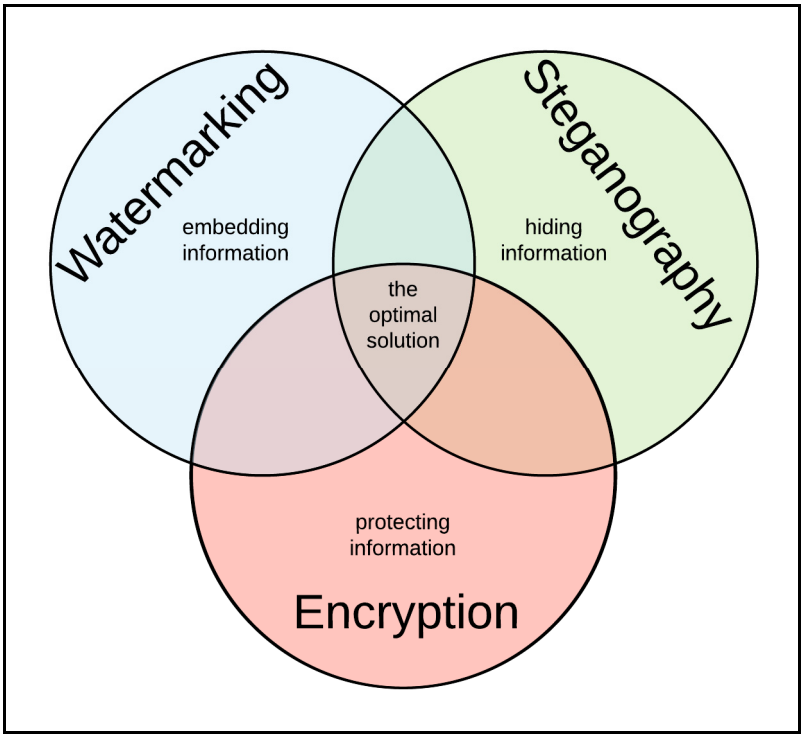



**Figure 11 The three information-related fields**

## 4.1 Watermarking

Watermarking originated to embed visible meta-information to a product – the specific example being the Italian paper mills stamping their paper for proof of origin. Today this can be observed in toilet paper – the stamping used to hold the multiple layers together are often figures unique to the brand.

Watermarking has a main purpose of conveying extra information, such as – the owner/creator/manager of the media, details about the content itself, and sometimes unique information about a license – for example somebody who purchased limited rights to the media.

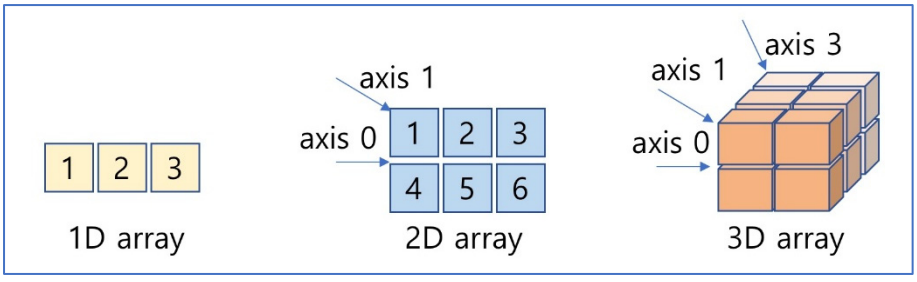



**Figure 12 Visual representation about the content in audio and images**

With the advent of digital technologies, watermarking moved from the spatial visible domain to the bit domain – utilizing space with low information importance, and the frequency domain – nearly imperceptible to the observer. Those domains are represented differently with respect to the type of

media: audio, images, video, with a quick illustration on Figure 12 – Audio is represented as a series of values, images have 2 dimensions to form the pixel plane, whereas video is a series of images, thus multiple planes like in a 3D array. Depending on the original source the encoded information can have multiple dependability between values or planes, an artifact which can freely be exploited in various watermarking schemes.

### 4.1.1 Spatial or Frequency Domain

One of the great discoveries in Signal Processing was the Fourier Transform, illustrated on Figure 13 for an imperfect square wave, which allowed the data to be "looked at" from a different perspective. In terms of information storage both domains offer capacity but differ in their properties. While audio is by nature 1D, it is important to note that 2D images also have a frequency domain, which is represented by a 2D table of magnitudes, usually through Discrete Fourier Transform or Discrete Cosine Transform. Videos, however, are rarely treated as one 3D signal, but are instead split into 2D frames for processing.

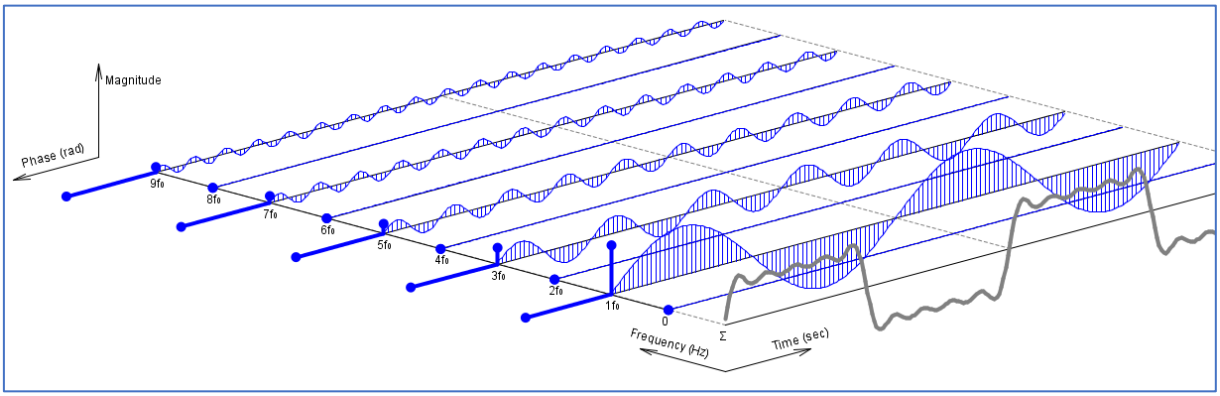



**Figure 13 The relationship between Spatial and Frequency domains** [15]

The spatial domain is represented by the bits of data each sample or pixel stores. This concept pushed further to hide the data from expert analysis will be discussed in the next subchapter, but in terms of storage information theory directs that bits have positional effect on the total magnitude in a word, with the Least Significant Bit, having, as the name suggests, least power. Therefore, information can be stored in the last n bits, allowing for great bandwidth, if they can be sacrificed without affecting the perceptible quality of the media – all sensors, visual and audio, have natural thermal noise, which could set the threshold for data storage. This storage method is however susceptible to all kinds of editing and compression – the popular MP3 and JPG standards explicitly compress those low importance areas to save on space, therefor any conversion could erase the extra data.

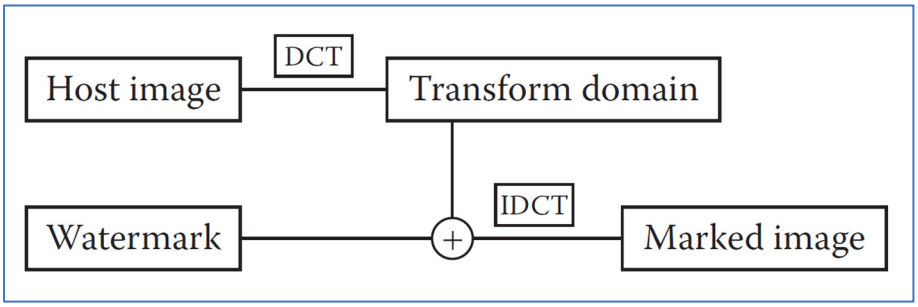



**Figure 14 Watermarking in the frequency domain using Discrete Cosine Transform [16]**

The frequency domain watermarking procedure (Figure 14) edits coefficients of the frequency magnitudes to store information similarly to the spatial, but then converts and stores the media again

in the spatial domain, where most of the possible media modifications happen. Due to the nature of the frequency-spatial relationship, coefficients are more robust to edits and thus preferred, however at the expense of less storage space.

### 4.1.2 Fragile or Robust

Watermarks are generally categorized with two main purposes with regards to media. Robust ones aim to embed additional information that must be resilient to modification in case the information is required for traceability purposes or just for convenience – a popular approach in music or video delivery to track for unauthorized distribution. Another important quality regarding robust watermarks is an overwrite protection – otherwise anybody with an access to a watermarking utility can simply replace it with its own, which creates interesting space constraints [16].

Fragile marks, on the other side, are used to provide a way to verify the integrity of the media just by analyzing the file itself. Logic dictates that they must be a compressed signature of the original media, or a hash in computer terms, which is recomputed to check for unauthorized edits to the media; a good algorithm would have the check fail even for the smallest possible modification, useful in medical or military images, where even the slightest dot (for example on an x-ray) might be of great importance. Both concepts are illustrated on Figure 15.

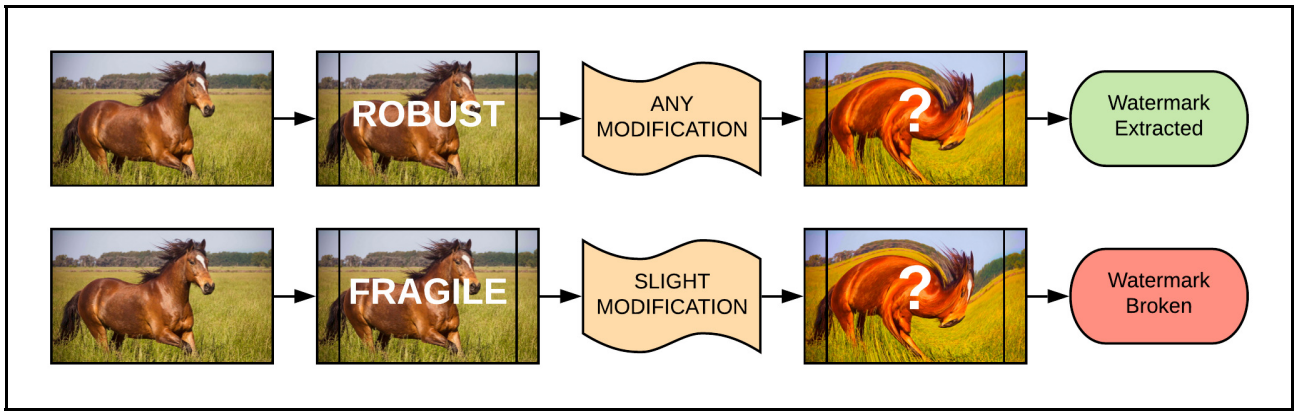



**Figure 15 The difference between robust and fragile watermarking**

### 4.1.3 Blind or Nonblind

An important distinction in watermark extraction is whether the original media is required for reference – this is highly algorithm-specific. Sometimes the verifier has no access to the original image, or the release of the original image would be inacceptable for security reasons. In other cases, the watermark might be a function of the original media, which is changed when the watermark is applied, therefore the process is irreversible, and extraction is not possible without the original.

## 4.2 Steganography

### 4.2.1 Classical Methods

The art of steganography was hiding information – passing information through a channel which normally would not permit it. This is the main difference from watermarking – the data in steganography is unrelated to the carrier media. History has found many use cases for this approach and movies about spies love it – a look at the newspaper ad section reveals nothing suspicious, but a closer look from somebody who knows what to search for delivers the information, as illustrated on

Figure 16. Further methods include hiding information in a picture's features, or even the word usage or text spacing in a paragraph.

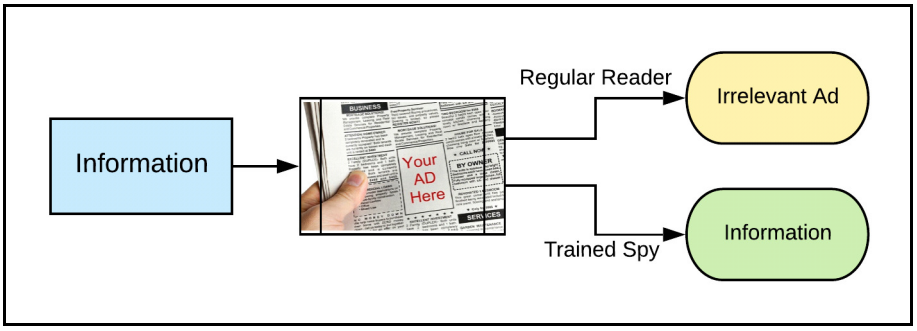



**Figure 16 The "Ad in the Newspaper" classic steganography method**

### 4.2.2 Digital Methods

Classical methods all rely on obscurity to hide the information. With the transference of methods to the digital medium, computers can comb for information through files with inhuman pace, making such methods unreliable. The previous subchapter presented multiple storage methods, which some might argue that are steganography themselves due to imperceptibility, but anybody familiar with the mechanism can detect it and extract it – violating the main rule a about undetectability of steganography.

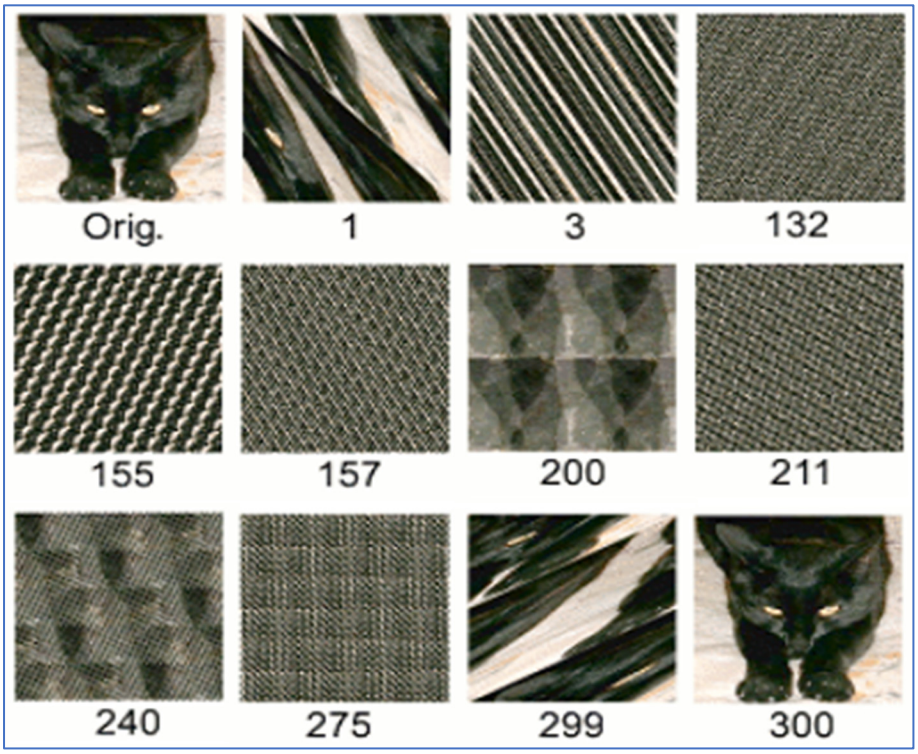



**Figure 17 Arnold's Cat Map – a cyclic chaotic map over iterations** [17]

Analysis of hidden data is mainly statistical – thermal noise from sensors is white, whereas meaningful embedded data is by definition not. Any successful digital steganography method must focus on that and other statistical properties to store data successfully without revealing its presence. One possible solution is scrambling – the usage of chaotic maps [18] to shuffle the data bits around to erase any visible statistical traces, which of course makes the same chaotic map required for

extraction as well. An example of a chaotic map is illustrated on Figure 17. This concept resembles encryption – the map being a secret key, but more on that in the next subchapter.

## 4.3 Encryption

If information has to be transmitted in an obvious way over an insecure channel, then the only available protection method is encryption – obfuscation of the data using a reversible algorithm (and if the algorithm is public, a secret key), making data unintelligible to third parties who might access it in transit, but understandable by the recipient.

Historical cyphers have used letter substitution due to the relatively low computational effort required by the human operator, but such efforts where easily thwarted by frequency analysis – the drawback of per-character encryption. With the development of technology machines were used to encryption and decryption, like the German Enigma during World War 2 – a mechanical device using electrical contacts to convert the message. The algorithm changed its internal state with every letter, thus being impervious by frequency analysis. The machine used a secret key with such complexity that no human effort could solve it within reasonable time. The development of the first computers in Britain, however, provided the computational power needed and by exploitation of weaknesses in the communication protocols the protection was eventually circumvented.

**Figure 18 The Enigma Machine, Museum for the Protection of Population in Rijswijk**

### 4.3.1 Block and Stream Cyphers

Contemporary ciphers are exclusively implemented using binary logic and computer processors. After many iterations of algorithms, several important requirements stand:

- Output Statistics: The output should be indistinguishable from a random permutation, with no apparent patterns connecting it to the input
- Input-to-Output change: The change of even one bit should ideally create an entirely different output, to protect against sensitivity attacks
- Key protection: The key should not be recoverable even when plaintext-cyphertext messages are available or chosen-plaintext attacks are executed
- Brute force: The only viable attack should be brute force, forcing the attacker to iterate through all possible values; any faster possible method indicates a flaw in the algorithm.

For the last 20 years the universal encryption standard has been National Institute for Standards and Technology's Advanced Encryption Standard (AES). It uses a series of permutations and substitutions (Figure 19), organized in subsequent rounds – the algorithm is published in its full extent, thus being tested publicly by crowdsourcing – a good empirical protection measure against flawed algorithms. It uses a 128-bit key (192 and 256 also available) and encodes data with the same size – modern computer CPUs even support special instructions just for that operation, providing enormous bandwidth, should one need it.

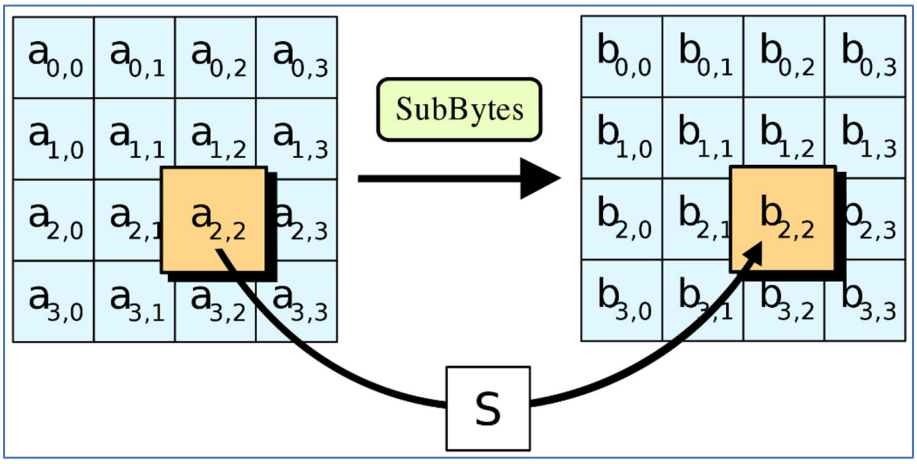



**Figure 19 A step of the AES algorithm, the substitution** [19]

Data is traditionally separated in two categories – blocks for a fixed number of chunks of data, like a file with a beginning and an end present at the time of encryption, and streams for telecommunication and other flows of data. Despite their early differences, it is accepted nowadays that both have merged, and similar algorithms can be applied – in various configuration modes.

The Electronic Code Book (ECB, left on Figure 20) mode is as simple as one could imagine: Each block of data is encrypted, independent of other blocks. Logically, this process is great for parallel computations since all computations can be carried out at the same time, however by looking at the cyphertexts one can spot identical plaintexts – and even worse, if an image is encrypted by graphical blocks, it turns out that mainly the colors are protected, but the outline of the image is still visible – illustrated on Figure 21. For this precise reason ECB is considered an insecure mode.

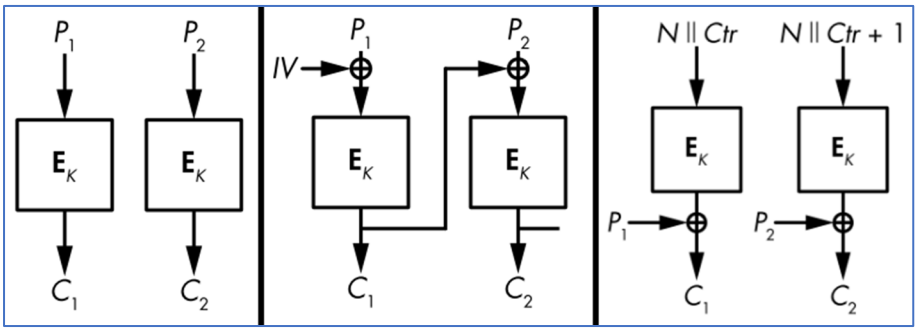



**Figure 20 Main encryption modes (left to right): ECB, CBC, CTR [20]**

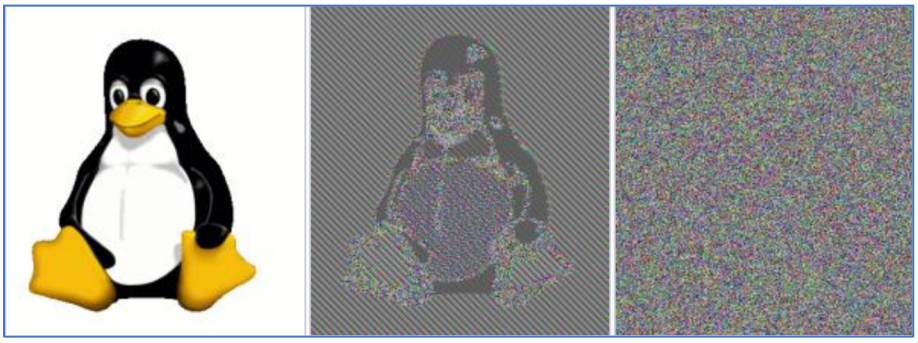


**Figure 21 The issue with ECB - "You can still see the penguin!"; (left to right) original, ECB, any other [20]**

Cipher Block Chaining (CBC, middle on Figure 20) introduces a dependability in the encryption process – each next plaintext is mixed (XOR-ed) with the previous cyphertext before encryption. By definition, the cyphertext should be indistinguishable from a random permutation, therefore this action introduces enough randomness to obfuscate any possible similarities between plaintexts. An interesting effect is that one corrupted-in-transmission cyphertext will affect up to 2 plaintexts – and not the whole decryption process. A huge drawback to this mode is the lack of parallelism in encryption –every block has to wait for the previous one to complete. This, however, is not true for decryption, which can be carried out identically to ECB.

A different approach, similar to historical stream cyphers, is the Counter (CTR, Figure 20) mode, which has the best of both previous modes – a single known nonce is combined with an incremental counter, which after encryption produces by definition codes that are indistinguishable from each other and have all qualities of a random permutation. That code is then XOR-ed with the plaintext, effectively obscuring it with the practical success of a one-time pad, if the encryption algorithm is secure. The process can be parallelized for both encryption and decryption, as long as synchronization is maintained.

### 4.3.2 Random Number Generators

Many algorithms require the generation of nonces – numbers with significant random properties – to function properly. A bad nonce might compromise a plaintext or even the encryption key.

**Figure 22 A physical source of randomness for a lottery game – a set of identical balls mixed together**

Random Number Generators (RNG) use a physical process's noise (like the airflow on Figure 22) to deliver a random number – but they are slow for the needs of cryptography. A popular approach is to use an RNG as seed for a Pseudo RNG (PRNG) – a mathematical algorithm that produces deterministic numbers with requested random properties. A good (or bad, depending on the situation) property of PRNG is that the same seed will deliver the same output – useful for synchronization, but bad if somebody is trying to crack the generator and get in sync with it.

A subset of PNRG is the Cryptographically Secure PNRG – an algorithm, that provides backward and forward secrecy for the generated numbers if part of them or the internal state is compromised – achieved through highly nonlinear functions that cannot be inverted.

### 4.3.3 Hashing

Data verification is another important field – to prove authenticity, a small footprint is created as a function of the data, used for integrity checking after prolonged storage or unreliable transmission. Hashing algorithms compress the data irreversibly to achieve a hash length of a few bytes for any file in size, like demonstrated in Figure 23.

Two important qualities of hashing algorithms are the preimage resistance – one cannot find a message that corresponds to a hash value with methods other than brute force, and collision resistance – no two messages with the same hash should be find by methods other than brute force. These qualities serve as protection of the hashing procedure, otherwise messages could be swapped and still passing a hash check. Two popular hashing algorithms are md5, widely used until recently when it was proven insecure, and the SHA family – the currently industry-accepted method.

A subset of hashing is keyed hashing – a group of methods that take a secret key and use it in the process of hashing, meaning the key is required for the subsequent verification. This addition provides authenticity on top of the integrity check on information.

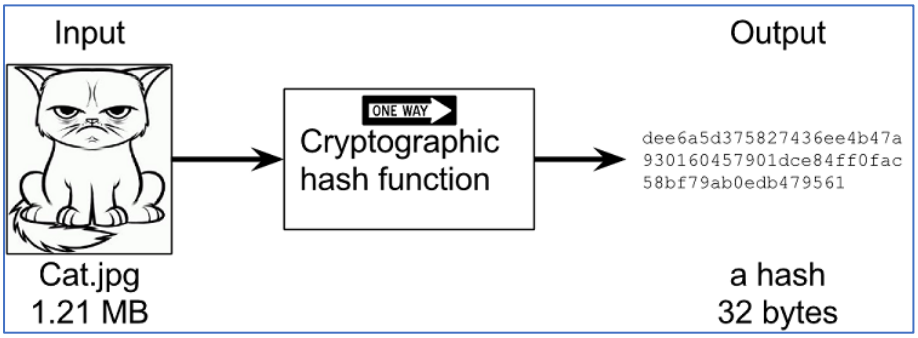



**Figure 23 An example of hashing on an arbitrary image**

# 5 The Mooren Algorithm

One existing solution for authentication is the recently published algorithm of M. Mooren [21], which proposes a linear scheme between a sender and recipient with content verification and proof of authorship through the use of fragile watermarking.

## 5.1 General description of the algorithm

As illustrated in Figure 24, the Mooren Algorithm applies a watermark on a data stream at the sender and then successfully removes it with verification about the data integrity at the receiver with the notion that the channel is insecure. The algorithm can be split into three main components as follows: (de)watermarking, new delay procedure, data validation, which will be explained in the further sections together with the algorithm initialization and assumptions.

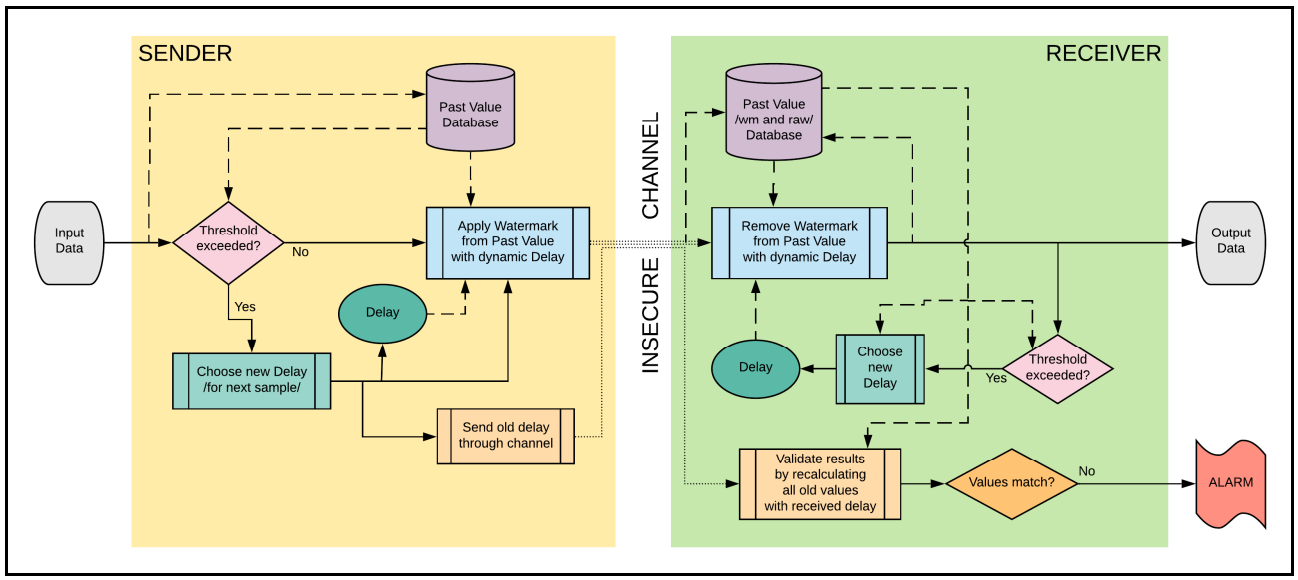



**Figure 24 A flowchart of the Mooren Algorithm**

### 5.1.1 Watermarking and Removal

The principle of echo hiding is applied here: a copy of a previous sample with lower magnitude is stored with the current one. The chosen method here is linear echo addition:

$$y_w(k) = y(k) + \alpha \cdot y(k-d) \tag{5.1}$$

That would require a buffer to store at least $d$ old values on the sender side to accommodate the watermarking function. In the equation $\alpha$ is a number less than 1 that sets the echo power – in this case it is as part of the algorithm.

The removal process is, evidently, the reverse process with subtraction. An important note here is the required possession of previously values without watermark – a notion which sounds recursive at best but will be further discussed in Section 5.1.4.

$$y(k) = y_w(k) - \alpha \cdot y(k-d) \tag{5.2}$$

### 5.1.2 Delay Change procedure

A key element in the algorithm is the delay change – a fixed delay would be of no interest since it can be estimated once and then used by the estimator's discretion. Besides the next delay value being only data-dependent, the procedure for the change is triggered by a threshold reached by the data:

$$\left|y(k) - y_{THR}\right| \overset{?}{>} \varepsilon \tag{5.3}$$

The next delay is calculated by performing a simple minimization problem based on the available data, in which a suitable delay is chosen so that the difference between subsequent values is minimal:

$$\min_{d} \left|y(k) + \alpha \cdot y(k-d) - y_w(k-1)\right| \tag{5.4}$$

This is performed mirrored at the receiver side, but for the problem to be solved the current value without watermark $y(k)$ is needed – meaning that the new delay has to be used from $y(k+1)$ onward on both sides, to allow for proper $y(k)$ extraction with the old delay on the receiver side. If the data in transfer is unaltered, the problems solved on both sides are equivalent and thus the same numerical result will be reach after each change period.

### 5.1.3 Data Validation

The described processes described so far are running in parallel but cannot highlight any differences between them. To achieve that, after every delay change, the old one is sent over the communication channel utilizing a form of digital steganography, where the last few least significant bits of the value contain the old delay or are kept null for clarification.

At the receiver side, once a sender delay is received and extracted, a recalculation is performed to confirm the linear relationship between saved watermarked data and raw data:

$$\left\| y_w - \left(y + \alpha \cdot y_{-d}\right)\right\|_1 \overset{?}{=} 0 \tag{5.5}$$

As discussed, in case the data is unaltered, the calculation should equal zero as $y_w$ is identical to $y + \alpha \cdot y_{-d}$ per Equation (5.1). If the received delay does not match the calculated one, or if any of the values is modified, but the watermark derived from it in the subsequent value is not, the equation would return a positive value which would indicate a breakage of the fragile watermark.

### 5.1.4 Initialization and Assumptions

As mentioned, the watermark removal process includes previously acquired values. A solution to this is to start the process of communication by not using a watermark in the first cycle. This would allow for proper syncing between the algorithms, which can afterwards run infinitely.

The mentioned problem highlights another issue: no packets should be lost for the algorithm to function properly, therefore this is one assumption that is required as per the author's notes.

## 5.2 Applied Simplifications

After careful analysis the variable delay period is not found to introduce any security advantages – the change can be detected by any observers by looking at the least significant bits transmission of the old delay parameters and any old messages can be retroactively recalculated by a third observing

party. By modern steganography standards this is not considered a steganographic message as its presence can easily be detected. For simplicity, the variable period can be replaced by a fixed one with desired length.

## 5.3 Analysis from security perspective

### 5.3.1 Kerckhoffs' principles

A major issue with this algorithm is the lack of conformity to the industry-acclaimed Kerckhoffs' 1st principle [22]: "Security depends more on the secrecy of the key than on the secrecy of the algorithm.". The presented algorithm offers no capabilities for secret key usage, therefor it cannot possibly offer any secrecy to two parties from any third parties that are familiar with the algorithm – in fact, any eavesdropper listening from the moment of initialization described above will be synced to the level of the receiver, therefore capable to extract the raw data and perform MITM attacks at their discretion. Furthermore, since the algorithm is not robust to dropped packets, a resyncing resetting mechanism must exist – therefore a DoS attack on a single packet would cause the system to revert to its initialization state.

### 5.3.2 The Data Validation Flaw

The idea behind the Data Validation mechanism is that first computed data is sent over a channel and later a key component used in the computation is sent over for verification, similar to sending a locked box and later the key to it after making sure that the box is in safe hands to avoid somebody else acquiring both. However the application here fails to authenticate the data itself – as long as both minimization procedures produce the same delay index (meaning the delay is dependent on only 1 value from the set, not on all of them), the validation will pass, because the same transmitted data is compared to itself, and not to something verifiably correct. This concept is elaborated via the scheme on Figure 25, where it is shown that the second assumption about data integrity is not true due to the infinite norm algorithm used and the reusage of the same received data.

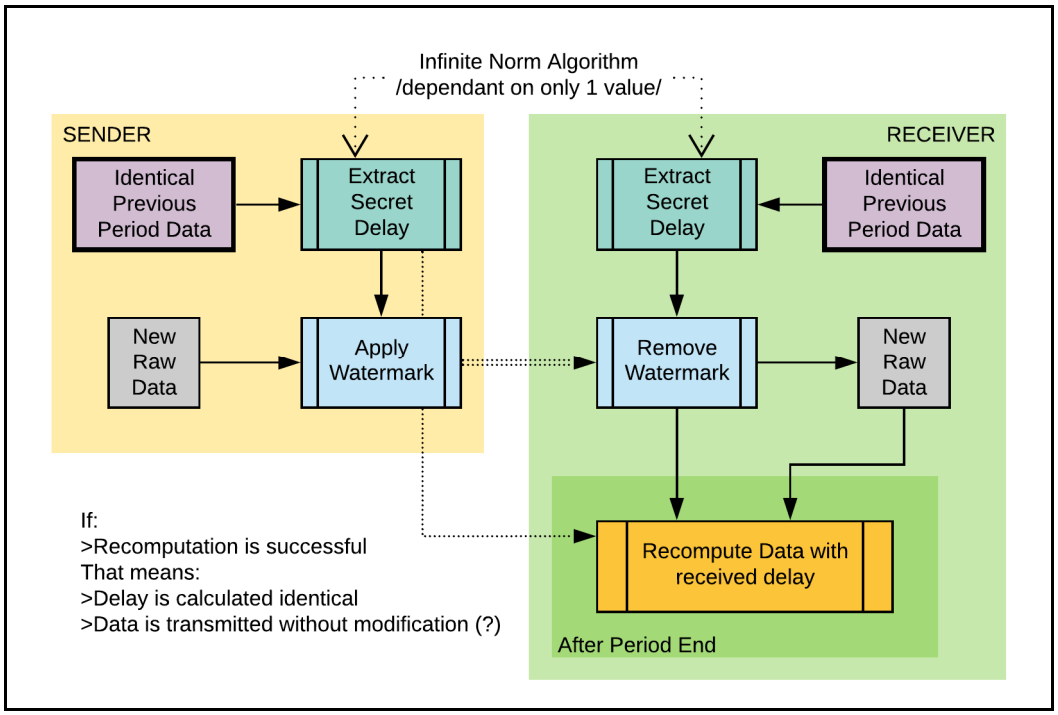



**Figure 25 The flawed assumption**

### 5.3.3 The Watermark Magnitude

Another concern would be the algorithm itself in terms of watermark magnitude. In a stream where one value remained watermarked (in this case $y_k^w$, due to lack of previous data $y_{k-1}$ to remove it), the amplitude of a single value can be traced (with the delay parameter *d* set to 1 for this example, any value for the amplitude parameter *a*):

$$\begin{aligned} Y_W &= \{y_k + \alpha \cdot y_{k-1} \quad y_{k+1} + \alpha \cdot y_k \quad y_{k+2} + \alpha \cdot y_{k+1}\} \\ X &= \{y_k^w \quad y_{k+1} + \alpha \cdot (y_k - y_k^w) \quad y_{k+2} - \alpha^2 \cdot (y_k - y_k^w)\} \end{aligned} \tag{5.6}$$

The magnitude of a watermark in a single value can therefore be expressed as a power series, where $p$ is the number of periods elapsed between the current value and the original watermark source:

$$A_W = (y - y_w) \cdot a^p \tag{5.7}$$

For a<1 the equation converges to zero. Per the author's recommendation that value is kept low to not obscure heavily the values in case they were used without removing the watermark, therefore that convergence can be theoretically exploited to estimate the states of the algorithm without interference.

### 5.3.4 An active attack on the Mooren Algorithm

Such an eavesdropping algorithm is presented in Figure 26: A passive listener collects watermarked data, waiting for a delay sync message. Once that happens, the received delay is used to try and perform a validation if enough past data is collected, identically to what a receiver would do. In this case the result is used to indicate whether the algorithm is properly synced to the sender, as opposed to the algorithm in the receiver which detects modifications. Until a sync is achieved, the data is used to attempt to calculate a new delay for next round's validation, while all received data is (poorly) cleaned from watermarks with the received true delay.

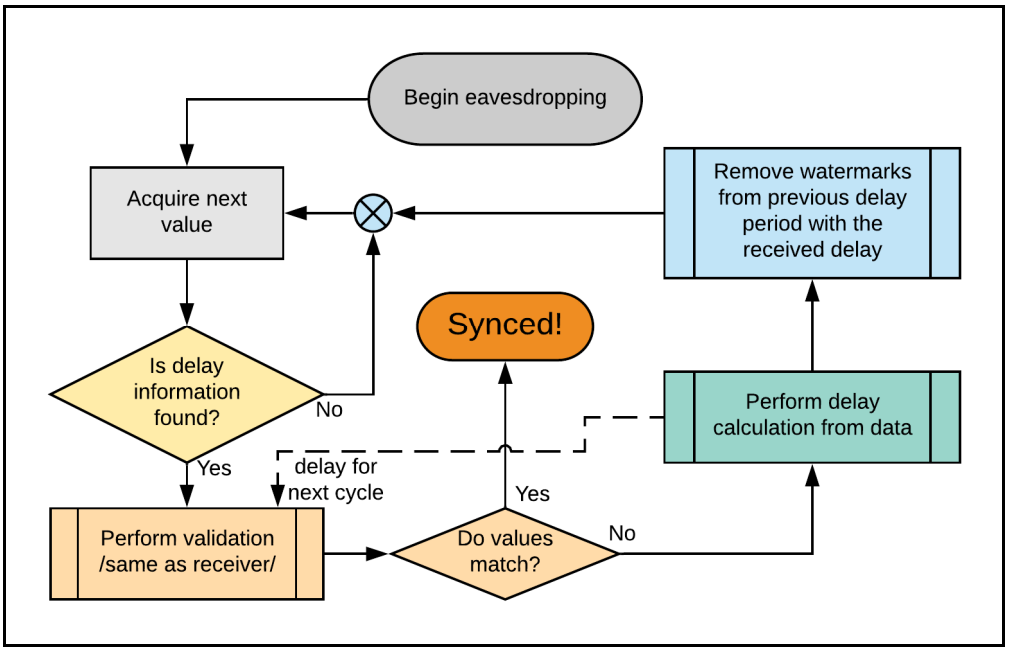



**Figure 26 Eavesdropping Sync Algorithm**

Results from successful execution of that algorithm are presented in Figure 27. The hacker has missed the first 40 samples from a communication which has the delay period set to 24, meaning the

communicating parties have synced and are transmitting watermarked data by that point (with a=0.01). The hacker performs 2 unsuccessful attempts at validation, visible from errors present and observable (to viewers) difference in calculated delays. On the 3rd attempt the error is zero, evident to the viewer also by the matching delays. After waiting one more period as a precaution, at sample 120 the attacker begins injecting data and overwriting the watermark of the sender with its own, completely hijacking the transmitted values to 1.2, which is supposedly outside the range of the sender who supports only values between 0 and 1. The delays no longer match the sender's, but the receiver cannot read the original mark and thus does not detect the attack.

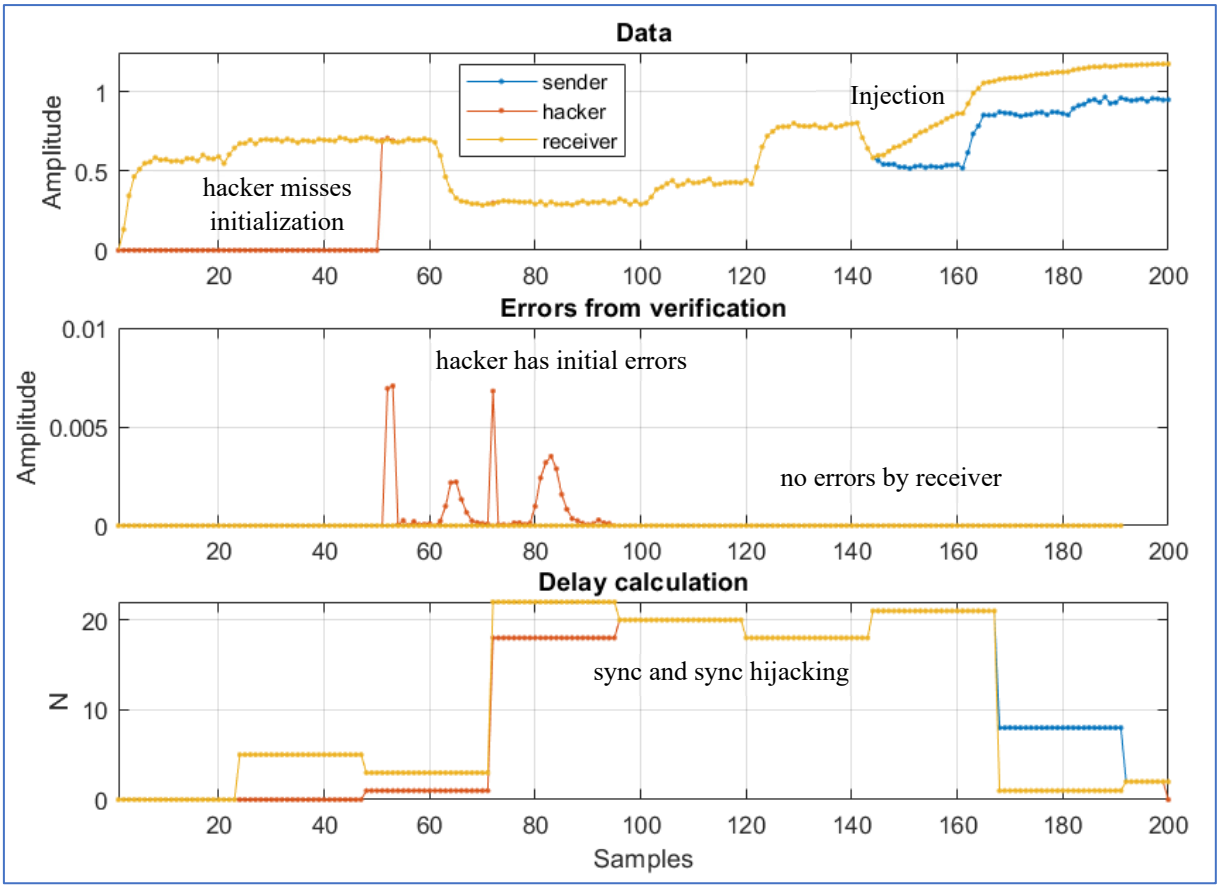



**Figure 27 Results from a successful MITM sync to the Mooren algorithm**

# 6 Evaluation of Linear Watermarking Schemes

An approach rooted in Control Theory would be to employ an additional system with "hidden" dynamics to watermark the protected signal. Benefits to that are clear – the same tools for modelling and plant monitoring can be used for watermarking, without the need of specific cryptographic methods. The data to be protected can be viewed as a series of samples and therefore a dynamic filter will be influenced by multiple samples, something desired when protecting information. Key or secret management however still has to be employed – otherwise sufficient algorithm knowledge would be enough to break the protection scheme. The main goal is to create a fragile watermark – one that can be successfully detected at the receiver side in the case of data match but broken or not detectable after any manipulation to the data in transit.

## 6.1 Additive Watermarking

One way of applying the watermark is to merge the two signals by addition – in which the watermark can be considered as noise and even hidden in the thermal noise spectrum or amplified to be of equal power to the signal. The watermarking process for a plant output $y(k)$ with watermark $w(k)$, as illustrated on Figure 28, should look in general like this:

$$y_w(k) = y(k) + w(k) \tag{6.1}$$

After transit, the watermarked signal is denoted with a bar, which indicates the possible presence of modifications $\varphi$ :

$$\bar{y}_w \in \{y_w, \quad y_w + \varphi\} \tag{6.2}$$

A validation algorithm afterwards would need to prove or deny both the existence of $w(k)$ in the transmitted $\bar{y}_W(k)$ and the integrity of the calculated $\tilde{y}(k)$ afterwards, expressed by:

$$\tilde{y}(k) = \bar{y}_w(k) - w(k) \tag{6.3}$$

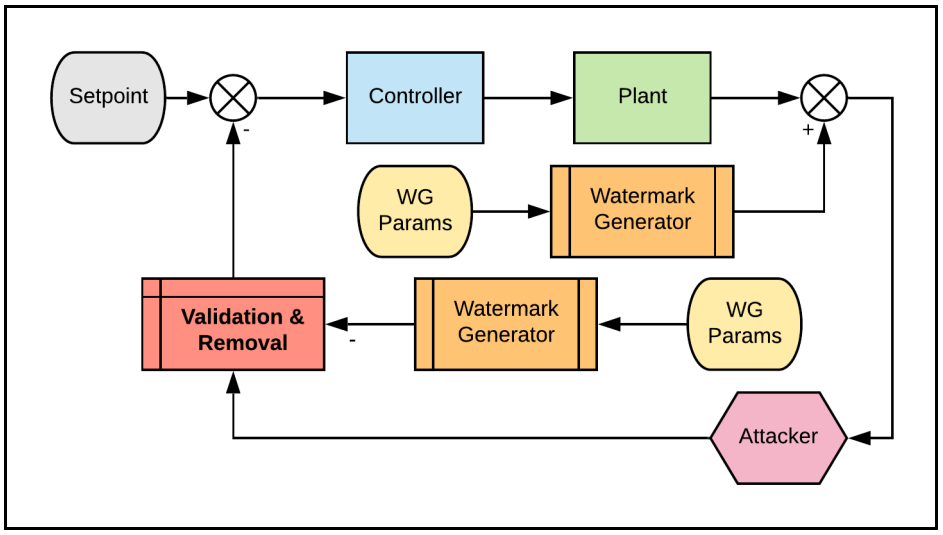



**Figure 28 Additive Watermarking Scheme**

A plant variable, in this case $y(k)$, is governed by the plant dynamics and is deterministically influenced by control actions and disturbances, including thermal noise during measurement. A model of the plant can account for the plant dynamics, however model uncertainty, noise, and

disturbances remain an issue, therefore that plant variable cannot be estimated with good certainty without the measurement output.

Information theory dictates that if you sum a random number with a known number, the result is also a random number as it has received the entropy of the initial random number, unless information is present about the statistical and deterministic properties of both numbers – the theory behind Kalman filters. This being said, a single watermarked value is a sum of a random number which needs to be verified and a known watermark signal which has to be detected. According to information theory, those two conditions cannot be met simultaneously due to the system being undetermined; that is, does not provide enough information about either the integrity of the signal and the presence of the watermark.

The other possibility is to perform validation in the time domain using previous samples:

$$\overline{y}_w(k) = \tilde{y}(k \mid k-1) + w(k) \tag{6.4}$$

The watermarked signal represents a sum of two parallel systems with known inputs and dynamics. However, it is important to note that the two systems are disjointed, meaning they do not influence each other. The exact signal of the watermark can simply be subtracted from the original system, but its actual presence cannot be proven. The remaining signal of the original plant has to be passed through a dynamics validator. This however has rendered the additive watermark useless as it does not depend on the signal nor indicates for any changes present. The problem has been reduced to the usual one:

$$\tilde{y}(k \mid k-1) \stackrel{?}{=} y(k) \tag{6.5}$$

Which is solved by using fault/attack detectors and suffers from the same vulnerabilities such as biased estimation and possible unobservable states. Due to the fact that watermark validation and data integrity verification cannot be performed, additive watermarking can therefore be considered as flawed for a watermarking scheme.

## 6.2 Multiplicative Watermarking

Additive watermarking does not indicate whether the protected value has been modified by an injection attack – since the watermark is not directly modified in the process:

$$\tilde{y}(k) = \big(y(k) + w(k) + \varphi\big) - w(k) = y(k) + \varphi \tag{6.6}$$

A different approach would be to have the watermark affect the signal multiplicatively:

$$\overline{y}(k) = y(k) \cdot w(k) + \varphi \tag{6.7}$$

The subsequent removal of the watermark is:

$$\tilde{y}(k) = w^{-1}(k) \cdot \overline{y}(k) \tag{6.8}$$

But it is important to note the possibility that $\overline{y}(k)$ is compromised:

$$\overline{y}_w(k) \in \{y_w(k), \quad y_w(k) + \varphi(k)\} \tag{6.9}$$

And therefore, the estimated signal might be compromised by:

$$\tilde{y}(k) = y(k) + w^{-1}(k) \cdot \varphi(k) \tag{6.10}$$

The presence of the watermark after extraction can influence the attack term $\varphi$, for example to diminish it or to highlight its presence. Since no information is known for the attack, the focus is to amplify the presence of a modification to allow for detection.

Dually to additive watermarking, this approach is flawed if the attack term $\varphi(k) = a \cdot y_w(k)$ is a multiplicative function of $y_w(k)$ - which would cancel out the sought after effect of $w^{-1}(k)$:

$$\tilde{y}(k) = y(k) + w^{-1}(k) \cdot \left(a \cdot w(k) \cdot y(k)\right) = \left(1 + a\right) \cdot y(k) \tag{6.11}$$

## 6.3 Combined "dye pack" watermark

To eliminate the weaknesses of the two discussed arithmetic attacks, a combination of both multiplicative and additive terms is used to design a new watermark:

$$y_w(k) = W_M \cdot \left(y(k) + W_A\right) \tag{6.12}$$

A choice made here is to perform the addition first and the multiplication second, to bind the addition term nonlinearly with the multiplication term. As such, an additive attack would result in:

$$\tilde{y}(k) = y(k) + W_M^{-1} \cdot \varphi(k) \tag{6.13}$$

And a multiplicative attack with $\varphi(k) = a \cdot y_w(k)$ would be:

$$\tilde{y}(k) = y(k) \cdot (1 + a) + W_A \cdot a \tag{6.14}$$

Both attacks leave traces of the watermark after removal and would thus "dye" any changes made to it, similarly to the dye packs put with banknote stacks when a bank robbery is taking place to explode shortly afterwards and mark all stolen cash permanently. The concept of modification marking can be seen as dual to the fragile watermark, which breaks after modification.

# 7 A dynamic watermark generator

So far various theoretical approaches for watermarking have been explored. In this chapter a combined watermark concept will be developed into a detailed model together with a validator algorithm. Various attack angles will be considered in theory to estimate the strength of the protection scheme and a modification to the scheme will be proposed to strengthen it against such attacks.

## 7.1 Overall scheme

For the multiplicative part $w(k)$ a dynamic transfer function as a filter is proposed, a concept introduced in [23], that takes the signal as input. The concept in terms of a control system is illustrated on Figure 31. The receiver has to perform watermark removal, in this case the inverse of the watermarking process, together with some validation to accept or reject the signal, as shown in the modified scheme of Figure 29.

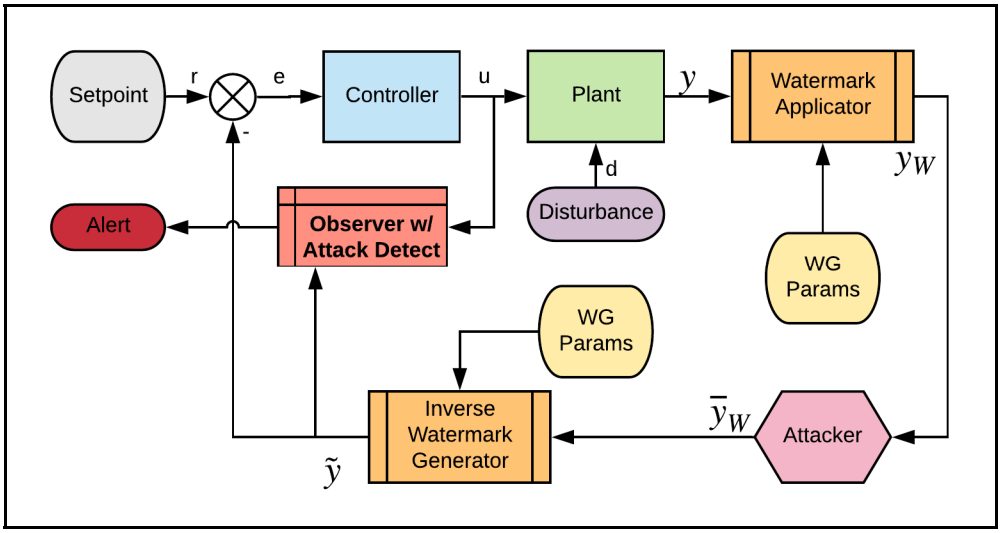



**Figure 29 Multiplicative Watermarking Scheme**

Borrowing from disturbance rejection, if we consider the multiplicative watermark as a simple gain, it might make sense to try and get rid of the attack term by minimizing it, which implies the following:

$$w^{-1}(k) \to 0, \Rightarrow w(k) \to \infty \tag{7.1}$$

Indeed a solution like that would work in basic cases where the attack is limited in amplitude, but building on section 3.2, an attacker can try to circumvent that measure by approximating the scale factor and amplifying his attack term by it. A solution with a more complicated secret "key" is needed, meaning more parameters involved in the computation behind the watermark. As mentioned, a dynamic biproper (for invertibility) transfer function as a filter is chosen for the multiplicative watermark as follows:

$$W(z) = \frac{b_0 + b_1 \cdot z^{-1} + \ldots + b_n \cdot z^{-n}}{1 + a_1 \cdot z^{-1} + \ldots + a_n \cdot z^{-n}} \tag{7.2}$$

The (2n+1) parameters are considered the secret, but the watermark generator also has (n) internal dynamic states which remain synced across sender and receiver with every transmission. An additive term as a constant, also a secret parameter, is added to the scheme to fulfil the combinative requirements. Expressed with the respective uppercase letters in discrete z-domain form notation, the

overall watermarking scheme of signal $Y(z)$ with multiplicative part $W_M(z)$ and additive part $W_A$ to calculate $Y_W(z)$ is:

$$Y_W(z) = W_M(z) \cdot Y(z) + W_A(z) \tag{7.3}$$

For simplicity while using control system notation, this watermarking function will temporarily be referred to as $Y_W(z) = W \cdot Y(z)$ and its inverse will similarly include the inverse actions to derive the original signal.

The watermarking process in control terms is represented with the plant P, controller C (internal process and measurement disturbance d not shown) and input error $E(z)=R(z)-\tilde{Y}(z)$, with notation from Figure 31:

$$Y_W(z) = W \cdot P(z) \cdot C(z) \cdot E(z) \tag{7.4}$$

The term $\tilde{Y}(z)$, part of the error signal, is calculated as follows, with a possible attack term $\vartheta(z)$:

$$\tilde{Y}(z) = W^{-1} \cdot \overline{Y}_W(z) = W^{-1} \cdot \left(Y_W(z) + \vartheta(z)\right) \tag{7.5}$$

A validator looking at a single sample of a watermarked value is constrained by the underdetermined system – but if the signal in the time domain is considered with knowledge of plant parameters, then a dynamic observer with anomaly detecting capabilities could statistically detect an ongoing attack with minimal delay.

The closed loop dynamics of the system are then derived:

$$\begin{aligned} Y(z) &= P(z) \cdot C(z) \cdot \left(R(z) - \tilde{Y}(z)\right) \\ Y(z) &= P(z) \cdot C(z) \cdot \left(R(z) - W^{-1}\left(W \cdot Y(z) + \vartheta(z)\right)\right) \\ Y(z) &= P(z) \cdot C(z) \cdot \left(R(z) - Y(z) - W^{-1}\vartheta(z)\right) \\ Y(z) &= \frac{P(z) \cdot C(z) \cdot \left(R(z) - W^{-1}\vartheta(z)\right)}{\left(1 + P(z) \cdot C(z)\right)} \end{aligned} \tag{7.6}$$

It can be concluded that if an attack is not taking place, the dynamics of the plant are as expected and not influenced by the presence of the watermarking scheme, whereas the attack term is affected by the unknown to the attacker watermark.

## 7.2 System description

The system is described by the following difference equations, starting with the controller in nominal discrete state space form, with current states $x_C$, future state denoted by $x_C^+$, inputs setpoint $r$ and plant output without watermark $\tilde{y}$, output $u$ and state space matrices $A_C, B_C, C_C, D_C$:

$$\begin{aligned} x_C^+ &= A_C \cdot x_C + B_C \cdot \left(r - \tilde{y}\right) \\ u &= C_C \cdot x_C + D_C \cdot \left(r - \tilde{y}\right) \end{aligned} \tag{7.7}$$

Similarly for the plant, whose states are also influenced by disturbance $d$ with disturbance matrix $E_D$ and output $y$ influenced by measurement noise $\eta$ with noise matrix $E_M$:

$$\begin{aligned} x_P^+ &= A_P \cdot x_P + B_P \cdot u + E_D \cdot d \\ y &= C_P \cdot x_P + D_P \cdot u + E_M \cdot \eta \end{aligned} \tag{7.8}$$

Although the plant can have multiple inputs and outputs, the watermarking mechanism will be applied to a one-dimension signal, meaning one of the outputs. The algorithm can be mirrored and deployed standalone to all of the system outputs without any cross-dependencies.

The watermarking process is described as another state space system with input addition $w_A$ and state space matrices time variant to facilitate any parameter changes due to switching decisions (omitted in following equations unless noted):

$$\begin{aligned} x_W^+ &= A_W(k) \cdot x_W + B_W(k) \cdot (y + w_A) \\ y_W &= C_W(k) \cdot x_W + D_W(k) \cdot (y + w_A) \end{aligned} \tag{7.9}$$

An attacker might influence the watermarked variable by addition $\varphi$ or multiplication with coefficient $a$:

$$\overline{y}_W = a \cdot y_W + \varphi \tag{7.10}$$

And lastly, an inverted watermark process to estimate the plant output:

$$\begin{aligned} x_{WR}^+ &= \left(A_W - B_W \cdot D_W^{-1} \cdot C_W\right) x_{WR} + B_W \cdot D_W^{-1} \cdot \overline{y}_W \\ \tilde{y}^+ &= -D_W^{-1} \cdot C_W \cdot x_{WR} + D_W^{-1} \cdot \overline{y}_W - w_A \end{aligned} \tag{7.11}$$

## 7.3 Watermark validator

Those descriptions allow for the creation of a dynamic observer, which through a properly selected threshold will act as a fault estimator, similarly to what has been done in [24]. The observer will receive the input $u$ and presumed output $\tilde{y}$ of the system, for simplicity written as $y$ in the scope of this subchapter, and calculate estimated states $\hat{x}_o$ and output $\hat{y}$:

$$\begin{aligned} \hat{x}_o^+ &= A \cdot \hat{x}_o + B \cdot u + \Lambda(y - \hat{y}) \\ \hat{y} &= C \cdot \hat{x}_o + D \cdot u \end{aligned} \tag{7.12}$$

As mentioned earlier, the described algorithm can be applied to MIMO systems without any constraints, but the notation used will account for one output $y$ for watermarking and estimating will be used, and an extension to MIMO will be presented in the later chapters.

From here a scalar residual can be computed, which is the same difference from the corrective term of (7.12) between the estimated and received outputs:

$$r = |y - \hat{y}| \tag{7.13}$$

A proper detector needs a threshold against which to test the residual and accept or reject the hypothesis that a fault/attack is present:

$$r \overset{?}{>} \overline{r} \tag{7.14}$$

The dynamic threshold $\bar{r}$ is defined by estimating the bound of the asymptotic error of the observer stemming from state *x* mismatch and maximal uncertainty of the model transfer function G, disturbances *d* and noise, in a worst case scenario when an attack or a fault is not present:

$$\bar{r}^{+} = \bar{\Lambda} \cdot \bar{r} + \left( f(G, x, d, \eta) - \hat{f}(\hat{G}, \hat{x}) \right) \tag{7.15}$$

For adequate threshold estimation, this definition is broken into separate problems for each uncertainty component: $r_x$ is state mismatch, $r_n$ for disturbances and $r_G$ for model uncertainty:

$$r = r_x + r_n + r_G \tag{7.16}$$

An observer will asymptotically follow the dynamics of the plant even when different initial conditions are present. It is common knowledge that the observer error state $e_o = x_P - \hat{x}_o$ will have the following dynamics:

$$\begin{bmatrix} x_P^{+} \\ x_P^{+} - \hat{x}_o^{+} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A - \Lambda C \end{bmatrix} \begin{bmatrix} x_P \\ x_P - \hat{x}_o \end{bmatrix} \tag{7.17}$$

The error caused by nonzero initial condition can be expressed as an autoregressive equation:

$$e_x^{+} = \bar{A} e_x \tag{7.18}$$

Where the initial state of the error $e_x(0) = x_P(0)$ participates in the output time domain equation as follows:

$$e_y(k) = C e_x(k) = C \bar{A}^k x_P(0) \tag{7.19}$$

The set of possible states $x_P(0) \in \{X_P\}$ for any system is defined and limited by operating conditions and linear dependencies. As such, a worst condition initial state can be established:

$$\bar{x}_P = \arg\max \left\| e_y(k) \right\|; \; \bar{x}_P \in \{X_P\} \tag{7.20}$$

An upper bound as a first order regression equation is developed with constant $\alpha_e$:

$$r_x^{+} = \alpha_e r_x \tag{7.21}$$

Which is parametrized by solving the following optimization problem:

$$\begin{aligned} &\min_{\alpha_e} |\alpha_e| \\ &r_x(k) \geq \left| e_y(k) \right| \end{aligned} \tag{7.22}$$

The resulting parameter $\alpha_e$, together with the results for the initial condition from (7.19), form the regression expression (7.21) for the upper bound on the residual for effects related to mismatch in initial conditions.

Disturbances are commonly present in any system, as denoted in (7.8). State and output disturbances are considered stochastic, but with known statistical properties – zero-mean gaussian noise with $\sigma$ standard deviation. An assumption is made for the disturbances to be statistically bounded – bandwidth limited white noise with a sample rate equal to that of the discrete system.

$$\begin{aligned} |d(k)| &\in \left(0, \bar{d} = 3 \cdot \sigma_d\right) \\ |\eta(k)| &\in \left(0, \bar{\eta} = 3 \cdot \sigma_\eta\right) \end{aligned} \tag{7.23}$$

Those disturbances cannot be measured or calculated beforehand and thus their effect can only be felt after they affect the system and the signals. Despite the noise being zero-mean in the long term and one might say its effect is net-zero, it is statistically possible that a number of consecutive values are similar and thus this should not be ignored. To include the disturbances in the threshold calculation their effect has to be calculated. The state-space plant output equation is split by grouping the deterministic part of the state evolution by $x_{\text{det}} = Ax + Bu$:

$$x^+ = Ax + Bu + E_d d = x_{\text{det}}^+ + E_d d \tag{7.24}$$

If we consider this to be happening for every historical value, a sequence has to be calculated:

$$x\,(k+1) = x_{\text{det}}(k+1) + \sum_{i=0}^{k}\left(A^{k-i} E_d d\right) \tag{7.25}$$

The total effect on the plant state is denoted as:

$$\Sigma_x = \sum_{i=0}^{k}\left(A^{k-i} E_d d\right) \tag{7.26}$$

This isolated state disturbance term then can be passed through the output equation with added output disturbance $\Sigma_y = E_\eta \eta$:

$$y = C\left(x_{\text{det}} + \Sigma_x\right) + \Sigma_y \tag{7.27}$$

The observer's state, however, will react to those additional terms with every time step:

$$\hat{x}_o^+ = A\hat{x}_o + Bu + \Lambda\left(y - C\hat{x}_o\right) \tag{7.28}$$

A worst-case scenario would be one where a number of consecutive disturbance samples have maximum amplitude, then change sign at a point in the past h:

$$\Sigma(k)\begin{cases} \bar{\Sigma}, k \le h \\ -\bar{\Sigma}, k > h \end{cases} \tag{7.29}$$

The observer state is evaluated at point k=h, with $\bar{A} = \left(A - \Lambda C\right)$, where due to asymptotic convergence it matches the plant state with maximum disturbance $x(k)$ from (7.25):

$$\hat{x}_o(k)\big|_{k=h} = \bar{A}\hat{x}_o + Bu + \sum_{i=0}^{h-1}\bar{A}^{h-i}\Lambda\left(C\left(x_{\text{det}} + \Sigma_x\right) + \Sigma_y\right) \tag{7.30}$$

The observer in this case will try to match the plant output value $y$ from (7.27), but will also be influenced by the output disturbance of current and past values:

$$\hat{x}_o(k)\big|_{k=h} = x_{\text{det}}(k) + \Sigma_x + \sum_{i=0}^{h}\bar{A}^{h-i}\Lambda\Sigma_y \tag{7.31}$$

With $\Sigma_o = \Sigma_x + \sum_{i=0}^{h}\bar{A}^{h-i}\Lambda\Sigma_y$, that is passed to the observer's output:

$$\hat{y}_o(k)\big|_{k=h} = C\left(x_{\det}(k) + \Sigma_o\right) \tag{7.32}$$

After the change of sign from (7.29) occurs, the plant state of (7.25) is modified ($x_{\det}(k)+\Sigma_x$ is considered a steady point until the change, therefore the multiplication of the state by A is omitted):

$$x(k+n)\big|_{k=h} = \left(x_{\det}(k) + A^n\Sigma_x\right) - \sum_{i=1}^{n}\left(A^{n-i}E_d d\right) \tag{7.33}$$

For notation purposes $\Sigma_n = \sum_{i=1}^{n}\left(A^{n-i}E_d d\right)$. And the output, with negative output disturbance, becomes:

$$y_P(k+n)\big|_{k=h} = C\left(\left(x_{\det}(k) + A^n\Sigma_x\right) - \Sigma_n\right) - \Sigma_y \tag{7.34}$$

This translates to the following changes in the observer state equation (7.31):

$$\hat{x}_o(k+n)\big|_{k=h} = \left(x_{\det}(k) + A^{n-1}\Sigma_o\right) - \sum_{i=1}^{n-1}\bar{A}^{n-1-i}\Lambda\left(C\Sigma_n + \Sigma_y\right) \tag{7.35}$$

With the observer output after the changes becoming:

$$y_o(k+n)\big|_{k=h} = C\left(x_{\det}(k) + A^{n-1}\Sigma_o\right) - C\sum_{i=1}^{n-1}\bar{A}^{n-1-i}\Lambda\left(C\Sigma_n + \Sigma_y\right) \tag{7.36}$$

Which allows for calculation of the residual, expressed as the difference between (7.34) and (7.36), only as a function of the disturbances and system matrices, and not the states:

$$r(k+n)\big|_{k=h} = C\left(A-I\right)A^{n-1}\Sigma_x - C\Sigma_n - \Sigma_y - C\sum_{i=0}^{h}\bar{A}^{h-i+n-1}\Lambda\Sigma_y + C\sum_{i=1}^{n-1}\bar{A}^{n-1-i}\Lambda\left(C\Sigma_n + \Sigma_y\right) \tag{7.37}$$

To bound this converging (since the observer is asymptotic) expression successfully, a maximum has to be evaluated over the window from the occurrence of the change to convergence:

$$r_n = \max_{n,n\in(1,\bar{n})}\left|r(k+n)\right| \tag{7.38}$$

This bound is static for the system since it does not depend on any states and has to be computed only once.

Model uncertainty is another important issue – it is possible that a plant's dynamics vary in time and a robust control algorithm has been applied, therefore a currently accurate model other than the generic one is not available. The findings of [25] are used to describe and bound model uncertainty.

In linear systems the state space equation with model uncertainty can be presented as follows:

$$\begin{aligned} x^+ &= (A + A')x + (B + B')u \\ y &= (C + C')x \end{aligned} \tag{7.39}$$

All uncertainty matrices are bounded in element magnitude by some known upper limit $\left|m_{i,j}\right| \le \bar{m}_{i,j}$.

The evolution of the state error between a plant and an observer can be expressed as follows:

$$e_x^+ = x^+ - \hat{x}_o^+ = (A - LC)e_x + A'x + B'u \tag{7.40}$$

This translates to output error as such:

$$e_y = Ce_x + C'x \tag{7.41}$$

This concept expanded back in time becomes, with $\overline{A} = A - LC$:

$$e_y(k) = C\left( \overline{A}^k e_x(0) + \sum_{i=0}^{k-1} \overline{A}^{k-i-1} \left( A'x(i) + B'u(i) \right) \right) + C'x(k) \tag{7.42}$$

From which the observer error due to unmodeled dynamics can be extracted:

$$r_G(k) = C\left( \sum_{i=0}^{k-1} \overline{A}^{k-i-1} \left( A'x(i) + B'u(i) \right) \right) + C'x(k) \tag{7.43}$$

Since the system state x is unknown, a maximum state based on the observer state $\overline{x} = \beta \left| \hat{x}_o \right|$ is assumed, where $\beta$ is a parameter calculated from a worst-case Monte-Carlo analysis on initial states with model uncertainty present. A boundary $f(u)$ is introduced by evaluating the matrices $\overline{A} = \begin{bmatrix} \overline{a}_{1,1} & \cdots \\ \cdots & \overline{a}_{i,j} \end{bmatrix}$ and $\overline{B} = \begin{bmatrix} \overline{b}_{1,1} & \cdots \\ \cdots & \overline{b}_{i,j} \end{bmatrix}$ with the upper limits established for (7.39):

$$f(u) = \max_{x \in R^x} \left( \left\| \overline{A}'x + \overline{B}'u \right\| \right) = \left\| \overline{A}'\overline{x} + \overline{B}'u \right\| \tag{7.44}$$

Equation (7.43) is modified to a scalar equation as:

$$r_G(k) = \alpha\left( \sum_{i=0}^{k-1} \delta^{k-i-1} f\left(u(i)\right) \right) + \overline{C}' \cdot \overline{x} \tag{7.45}$$

The constants $\alpha$ and $\delta$ are chosen to satisfy $\left\| C\overline{A}^k \right\| \le \alpha \cdot \delta^k \le \|C\| \cdot \left\| \overline{A}^k \right\|$, for $\alpha > 0$ and $\delta \in (0,1)$, which finalizes the expression $r_G$ for the effect of unmodeled dynamics onto the observer error.

The summation of equations (7.21) – regressive, (7.38) – constant, and (7.45) – function of the states and input, represents the worst case scenario expected residual magnitude under normal conditions and is thus suitable for threshold estimation. Any added disturbances with reasonable magnitude will cause a threshold violation and thus trigger an alarm that the watermark validation process has failed.

## 7.4 The attacker estimation problem

An attacking party that is well prepared will have knowledge of the controller C and the plant P, as they are part of the industrial project's blueprints and thus considered available. Data about the setpoint input to the controller is assumed to be known – it cannot be viewed as a secret as controlled processes often have a cycling nature or are interconnected to other processes, meaning their states can be estimated or inferred from other easily accessible information. With enough time for observation the internal states of C and P are also considered estimated to some degree, not accounting for possible disturbances d present. The known and unknown information to the attacker using notation from Figure 30 is summed up in Table 2.

Laid out like that, this protective scheme looks increasingly like an identification problem – and this is the key towards breaking it or protecting it. Since every scheme is a subject to successful parameter identification and state estimation when enough data is present, a key element will be the introduction

of switching of those parameters in a reasonable time window to avoid statistically probable estimation.

| | Known | Unknown |
|---|---|---|
| System Parameters | C, P | W |
| Internal States | of C, P - estimated | of W |
| Signals | r – exact,<br>e, u, x – estimated,<br>$y_W$ - received | d |

**Table 2 Attacker information**

### 7.4.1 Formulating the estimation problem

The key to a successful attack is carrying out actions that influence the plant in a desired by the attacker way without triggering the threshold alarm discussed earlier. The threshold is always computed for the worst-case scenario; therefore, it is expected that the actual residual will have a magnitude much lower than it. This is where an attack vector appears: If an attack term masquerading as biased noise is inserted into the feedback, it is possible to pass the attack detection test while influencing the plant. Moreover, the feedback link can be broken, and the real plant substituted by an ideal virtual one to gain full control of the attack detector state – a concept illustrated on Figure 30.

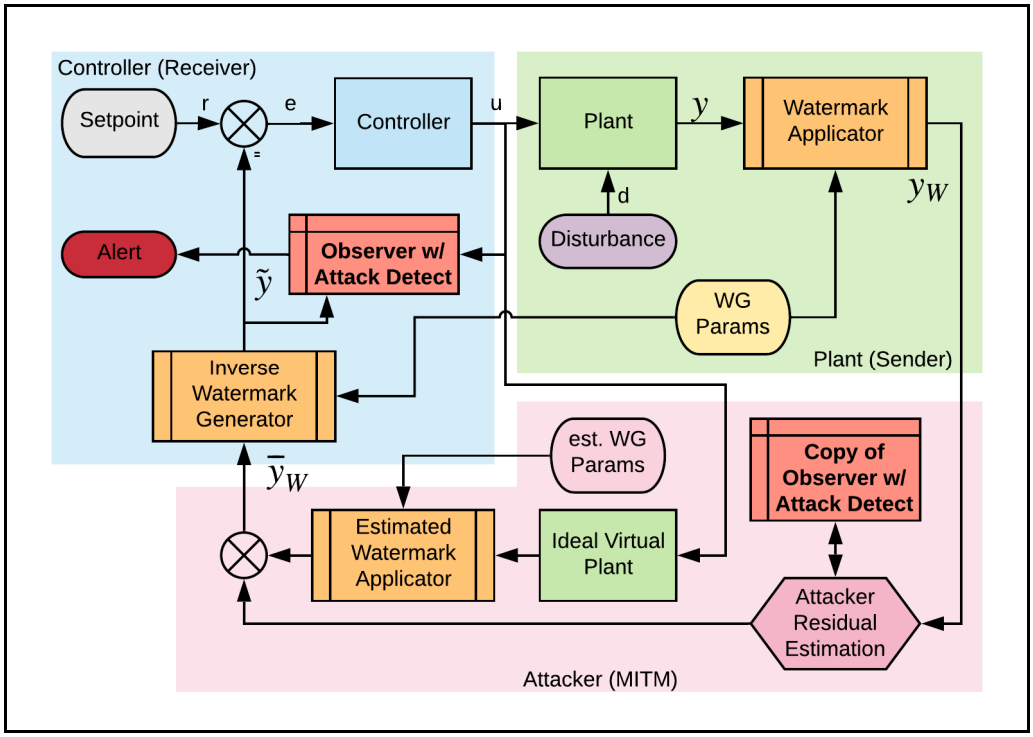



**Figure 30 Broken feedback link with a virtual plant**

For that attack to be successful, identification of the watermark parameters and estimation of the states is required. The attacker has to first perform a snooping attack, only recording the data and performing calculations. To achieve this, the system without an attack present is reorganized as shown

in Figure 31, where the watermark generator is excluded from the feedback loop and the inverse of it is no longer required. This is developed as follows from Equation (7.6) and the equation for the closed loop $Y(z) = P_{\Sigma} \cdot \left[ U^T(z)\ D^T(z) \right]^T$ with disturbance $D(z)$, further expanded into:

$$Y(z) = P_{\Sigma} \cdot \left[ \left[ C \cdot \left( R(z) - Y(z) \right) \right]^T\ D^T(z) \right]^T \tag{7.46}$$

And the watermarked output $Y_W(z)$ is:

$$Y_W(z) = W \cdot Y(z) \tag{7.47}$$

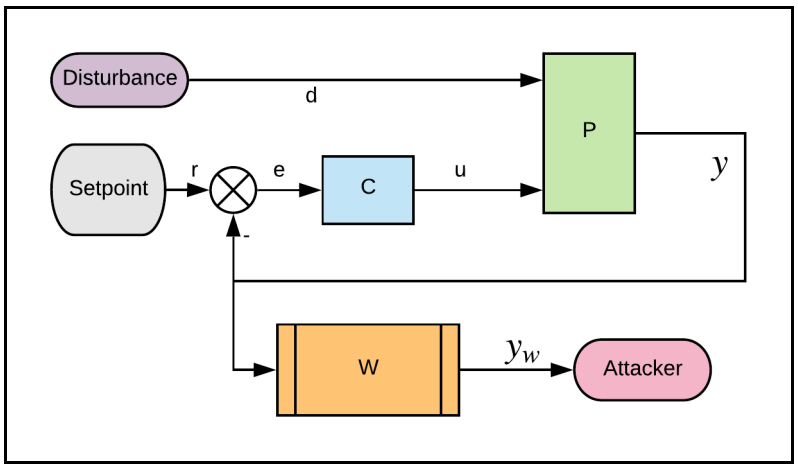



**Figure 31 Isolation of the watermark generator**

Due to the linearity of the system, the two inputs can be separated and evaluated on its own:

$$Y(z) = P_U \cdot C \cdot \left( R(z) - Y(z) \right) + P_D \cdot D(z) \tag{7.48}$$

Where $P_U$ is the transfer function from the control input to the plant output and $P_D$ is the transfer function of the disturbance to the output. This concept is illustrated on Figure 32.

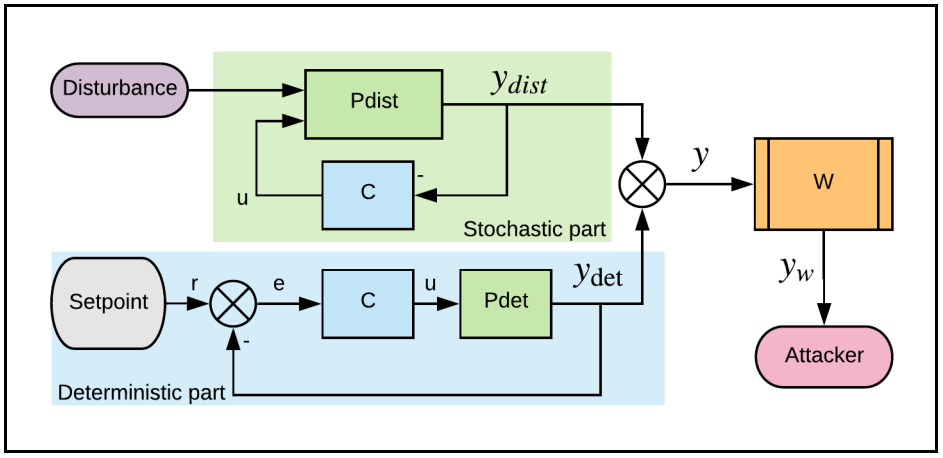



**Figure 32 Separation of deterministic and stochastic part**

The resulting two closed loops can now be evaluated separately by combining the plant models and controller:

$$Y(z) = \frac{P_U \cdot C}{1 + P_U \cdot C} \cdot R(z) + \frac{P_D}{1 + P_U \cdot C} \cdot D(z) \tag{7.49}$$

The two transfer functions are denoted respectively as $P_{UC} = \frac{P_U \cdot C}{1 + P_U \cdot C}$ and $P_{DC} = \frac{P_D}{1 + P_U \cdot C}$ for notation purposes. In the time domain, the following relationships hold:

$$\begin{cases} y_w(k) = \mathrm{Z}^{-1}\{W(z) \cdot Y(z)\} \\ y(k) = y_d(k) + y_u(k) \\ y_d(k) = \mathrm{Z}^{-1}\{P_{DC}(z) \cdot D(z)\} \\ y_u(k) = \mathrm{Z}^{-1}\{P_{UC}(z) \cdot R(z)\} \end{cases} \tag{7.50}$$

The simplified form of the resulting system description is illustrated on Figure 35. Since the setpoint of the system is considered to be known, the expression for $y_u(k)$ can be fully calculated:

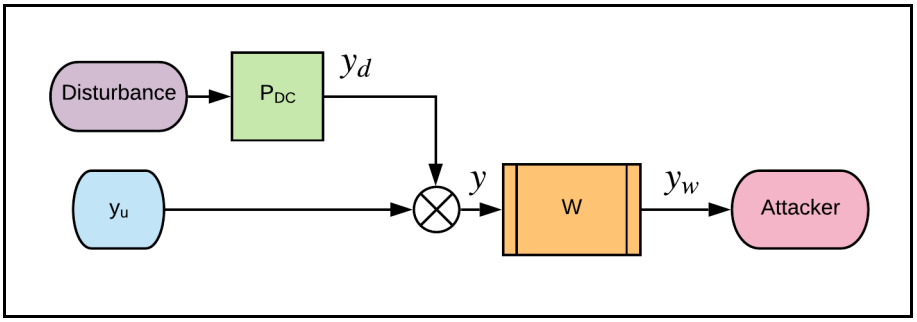



**Figure 33 Simplified form of the identification problem**

Where the final form of the problem is derived – with unknown disturbance $D(z)$ and watermark $W(z)$ parameters:

$$y_w(k) = \mathrm{Z}^{-1}\left\{W(z) \cdot \mathrm{Z}\left\{y_u(k) + \mathrm{Z}^{-1}\left\{P_{DC}(z) \cdot \mathrm{Z}\{d(k)\}\right\}\right\}\right\} \tag{7.51}$$

### 7.4.2 Solving the estimation problem

At this point, an attacker is faced with a nonlinear problem – the unknown set of disturbances $d(k)$ has a size $k$ relative to the length of the data acquired, but that set is required for the textbook input-output system identification of the watermark $W(z)$. The only known qualities of the disturbance input are its statistical properties – zero-mean noise with fixed variance. 3 different possible approaches for $W(z)$ estimation will be evaluated in the following paragraphs.

#### 7.4.2.1 Solving by simplification

Perhaps the easiest modification to the problem at hand would be to ignore the effect of the disturbance by assuming:

$$d(k) = 0,\ k \in (0, \infty) \tag{7.52}$$

This concept is illustrated on Figure 34. One argument for the successful application of that assumption would be that the term in question consists of zero mean noise, which translates to a net-zero effect on the system over time.

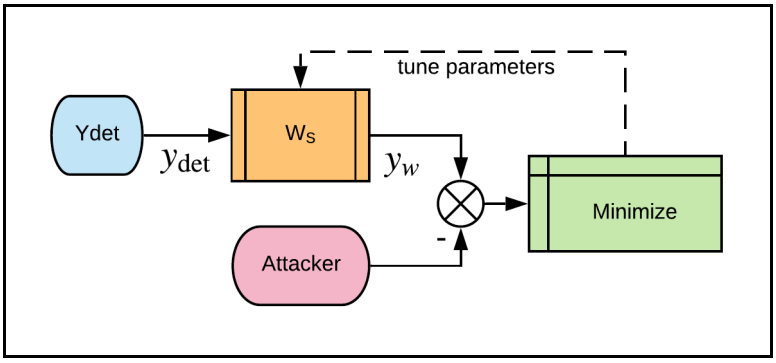



**Figure 34 Identification by Simplification**

The application of this argument can be evaluated by looking at the statistical properties of the system output. For that a stochastic input $\eta(k)$ is defined with the following statistical properties expected value and variation:

$$\begin{vmatrix} E\left[\eta(k)\right] = m_\eta \\ Var\left[\eta(k)\right] = V_\eta \end{vmatrix} \tag{7.53}$$

For any system $Y(z) = G(z) \cdot H(z)$, where $H(z) = \mathrm{Z}\{\eta(k)\}$, with state space matrices $A, B, C, D$, the output $y(k) = \mathrm{Z}^{-1}\{Y(z)\}$ will have the following expected value [26]:

$$m_y = C\left(zI - F\right)^{-1} B m_\eta \tag{7.54}$$

That expression directly connects the expected value of the input to the output via the DC gain of the system, as $C\left(zI - F\right)^{-1} B = G(z)\big|_{z=1}$. The variation of the states $V_x$ of the system $G(z)$ for $k \to \infty$ is the solution to the following algebraic equation:

$$V_x = A V_x A^T + B V_\eta B^T \tag{7.55}$$

From there, the variation of the output is:

$$V_y = C V_x C^T \tag{7.56}$$

Meaning the output variation is fixed for a chosen system and statistical properties of the input. Since the evaluated case is about white noise, $m_\eta = 0$ and therefore $m_y = 0$ regardless of the value of the system gain, and the variation will change by a constant gain depending of the system, which, if small in magnitude, could be ignored.

Another reason for such a decision would be that noise power is much smaller than signal power and thus can be ignored. That, of course, is application specific, but nevertheless, an interesting opportunity to explore.

After the simplification of (7.52), the problem is reduced to the usual system identification where numerous well-known methods such as Output-Error Model Estimation or Subspace identification can be used for parameter calculation by solving the following least squares problem:

$$E_S = \min_{W_S} \sqrt{\sum_{k=0}^{\bar{k}} \left(\mathrm{Z}^{-1}\left\{W_S(z) \cdot Y_{\mathrm{det}}(z)\right\} - y_w(k)\right)^2} \tag{7.57}$$

Since the error is minimized, the error term for the correct solution $W$ will always be lower-bounded:

$$E\big|_{W_S=W} \geq \left|Z^{-1}\left\{P_{DC}(z)\cdot D(z)\right\}\right| \tag{7.58}$$

However, the dropped term represents colored noise, and as such can bias the result. The difference between the optimal solution $W$ and the result of the minimization $W_S$ can be upper bounded to estimate the accuracy of the identification:

$$\left|Z^{-1}\left\{\left(W(z)-W_S(z)\right)\cdot Y_{\text{det}}(z)\right\}\right| \leq \left|Z^{-1}\left\{P_{DC}(z)\cdot D(z)\right\}\right| \tag{7.59}$$

#### 7.4.2.2 Solving by Frequency Shaping

Control systems are deterministic – so when fed a stochastic input, the output's statistical properties will be affected in a deterministic manner, a topic that was examined earlier. This is true in the time domain but can also be exploited in the frequency domain. A good "Meet-in-the-middle" frequency approach would be to calculate the power spectral density of the output of the plant as influenced by the disturbance and then optimize the watermark generator to produce in inverse a signal with similar properties in the frequency domain. The concept has been illustrated on Figure 35.

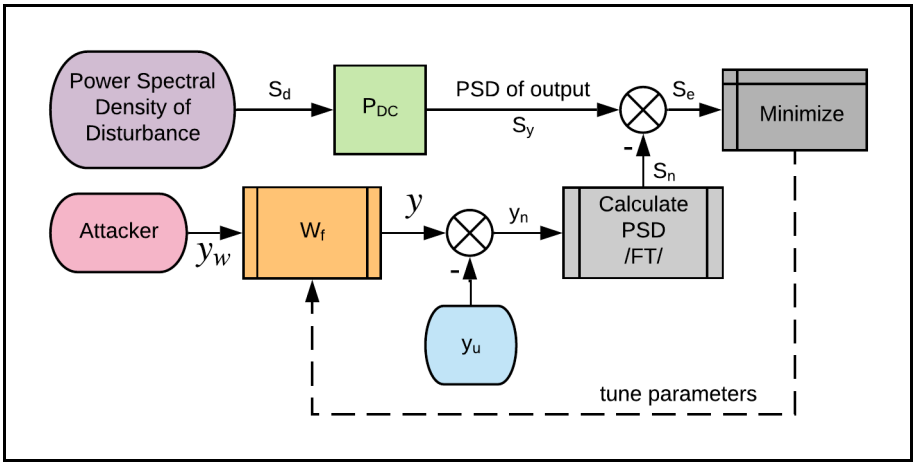



**Figure 35 Identification by Frequency Shaping**

In a system like $P_{DC}$ the relationship between the PSD $S_d(z)$ of the input $d_w(k)$ to the PSD of the output $S_y(z)$ is described as [26]:

$$S_y(k) = P_{DC}(z)\cdot S_d(z)\cdot P_{DC}(z^{-1})\Big|_{z=e^{j2\pi k}} \tag{7.60}$$

Where the expression is evaluated over the desired frequency spectrum.

The other part of the system $y_u(k)$ is evaluated with the available data in the time domain:

$$y_n(k) = Z^{-1}\left\{W_f(z)\cdot Y_W(z)\right\} - y_u(k) \tag{7.61}$$

And is then transformed to the frequency domain by a Discrete Fourier Transform:

$$X_k = \sum_{n=0}^{N-1} y_u(n)\cdot e^{-j2\pi k\frac{n}{N}} \tag{7.62}$$

Where the spectral density of the signal can be calculated:

$$S_n(k) = \left| X_k \right|^2 \tag{7.63}$$

With the expressions for the two spectral densities $S_y(k)$ and $S_n(k)$ developed, if the watermark dynamics match the true ones $W_f(z) = W(z)$, the two should be identical. This defines the optimization problem as a minimization function of the error between the two spectral densities:

$$E_f = \min_{W_f(z)} \sum_{k=0}^{N} \left| S_n(k) - S_y(k) \right| \tag{7.64}$$

### 7.4.2.3 Solving by Model Inverse

A good quality of the Output-Error Model Estimation technique is that it already assumes a disturbance present, in this case at the output. However, the presented system has a disturbance at the input, which can be circumvented if the system is evaluated in the reverse direction – meaning the plant and watermark are inverted, as presented on Figure 36.

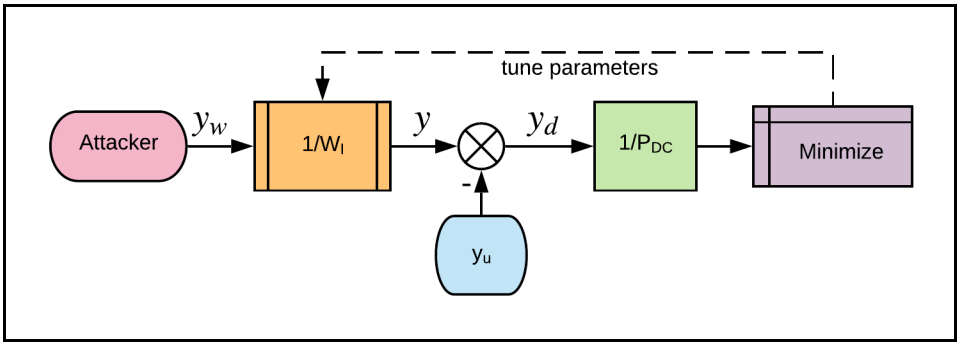



**Figure 36 Identification by Plant Inverse**

This problem is formulated as follows:

$$E_I = \min_{W_I} \sqrt{\sum_{k=0}^{\bar{k}} \left( Z^{-1} \left\{ P_{DC}^{-1}(z) \cdot \left( Y_U(z) - W_I^{-1}(z) \cdot Y_W(z) \right) \right\} \right)^2} \tag{7.65}$$

The watermark is designed to be invertible, therefore $W_I^{-1}$ is feasible and causal. However, the plant $P_{DC}$ is not guaranteed to have a causal inverted form as some models have less zeroes than poles in their dynamics. Although mathematical causal inversion might not be possible in this case, certain approximations can be derived or the causal condition circumvented [27].

It is important to observe that the plant model $P_{DC}$ can have higher dimensionality at the input, since disturbance can affect multiple states. With high-dimensionality systems a difference can be made between left and right inverses, similarly to non-square matrix inversion. In this case, a right inverse $P_{DC}^{-R}$ is suitable, to calculate input that produces a certain output, where:

$$P_{DC} P_{DC}^{-R} = I \tag{7.66}$$

Due to the configuration, the system is classified as a "fat" system with more inputs than outputs and theoretically has infinitely many solutions. To reduce this to a single solution and with knowledge that the disturbance is zero-mean, an input with minimal norm will be selected:

$$\min \left\| d_i(k) \right\|, \; d_i(k) = Z^{-1} \left\{ P_{DC}^{-R}(z) \cdot Y_D(z) \right\} \tag{7.67}$$

Due to that operation, it is not expected the calculation to return the actual disturbance values due to the possibility of multiple disturbances canceling each other out. However, the results will represent the total energy exerted over the system by disturbances, which then will serve as a measure against which to optimize the watermark parameters.

### 7.4.3 A sideline approach to existing additive watermarking schemes

Unlike the goal in this work to protect the integrity of signals, the approach of "physical" watermarking [28] includes passing watermarked data through the system and recognizing the presence of the watermark in the system output. Similarly to the other schemes, this can also be reduced to an estimation problem. The watermark $w(k)$ is applied as input disturbance to a plant $P(Z)$ with input $u(k)$ and output $y(k)$, where a feedforward system is assumed for simplicity:

$$y(k) = Z^{-1}\left(P(z)\cdot Z\left(u(k)+w(k)\right)\right) \tag{7.68}$$

The verification algorithm would look for a strong positive correlation between this expression, and a simulated response $y_0(k)$ to the input signal with virtual plant $P_0(z)$:

$$\rho = corr(y, y_0) \tag{7.69}$$

An attacker with knowledge of the plant and algorithm would need to recover the signal $w(k)$ to replicate its effects on the system during a subsequent attack. This can be reduced to the following problem, also illustrated on Figure 37:

$$W(Z) = P^{-1}(z)\left(Y(Z) - P(Z)U(Z)\right) \tag{7.70}$$

Where the major issue is the plant inversion problem, which was mentioned earlier and will be examined in practice later.

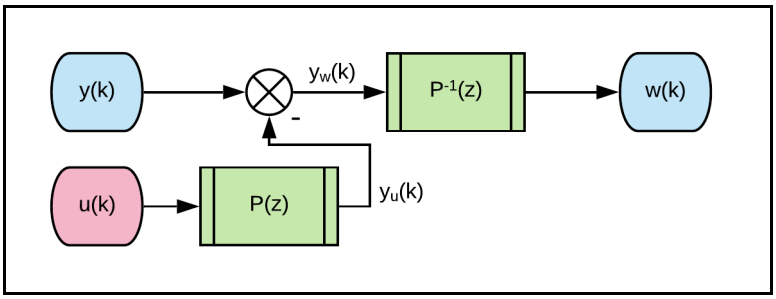



**Figure 37 The physical watermark estimation problem**

## 7.5 The need for parameter switching

The described methods and they applicability to the situation suggest that any watermarking algorithm based on dynamic systems can sooner or later be estimated, both parameters and states. Since such an occurrence would most likely lead to a compromise in the provided security, in terms of signal integrity verification, one logical fix to that is to implement parameter switching – regular change of the secret parameters, to impede the acquisition of the currently used secret, similar to the daily codes used in WW2. An implementation of this mechanic would also strengthen the resilience to replay attacks – which would otherwise be easy to perform in an industrial setting.

## 7.6 Private Parameter Generation

To create a truly strong algorithm requires secure generation of parameters – if a simple equation is used to generate them, then an attacker would be highly motivated to estimate that one as well and acquire all future secrets, which would allow him to fully spoof the plant communication regardless of the parameter change.

This is in fact the issue of key generation in the scope of cryptography: Random keys are required for the operation, but the compromise of some should not lead to the compromise of the rest. This is dealt with by using a cryptographically secure pseudorandom number generator – which, when initialized identically on both sides, will provide secure keys for operation without revealing any internal states and thus future keys.

A good algorithm for this specific application would be one based on a symmetric cipher running in counter mode, where the counter is dependent on the universal time. A hashing function is used as a one-way function to prevent any attempts at acquiring the ciphertext, since in this case the plaintext is known, despite the fact that modern ciphers can resist known-plaintext attacks. The latter is required, since the sender and receiver only communicate in one direction and therefor local synchronization is difficult, but the devices will be supplied with the time from their network. This scheme is developed on Figure 38 and has to be deployed on both the sender and receiver with the same secret key, whereas synchronization should occur instantaneously.

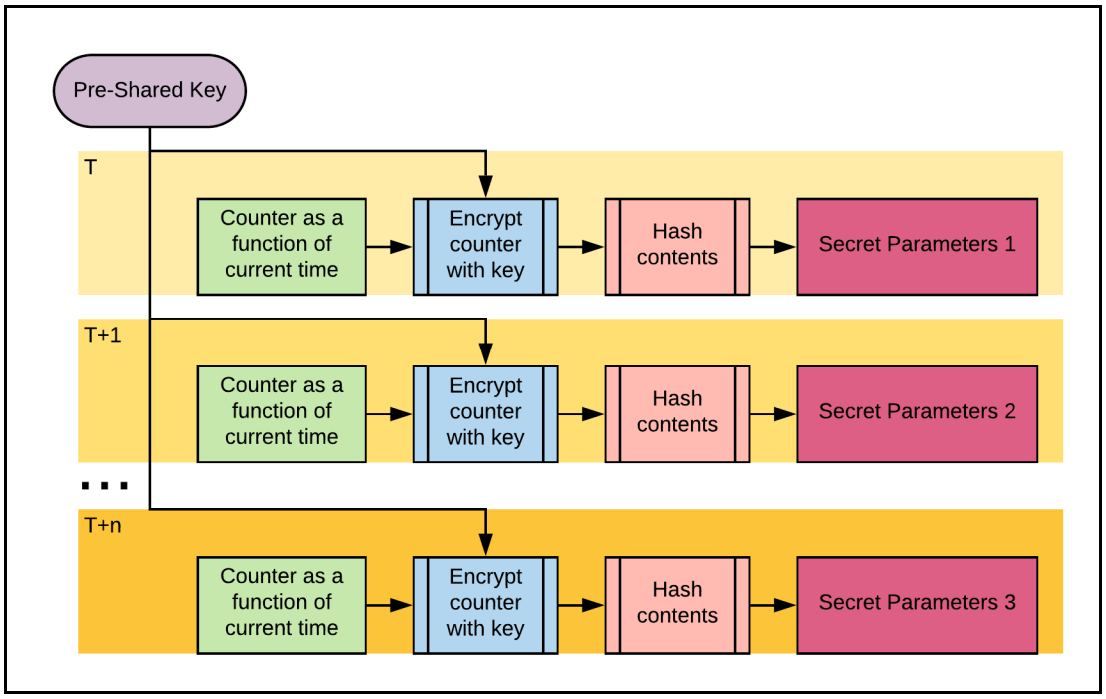



**Figure 38 A proposed secure parameter generator**

# 8 Implementation of Scheme and Possible Attacks

For proper implementation and testing of the watermarking and attack algorithms, a set of tools were developed in the MATLAB environment. The next sections will describe the plant and control parameters with the calculation of the dynamic residual threshold. Attacks will be practically implemented through the use of the described identification procedures and then respective attacks carried out. Attention is paid to the feasibility of online attacks and the effect of parameter switching to an ongoing one.

## 8.1 The Plant, Controller and Watermark

A sample 2nd order plant with the following transfer function in the continuous S domain was used:

$$P(s) = \frac{3 \cdot s + 1}{(4 \cdot s + 1)(2 \cdot s + 1)} \tag{8.1}$$

A simple feedback controller was designed to improve the performance of the plant:

$$C(s) = 0.5 \frac{4 \cdot s + 1}{s} \tag{8.2}$$

Both were converted to the discrete Z domain with discretization step of 0.1s using the Zero-Order Hold method and then represented as State Space in modal canonical form, the plant having the following parametrized model of (7.8):

$$\begin{aligned} x_P^+ &= \begin{bmatrix} 0.9753 & 0 \\ 0 & 0.9512 \end{bmatrix}_{A_P} \cdot x_P + \begin{bmatrix} 14.5 \\ -14.33 \end{bmatrix}_{B_P} \cdot u + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}_{E_D} \cdot d \\ y &= \begin{bmatrix} [0.8513 & -1.702] \cdot 10^{-3} \end{bmatrix}_{C_P} \cdot x_P + [0]_{D_P} \cdot u + [1]_{E_N} \cdot \eta \end{aligned} \tag{8.3}$$

State disturbance $d$ and output disturbance $\eta$ have been implemented as band-limited white noise. For the following simulation, the total plant disturbance measured at the output is fixed at 4% RMS compared to the setpoint response, divided equally between state and output disturbance. A random bias of up to 1% is present in the poles, zeros, and gain for the real plant, while all observers and calculations use the exact model.

The watermark is chosen as a second order biproper transfer function with complex poles and zeroes:

$$W(s) = K_W \frac{T_z^2 \cdot s^2 + 2T_z\xi_z \cdot s + 1}{T_p^2 \cdot s^2 + 2T_p\xi_p \cdot s + 1} \tag{8.4}$$

Where $K \in [0.2,\ 5]$, $T \in [0.5, 2]s$, $\xi \in [0.2, 1]$, with the 5 parameters chosen randomly from a uniform distribution. The watermark is also discretized identically to the plant. The watermark has an additive term fixed for the following simulations at $W_A = 0.5$:

$$Y_W(z) = W(z) \cdot (Y(z) + W_A(z)) \tag{8.5}$$

A sample watermark would look like this:

$$W(k)=\left|\begin{array}{l} x_W^+ = \begin{bmatrix} 0.9791 & 0.0583 \\ -0.0583 & 0.9791 \end{bmatrix}_{A_W} \cdot x_W + \begin{bmatrix} 0 \\ 0.327 \end{bmatrix}_{B_W} \cdot u \\ y = \begin{bmatrix} 0.3940 & 0.0772 \end{bmatrix}_{C_W} \cdot x_W + \begin{bmatrix} 0.2834 \end{bmatrix}_{D_W} \cdot u \end{array}\right. \tag{8.6}$$

The feedback loop is driven by a setpoint changing at a fixed period of 10s to random but known values with uniform distribution in the range $r(k) \in [-1,1]$.

A Luenberger observer is set up by pole placement, as described in (7.12), to provide Attack Detection capabilities. The observer gain used to guarantee asymptotic error convergence to zero is as follows:

$$\Lambda = \begin{bmatrix} 7.6996 \\ 3.3985 \end{bmatrix} \cdot 10^3 \tag{8.7}$$

## 8.2 The residual threshold calculation

To calculate the residual threshold for the Attack Detector the algorithms from Subchapter 7.3 are developed as methods in MATLAB functions. Their dependencies, as well as online/offline computation are described on Figure 39.

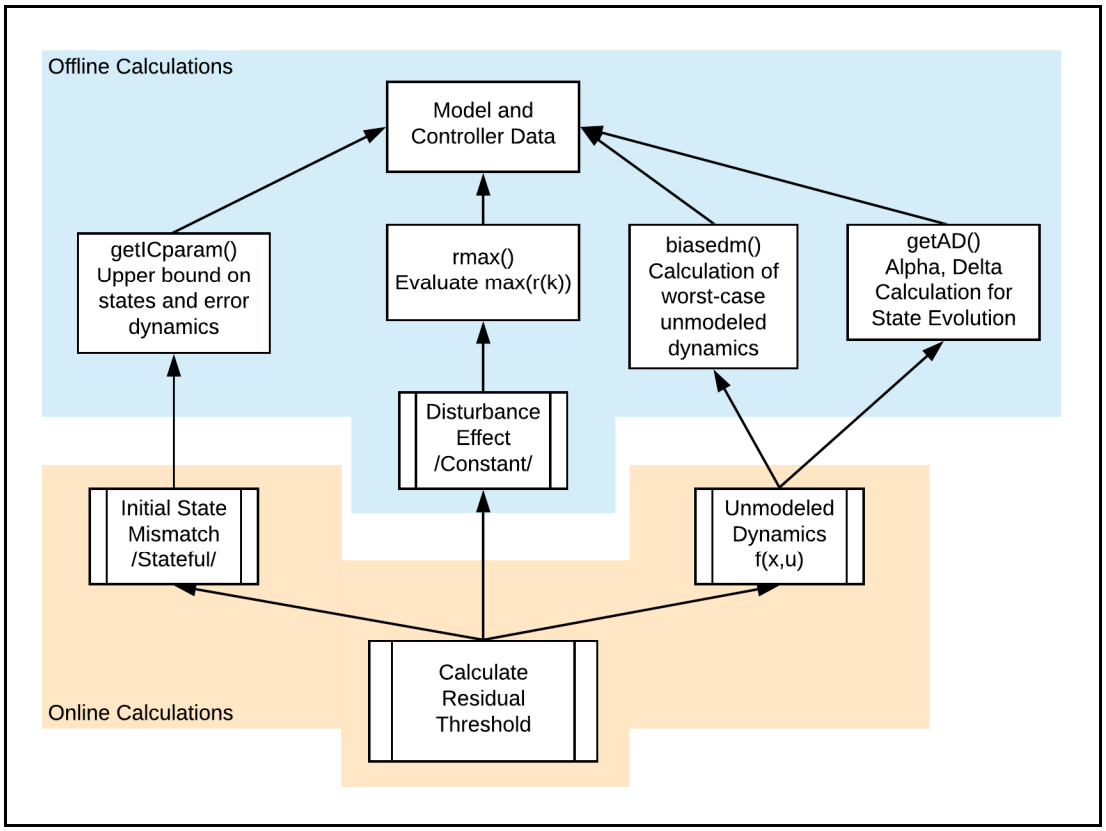



**Figure 39 Function dependencies developed for the residual calculation**

As it can be seen, the computationally heavy methods are performed offline and operation of the Attack Detector is possible in real time. The threshold has a constant component but is also dependent on the current states and input. In the beginning the threshold is exceptionally large to allow for the observer to converge to the expected states.

The initial condition regression expression from (7.21) is parametrized with the evaluation illustrated on Figure 40, where the family of transient effects is upper bounded by a 1st order equation with the respective initial condition.

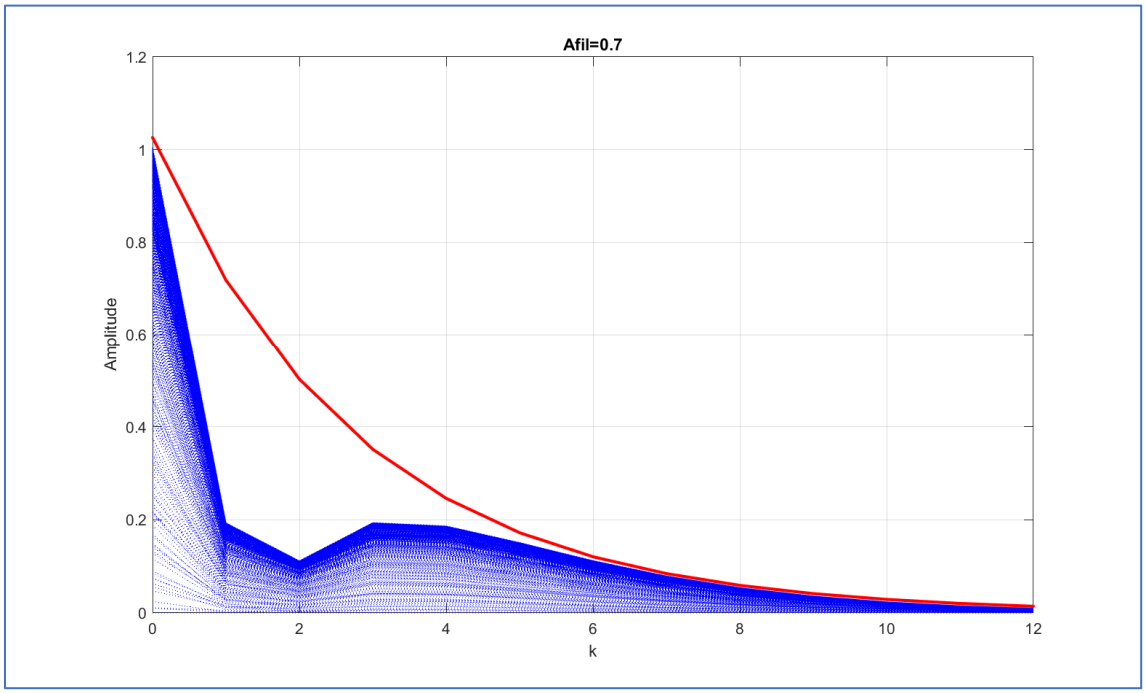



**Figure 40 Upper bound on Initial Conditions for Observer**

Equation (7.37) is evaluated on Figure 41 for the first 10 time steps and the maximum of the absolute value is selected, which is in this case the first sample. It has to be noted however, that this is not always the case depending on the state dynamics and disturbance amplitude.

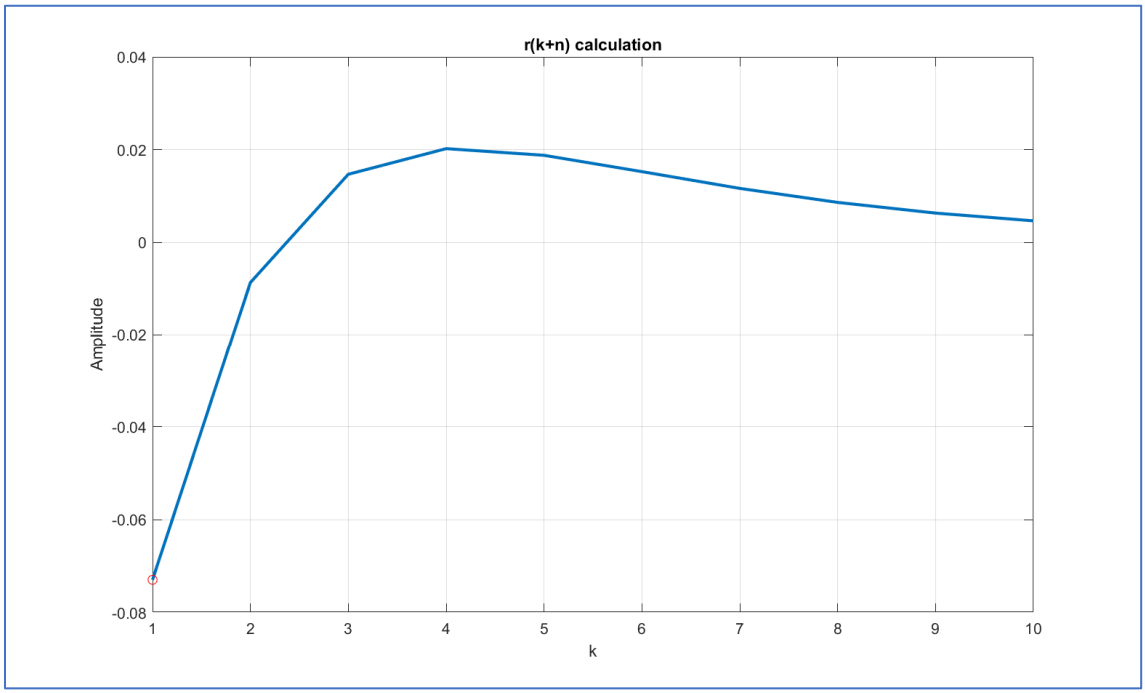



**Figure 41 Rmax evaluation**

The state mismatch $\overline{x} = \beta |\hat{x}_o|$ for (7.44) is evaluated over a random set of systems with uncertainty and the results summarized in Figure 42, where $\beta$ is calculated to be 1.32 for the current parameters.

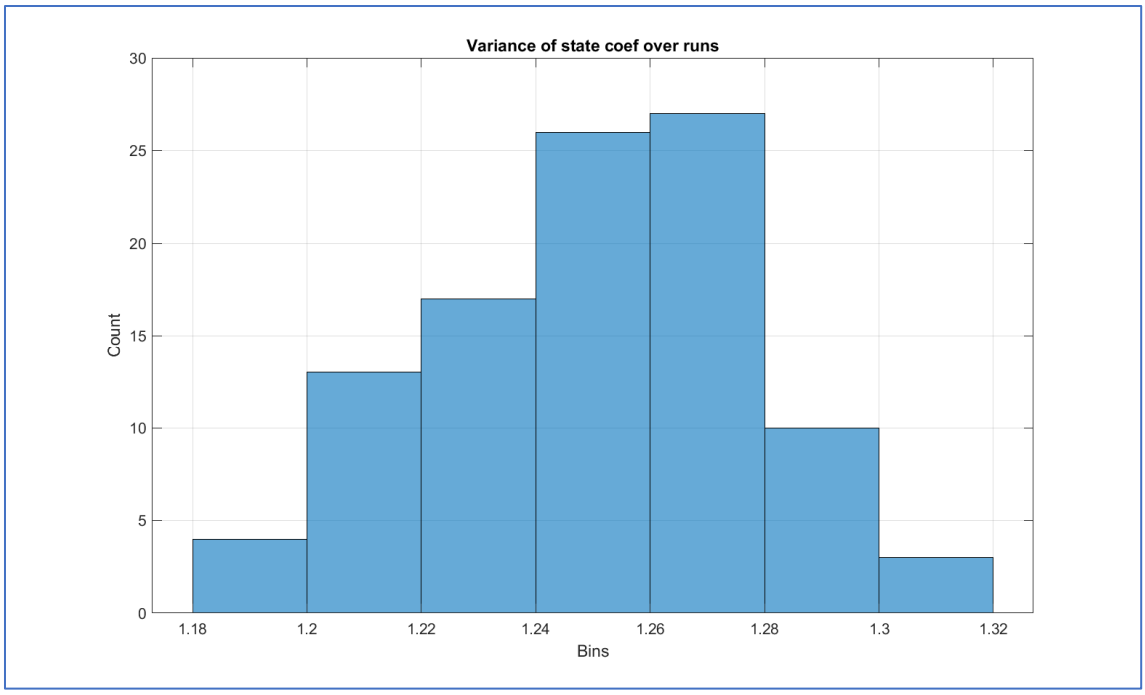



**Figure 42 Upper bound on state mismatch $\beta$**

$\alpha$ and $\delta$ for Equation (7.45) are parametrized according to the set upper and lower bound over a region of acceptable inputs. For illustrating purposes the upper and lower bounds can be seen on Figure 43 in red and blue respectively, whereas the successfully parametrized equation is plotted in yellow between the bounds for the region of interest of *k*.

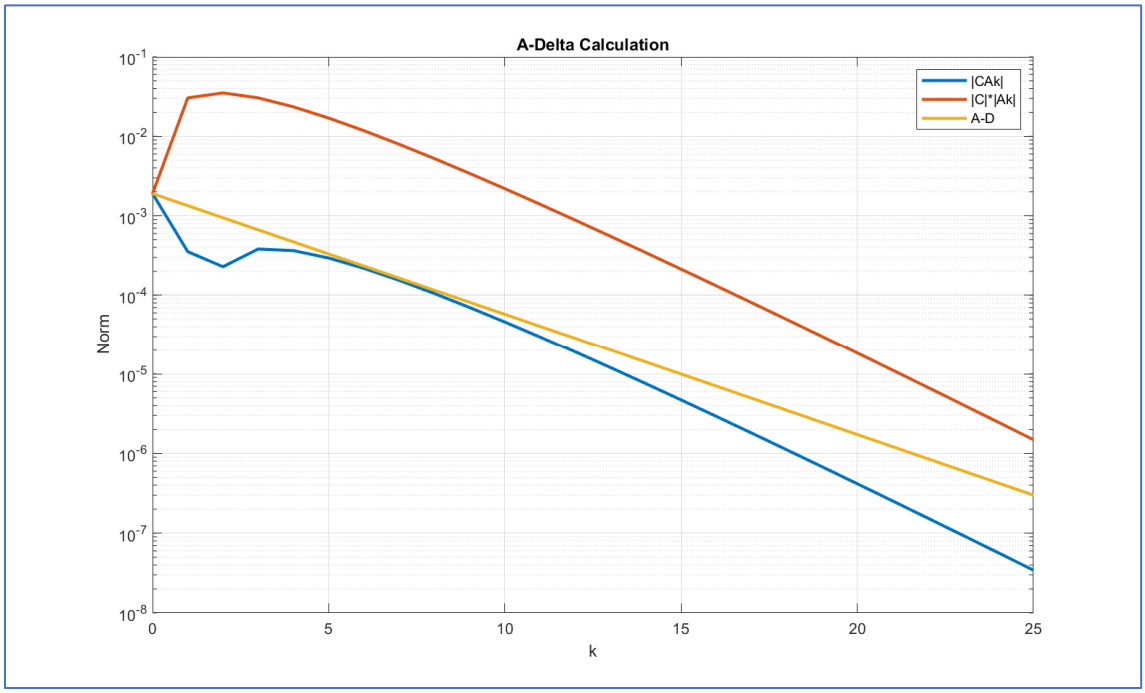



**Figure 43 Alpha-Delta calculation**

State matrices mismatch relative to the unbiased matrices is described as follows for 1% variation in the original equation's parameters:

$$\begin{bmatrix} \bar{A} & \bar{B} \\ \bar{C} & \bar{D} \end{bmatrix} = \begin{bmatrix} 0.05 & 0 & 5.7 \\ 0 & 0.10 & 5.7 \\ 11 & 14 & 0 \end{bmatrix} \% \cdot \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{8.8}$$

A sample simulation of the system with duration of 100s is presented on Figure 44 with observer running from the 5th second and its residual compared to the calculated dynamic threshold:

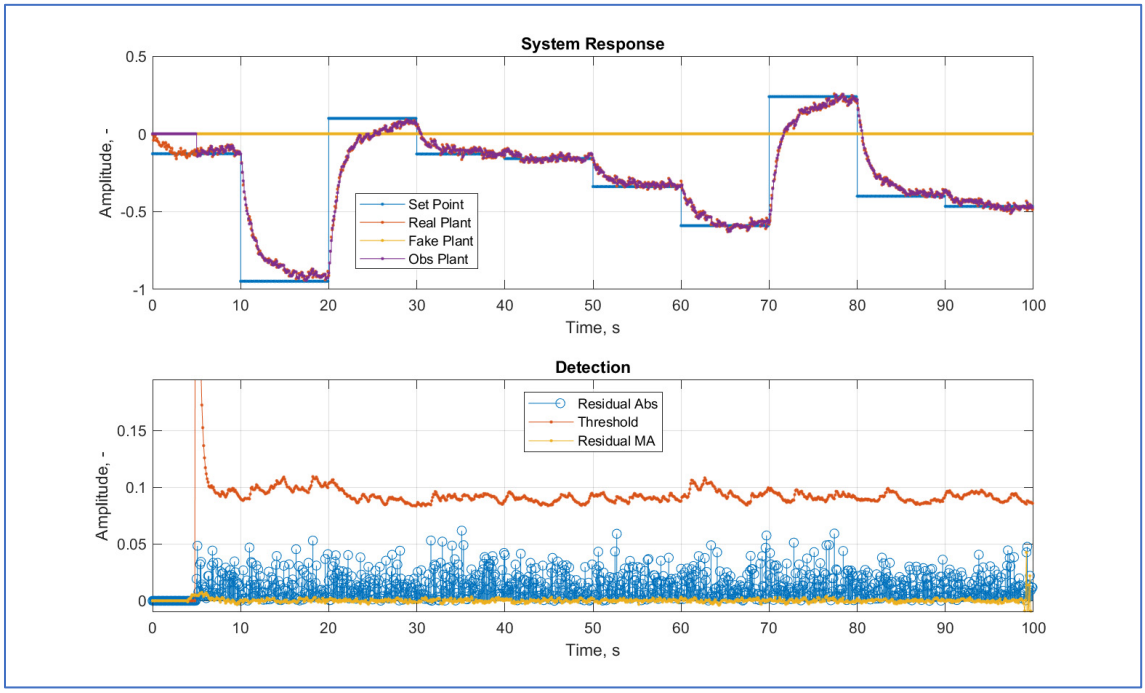



**Figure 44 Simulation with observer and residual**

The plant is being successfully actuated according to the setpoint, as designed, and the observer serving as an Attack Detector is following it without the residual crossing the threshold at any point. The fake plant noted in the top graph is not used in this scenario as it is part of the attack mechanism.

The following three sections of offline identification will each use 1000s of recorded data with 20% RMS noise present and 1% model bias.

## 8.3 Identification results on the Simplification Method

An identification attempt when ignoring any disturbances on the system is performed. The key to successful watermark estimation lies in correct estimation of the addition component, described in (8.5), which is not part of the watermark dynamics. In a system this would manifest as a constant disturbance, or in other terms an additional uncontrollable mode to the system at $z = 1$.

To achieve that identification and estimation Subspace Identification is applied, as other simpler algorithms would fail due to the presence of either uncontrollability or disturbances. Once a model is parametrized, a backwards system observer is used to estimate the initial state of the system.

The numeric results of one such identification are as follows:

$$\tilde{W}(k)=\begin{cases} x_W^+ = \underbrace{\begin{bmatrix} 0.9774 & 0.0597 & 0 \\ -0.0597 & 0.9774 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\tilde{A}_W} \cdot x_W + \underbrace{\begin{bmatrix} -0.1737 \\ 0.3194 \\ 0 \end{bmatrix}}_{\tilde{B}_W} \cdot u \\ y = \underbrace{\begin{bmatrix} 0.2849 & 0.2451 & 0.4461 \end{bmatrix}}_{\tilde{C}_W} \cdot x_W + \underbrace{\begin{bmatrix} 0.2507 \end{bmatrix}}_{\tilde{D}_W} \cdot u \end{cases} \tag{8.9}$$

And the initial states can be calculated to be:

$$x_W^0 = \begin{bmatrix} -1.593 & -2.336 & 2.692 \end{bmatrix}^T \tag{8.10}$$

A quick check of the uncontrollable mode's constant effect on the system gives us $W_A = x_{W,3}^0 \cdot C_3 \cdot \left( D + C\left(I - A\right)^{-1} B \right) = 0.5068$, a good estimation of the additive term 0.5.

The dynamic results are summarized on Figure 45, where it can be seen that a slight difference is present in the frequency domain towards the high-frequency range. Nevertheless, two metrics: Variance Accounted For, where the expected and estimated noise are compared as vectors, and Error ACcounted, which measures the variance of the present error and compares it to the expected error, show results as 83.7% and 92.2% respectively. Subsequent testing reveals that this method performs well with no noise present, but worsens progressively as the error magnitude increases, due to it being unaccounted for in the model.

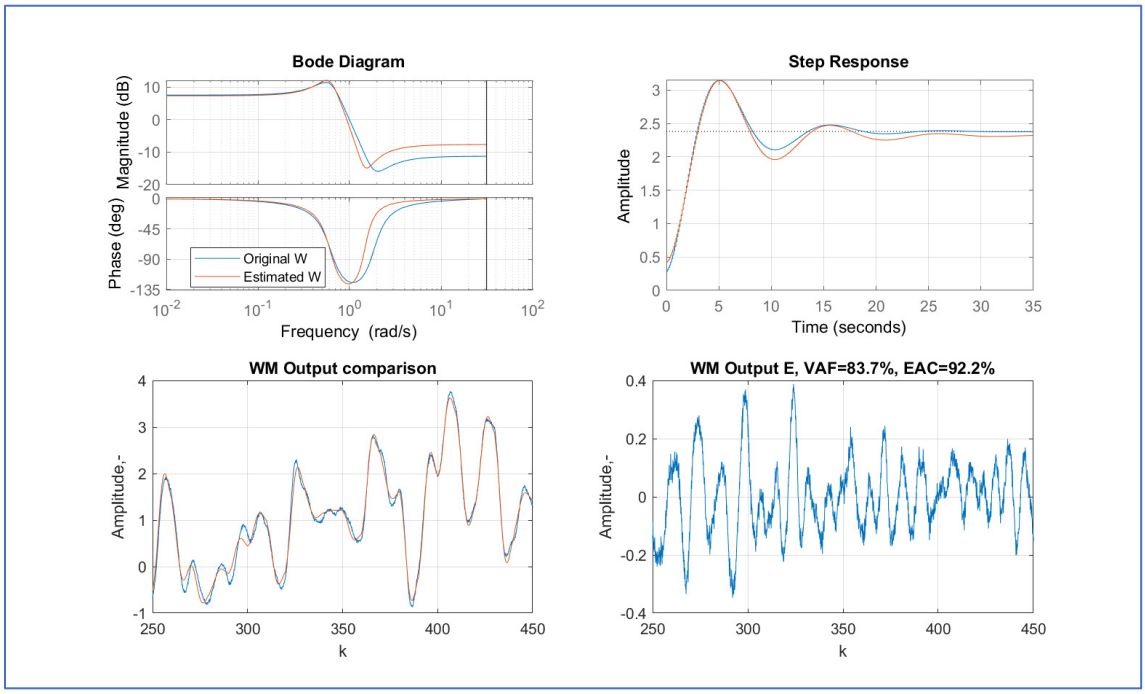



**Figure 45 Identification results from Simplification**

## 8.4 Identification using Frequency Optimization

Two approaches can be applied for estimation in the frequency domain. The more verbose one would use the input data $U(z) = Z\{U(k)\}$ from the time domain to calculate the estimated output, which is then compared in the frequency domain to the original output $S_y(k) = F\{Y(k)\}$:

$$E_f = \min_{W_f(z)} \left\| F\left\{ Z^{-1}\left\{ W_f(z) \cdot U(z) \right\} \right\} - S_y(k) \right\|_w \tag{8.11}$$

Using a weighted norm with weight $w$ could be useful in cases where the high-frequency part of the spectrum is noisy or unreliable. The results from one such optimization are displayed on Figure 46, where successful optimization has been performed with an unbiased parameter identification. The numerical accuracy is estimated as VAF=95.8% and EAC=99.5%.

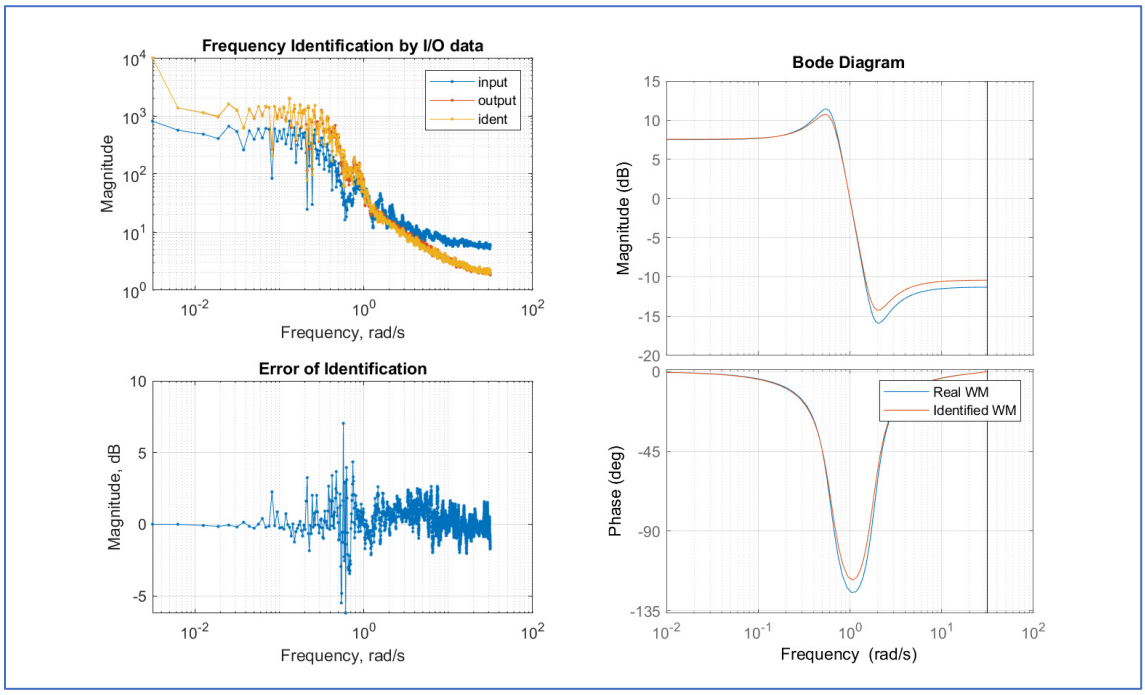



**Figure 46 Frequency Identification by IO data**

This procedure, however, is somewhat computationally expensive, as it involves multiple simulations in the time domain and then transformations to the frequency domain. Another option would be to extract the frequency profile of the system itself and then estimate the transfer function, which requires no simulations and works on a significantly reduced dataset by just evaluating a transfer function in its frequency domain:

$$E_f = \min_{W_f(z)} \left\| \frac{S_y(k)}{F\{U(k)\}} - W_f(e^{-j\frac{k}{N}\omega T_o}) \right\|_w \tag{8.12}$$

The results of this optimization are displayed in Figure 47, where it can be seen that the identification is somewhat worse than the previous, owing to the possible loss of information during transformation and the lack of high-frequency excitation in the system overall on the set conditions. In fact, this type of identification actually works better with the presence of noise since the latter excites the system in the whole frequency spectrum and provides valuable data outside of the normal operating region.

It has to be noted that the frequency domain has another distinct advantage here: the additive term of the watermark influences only the $0^{th}$ DC harmonic, allowing for separation of the additive and multiplicative part and their accurate estimation separately.

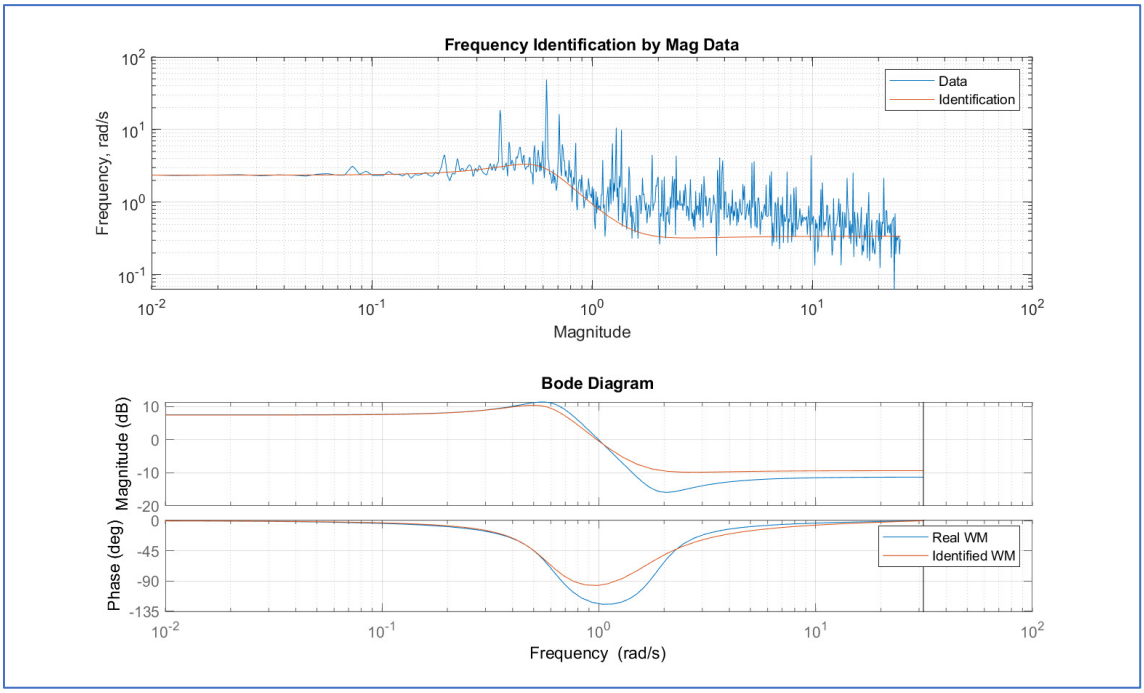



**Figure 47 Frequency Identification by Magnitude Data**

## 8.5 Identification using Model Inverse

A traditional model for system identification with present noise is the Box-Jenkins model, which accounts for noise influenced by foreign dynamics:

$$Y(z)=\frac{B(z)}{F(z)}U(Z)+\frac{C(z)}{D(z)}E(z) \tag{8.13}$$

However, in this case, the colored noise is fed to the input of the system and also the coloring filter is known. To simplify the problem, a whitening filter $H(z)=\frac{D(z)}{C(z)}$ can be derived and applied to the input and output of the system, thus reducing the problem to the following Output Error model:

$$Y(z)H(z)=\frac{B(z)}{F(z)}U(z)H(z)+E(z) \tag{8.14}$$

This model is not applicable in this form to the problem in hand, as in the watermarking scheme the disturbance error term is added to the signal before it is passed through the filter. An elegant solution for that is to switch the places of the input and output and solve for the inverse of the watermarking filter, which is causal by definition (the error term is zero-mean):

$$Y_U(z)H(z)=W_f^{-1}(z)Y_W(z)H(z)+E(z) \tag{8.15}$$

The overall whitening scheme is presented on Figure 48, where the "color" of the noise is "bleached". The filtered input-output data is subjected to offline identification using an Output Error model. The results are displayed on Figure 49, where it can be seen that the identification is practically ideal, with numerical accuracy of VAF=99.5% and EAC=99.8%.

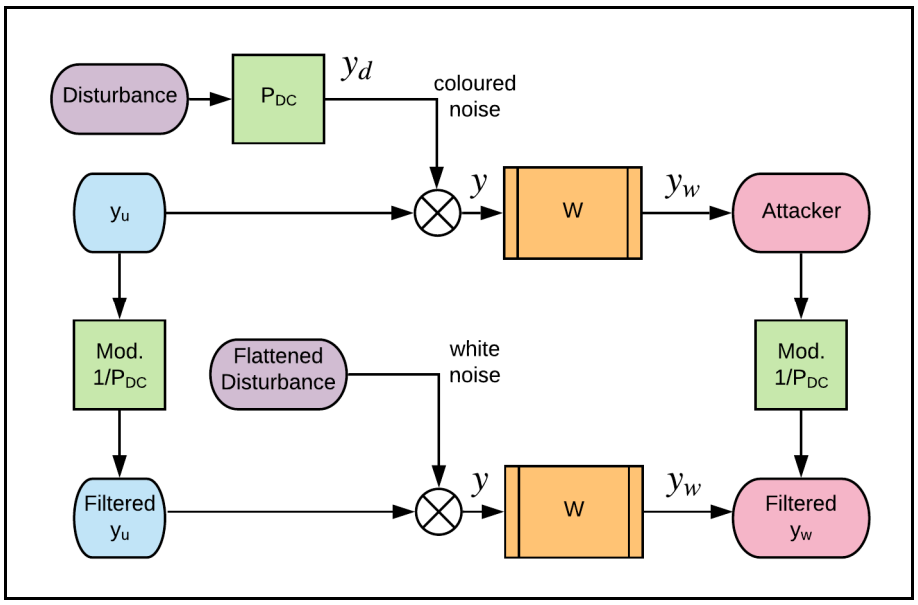



**Figure 48 Whitening of the IO Identification Data**

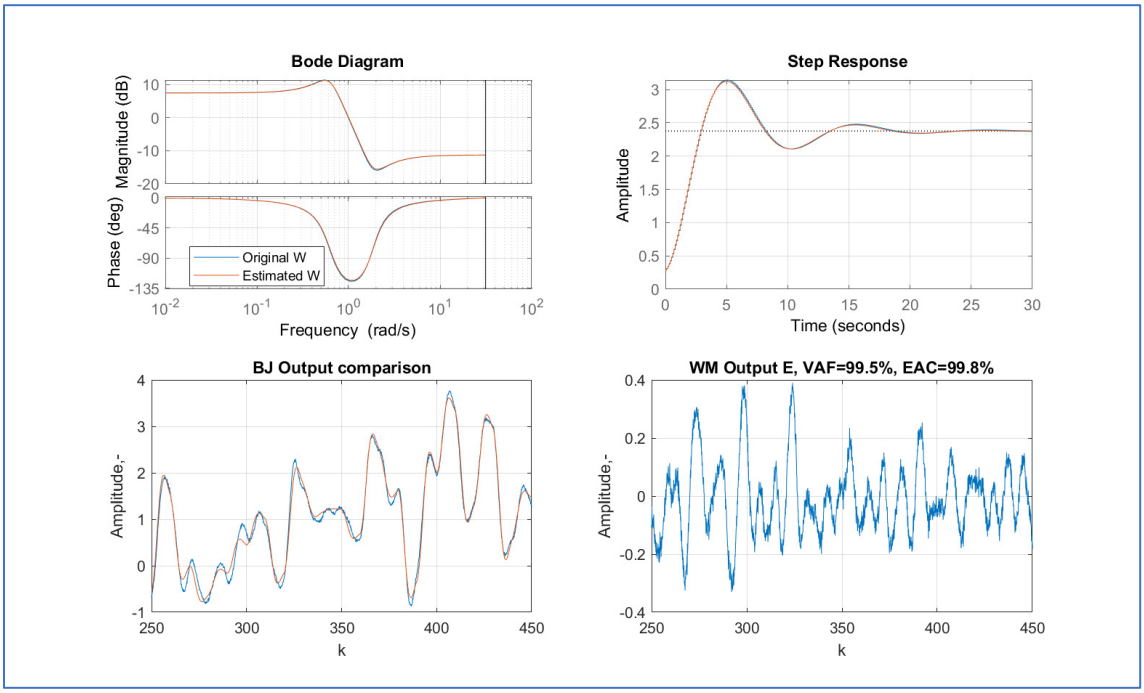



**Figure 49 Identification of an ARMAX model after whitening**

Due to the data preparation and the model assumptions, this model is robust to noise and similarly to the frequency identification noise can actually improve the results due to the full spectrum excitation.

## 8.6 Recursive Online Estimation

When dealing with a real-life scenario, perhaps of special interest will be the possibility to estimate the watermark at the earliest possible moment, even if it is with a tradeoff in accuracy. All attempted algorithms used big sets of data collected over a prolonged period of time. Achieving online estimation is not possible or feasible with subspace identification or frequency analysis due to the specific nature of those algorithms, but a number of algorithms exist for recursive estimation using the ARMAX family of models.

However, before that is possible, once again the problem of the additive term has to be solved. For simplification, the watermarking filter is reduced to a static gain and a system of equations from subsequent time steps is constructed to form a system that is not underdetermined:

$$\begin{cases} y_w(k) = \left(y(k) + W_A\right) \cdot W_M \\ y_w(k+1) = \left(y(k+1) + W_A\right) \cdot W_M \end{cases} \tag{8.16}$$

Although nonlinear, this system has 2 unknown coefficients and 2 independent equations. To solve it, the second condition is very important – meaning that the input variables must differ significantly to allow for proper estimation if noise (not included in the equations here) is present. The system is expanded to a linear problem by introducing the term $W_{MA} = W_M \cdot W_A$ with noise $e(k)$ as follows:

$$\begin{cases} y_w(k) = y(k) \cdot W_M + W_{MA} + e(k) \\ y_w(k+1) = y(k+1) \cdot W_M + W_{MA} + e(k+1) \\ \quad \dots \\ y_w(k+n) = y(k+n) \cdot W_M + W_{MA} + e(k+n) \end{cases} \tag{8.17}$$

Ideally the goal would be to minimize $e$, but that would still leave 2 variables to be optimized. Instead, the following convex optimization problem is solved by the simple 1-variable derivative-free method Golden-section search:

$$W_M = \arg\min_{W_M} \left[ Var\left(y_w - y \cdot W_{MA}\right) \right] \tag{8.18}$$

Which minimizes the variation of $W_{MA}$ by optimizing $W_M$. Once this is complete the additive term can be evaluated simply as $W_A = W_{MA} \cdot W_M^{-1}$ and its presence can be removed from the data.

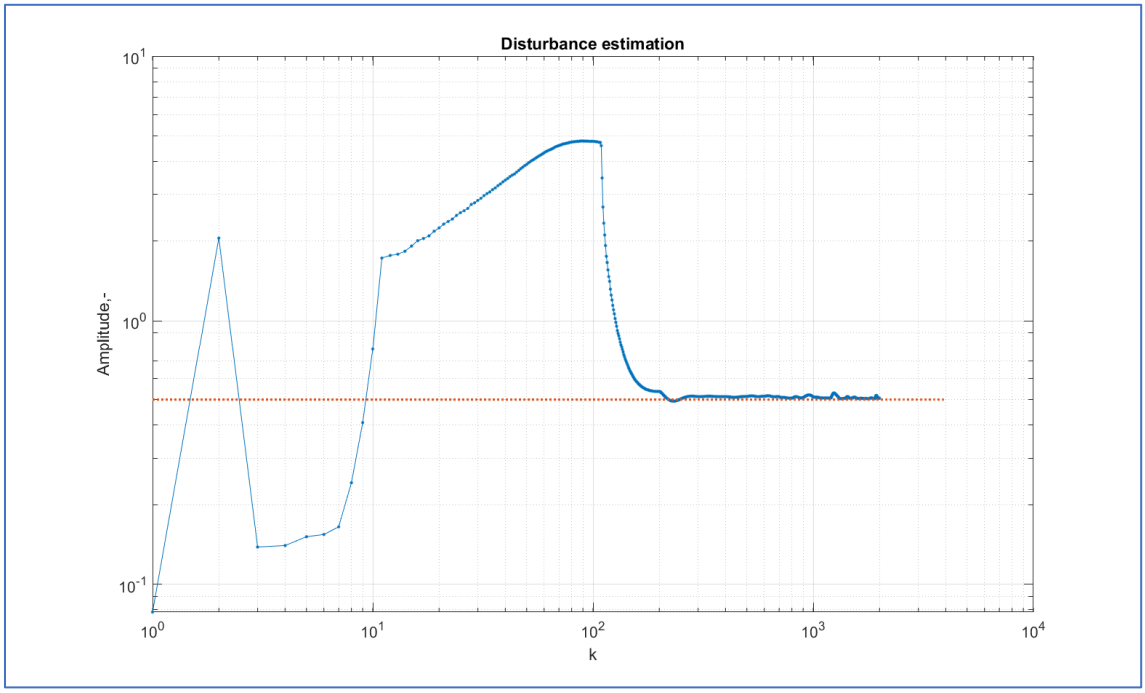



**Figure 50 Disturbance estimation via linear system solving**

This algorithm will produce accurate results even with small datasets and will converge to the true parameters as more data is collected. The main benefit of this algorithm is its simplicity in terms of speed of execution, which makes it suitable for recursive parameter estimation as a precursor to the

main identification algorithm. An example of this algorithm is presented on Figure 50, where it takes around 150 samples for the solver to arrive within 30% of the true value and the final error will be within 5% due to the previous assumption that the filter is a static gain.

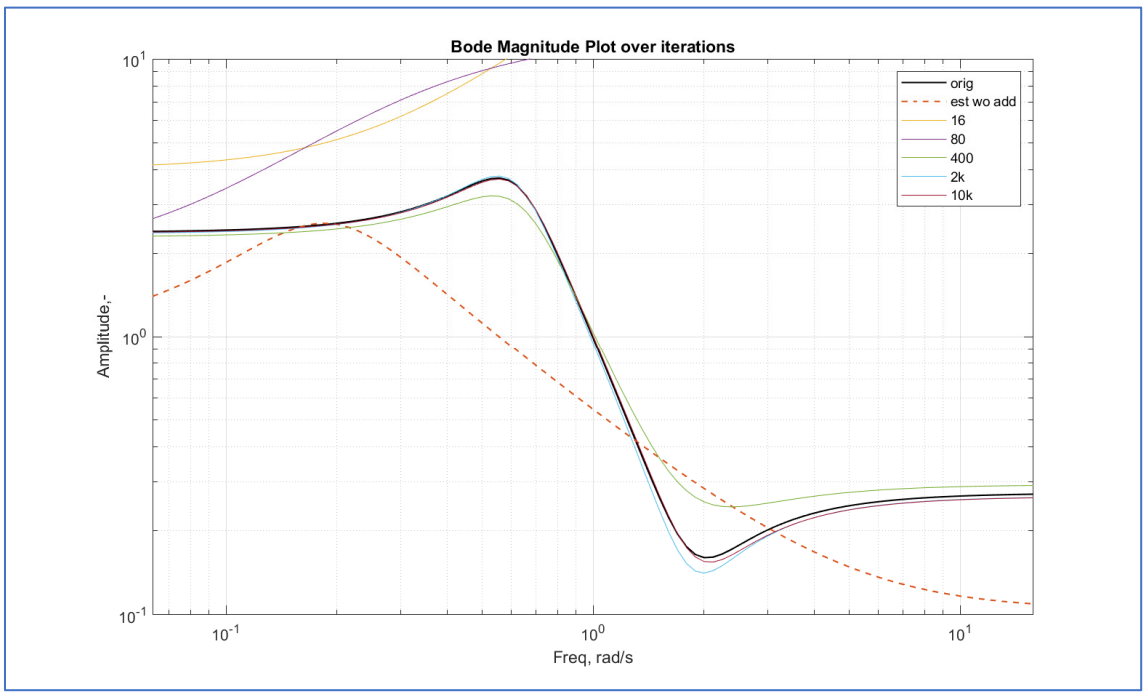



**Figure 51 Frequency Response Online Estimation Results**

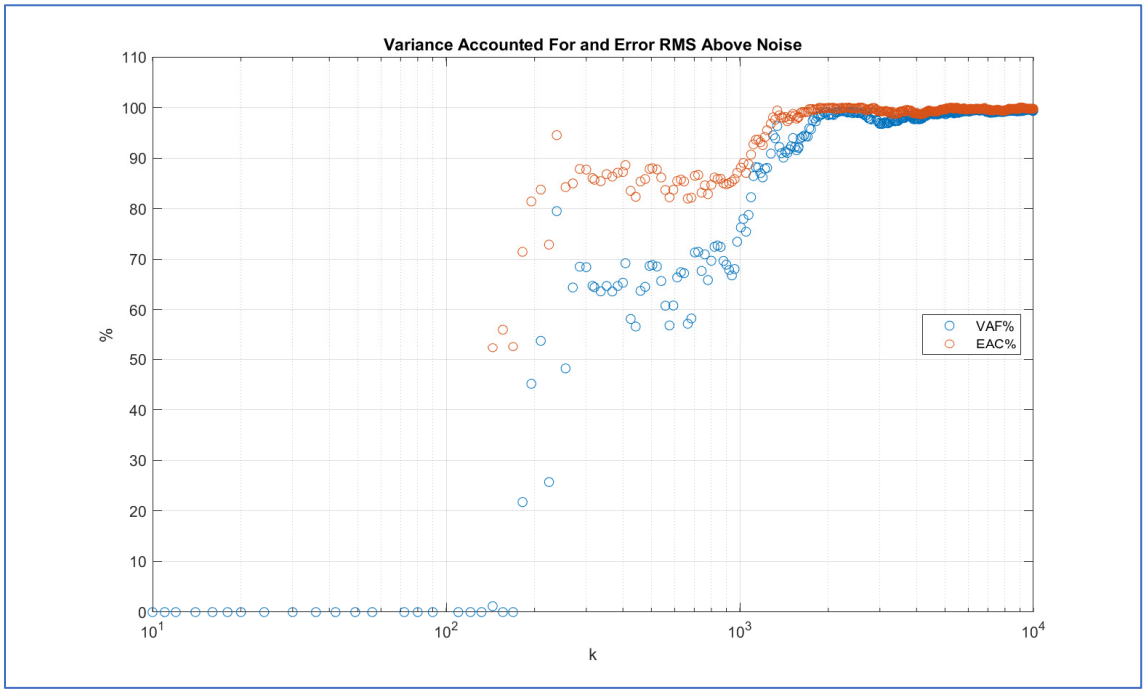



**Figure 52 Convergence of VAF and Error RMS on Online Estimation**

The recursive OE (or a modified BJ model where C=D=1) algorithm is applied on the now undisturbed dataset as follows:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k) \cdot \left( y(k) - \hat{y}(k) \right) \tag{8.19}$$

Where the last parameter estimate vector $\hat{\theta}(k)$ is updated with the difference between the predictor and actual output of the system by the gain $K(k)$. The equation in this form will converge onto the optimal system which minimizes the prediction term. The calculation is computationally light and thus can be performed online with each new measurement.

The results of the developed recursive estimation strategy with disturbance estimation are displayed on Figure 51, where the true watermark and a block estimation without removing the additive part are displayed in the frequency domain. 5 models after a number of steps of the process are visible converging to the true frequency response of the watermark filter, achieving good similarity at 400 samples. Measured quantitively on Figure 52, the familiar VAF and EAC show practically no possible identification up until 200 samples, from when on a significant increase can be observed, further improving at 1000 samples and converging towards their optimal limit at 100% as predicted in the previous chapter.

## 8.7 Performing an Attack after Data Gathering

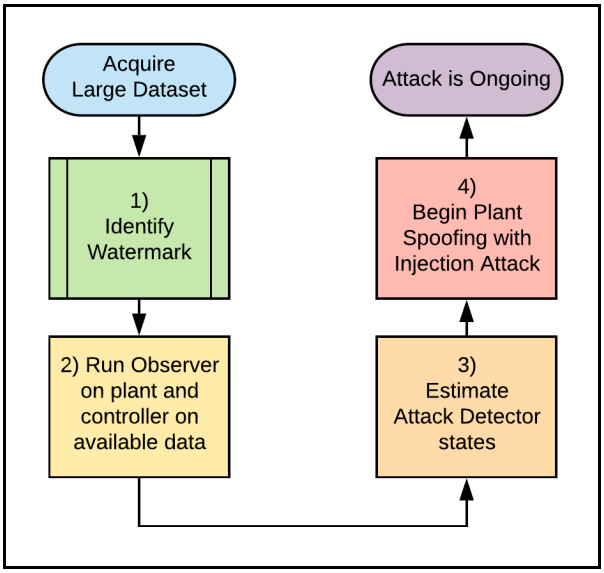



**Figure 53 Algorithm for Batch Attack**

Once suitable methods for identification and estimation have been established, an “infinite history” attack is performed after having gathered a sufficiently large dataset. A general algorithm with 4 key steps is presented on Figure 53, with the first one being one of the suggested subroutines from Sections 8.3 to 8.5. The subsequent 2) and 3) are general methods for state and input estimation, whereas the attack vector 4) is bounded in amplitude by the residual threshold of the detector, which guarantees operation without detection.

This approach has been successfully applied to the system described in Section 8.1 with the Attack Detector from Section 8.2. The results of a moderate injection attack aiming to boost the plant output to above 2 is performed and the results have been displayed on Figure 54. The identification results have not been shown here, but they are identical to the ones described in prior sections.

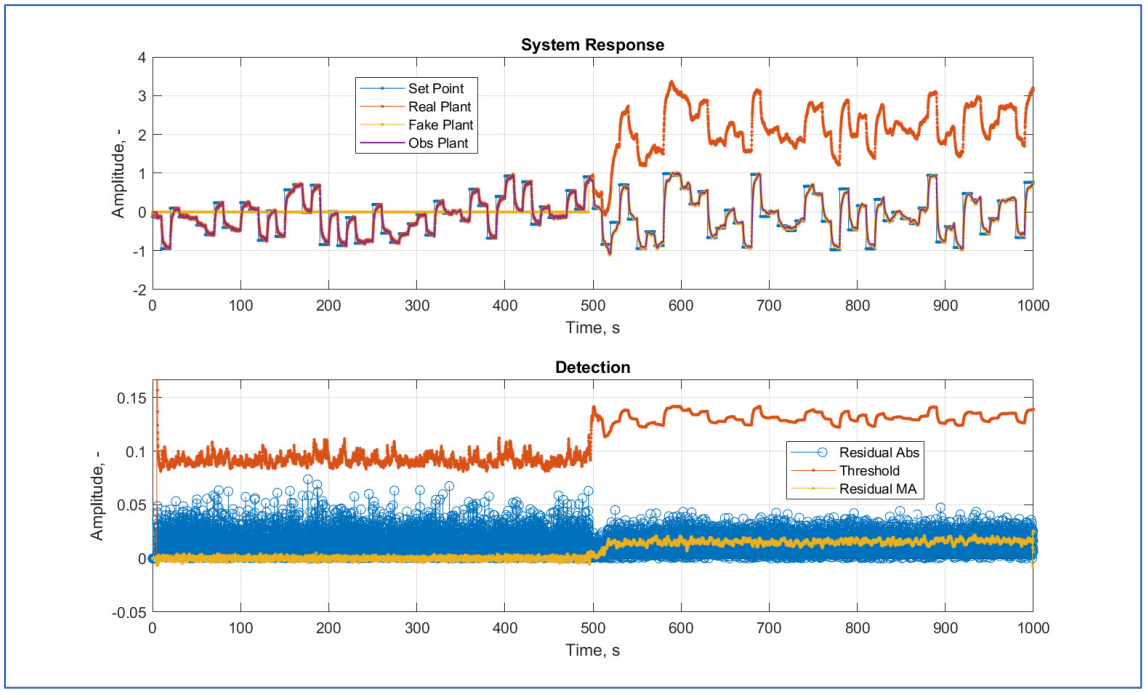



**Figure 54 Injection Attack after Data Gathering**

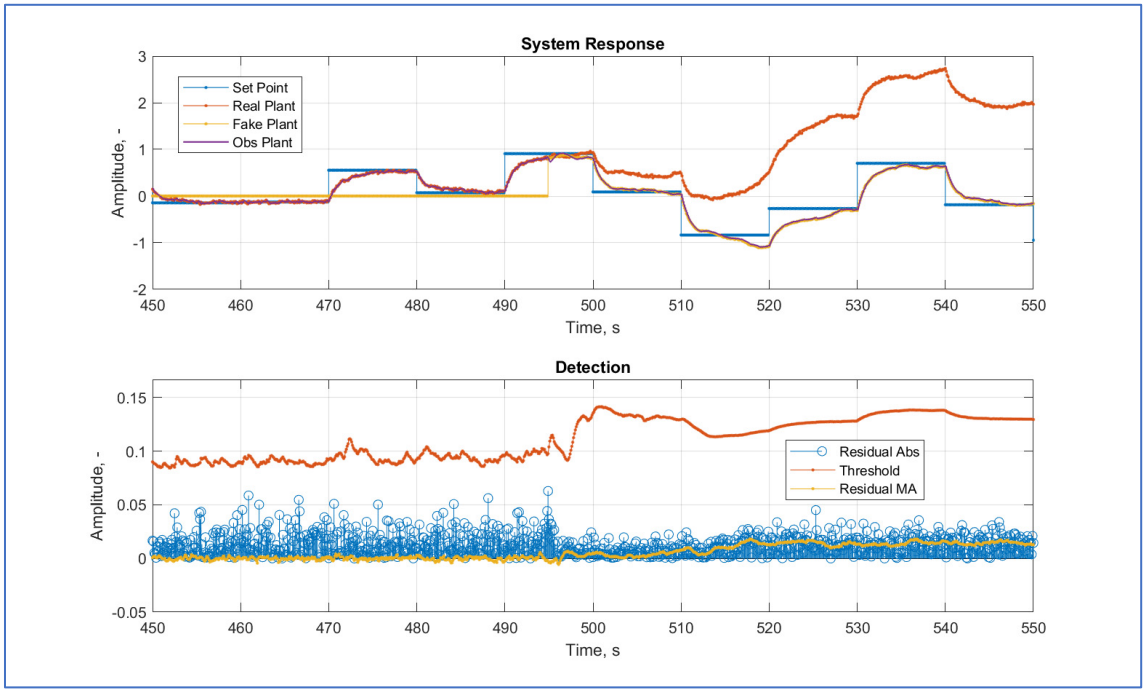



**Figure 55 Zoom of region from Injection Attack**

It can be seen, also from the zoomed-in region of Figure 55, that the attack starting at T=495s successfully "kidnaps" the plant (in red) over the course of 30 seconds and sustains it (as long as the attack is ongoing) with no alarms being raised due to the residual (blue stems bottom) not violating its threshold (red bottom). Furthermore, the Attack Detector wrongly estimates the plant position as being within norms (purple), due to the successful application of the Virtual Ideal Plant (yellow) mechanic.

The only observable change in the system is the change of mean of the residual (yellow bottom) from zero to slightly above zero – but that is not a subject to monitoring in the threshold detection.

Additionally, it can be seen that the bandwidth with the current setup allows for much higher in amplitude attacks – but that is entirely dependent on the noise present and Attack Detector settings.

But that is not all what can be performed using an injection attack. A more aggressive approach, a spoofed Control Attack aims to stabilize the real plant around a certain setpoint or have it follow a trajectory. This is accomplished by applying state feedback control to the already known states of the controller and plant to a new malicious setpoint, while still spoofing correct operations of the plant to the Attack Detector. This attack is limited in performance due to the abrupt changes required by the feedback controller being limited by the threshold, but satisfactory results can still be produced, as seen on Figure 56 and the zoomed in region of Figure 57.

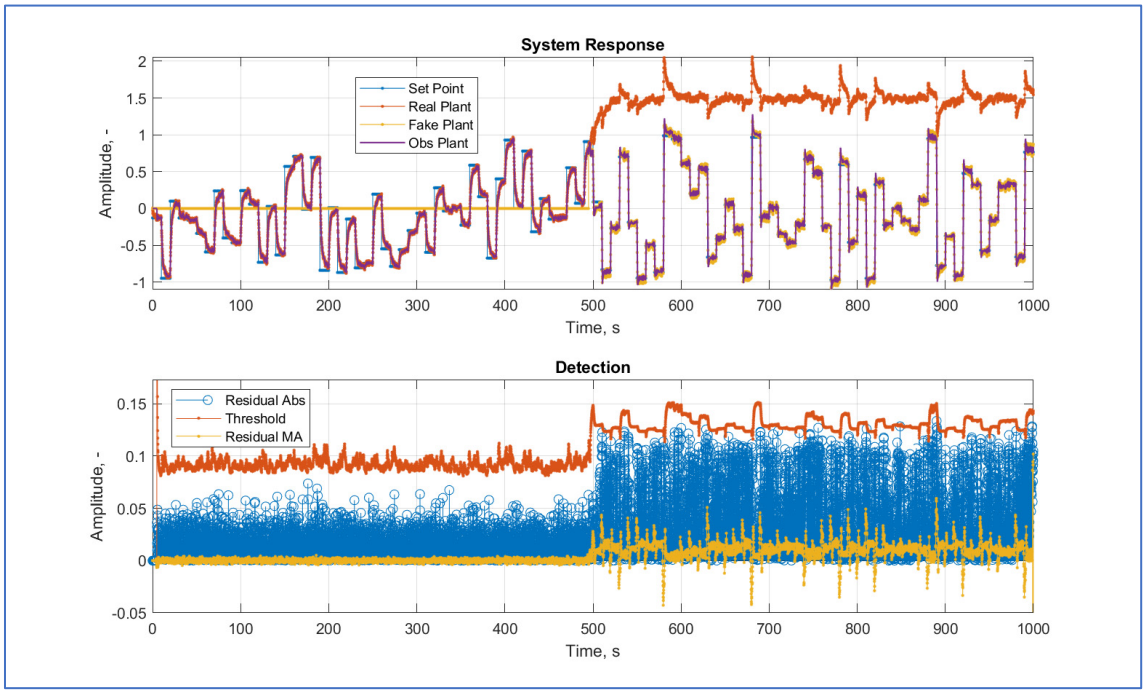



**Figure 56 Control Attack after Data Gathering**

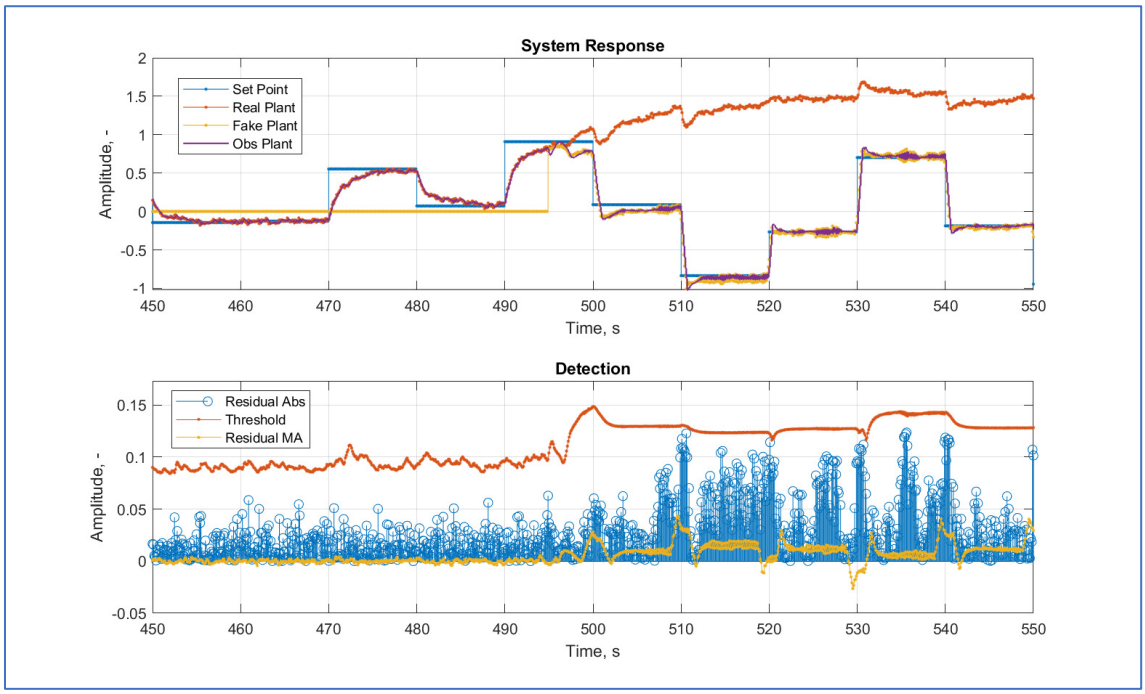



**Figure 57 Zoom of region from Control Attack**

The system operations resemble disturbance rejection, where the control of the system is the disturbance and the malicious action the controller achieving rejection. Although the residual is much more aggressive this time, its mean is still quite below the threshold. Even though the threshold appears to be very close to violation at 510s and 520s, the attack is robust due to the utilization of the fake virtual plant for full control over the attack detector states. One flaw of this approach can be seen in the observed plant, whose dynamics no longer visually represent the original dynamics without an attack, however they pass the Attack Detector test and are considered valid.

## 8.8 Performing an Attack in real-time

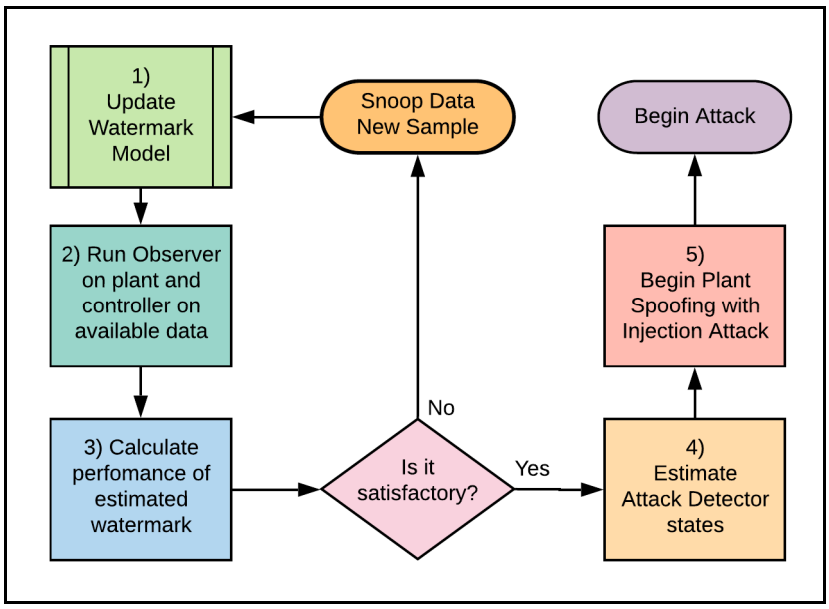



**Figure 58 Algorithm for Identification and Attack in realtime**

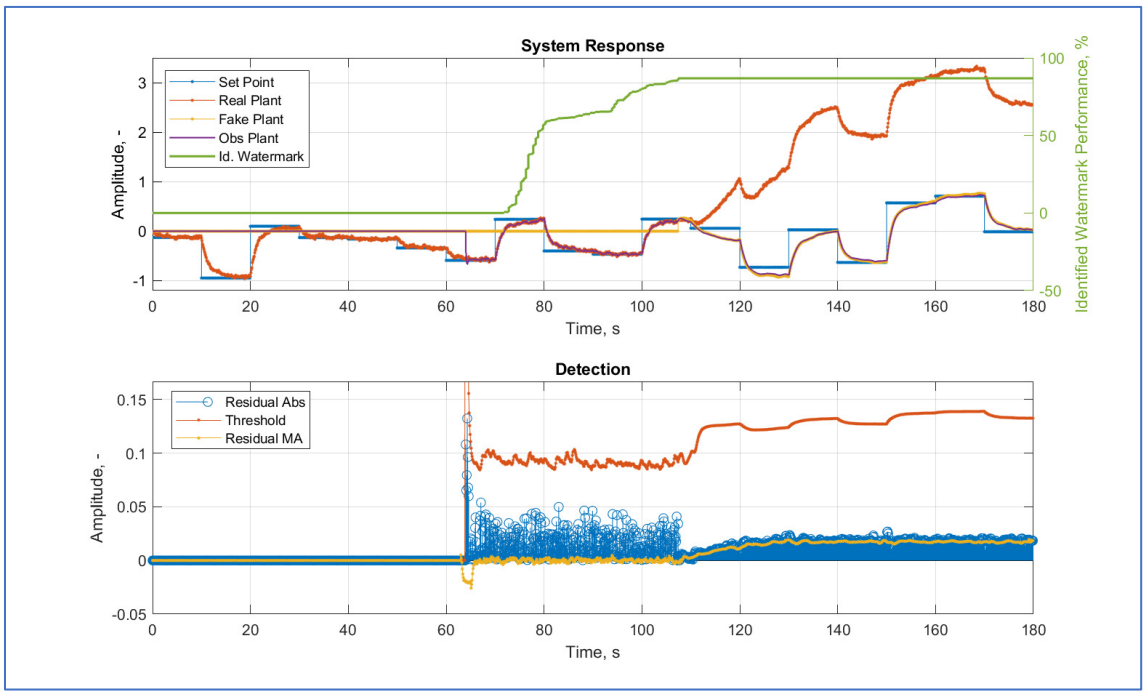



**Figure 59 Online Estimation Injection Attack**

A more realistic scenario would be when limited data with the watermark is available and an attack should be mounted as soon as possible, for example before a parameter switch occurs. The algorithm

of Figure 53 is modified to be recursive on Figure 58, using the methodology developed in Section 8.6. A Performance Indicator is calculated at each step to estimate the accuracy of the watermark and once a satisfactory number is reached, an attack can begin.

An application of this algorithm is presented on Figure 59, where the attacker and the Attack Detector start acquiring data at T=64s. The progress of the watermark estimator can be tracked via the top green plot on the right axis. Attack begins at T=107s and performs similarly to what was shown earlier in Figure 54.

## 8.9 Switching of parameters Mid-Attack

As was shown, an attack could take place after some minimal time necessary for identification. Here, the case for parameter switching is made, as discussed in Section 7.5. To illustrate that point, the last used system undergoes a parameter change at T=145s, at which the watermark parameters are changed simultaneously with new ones.

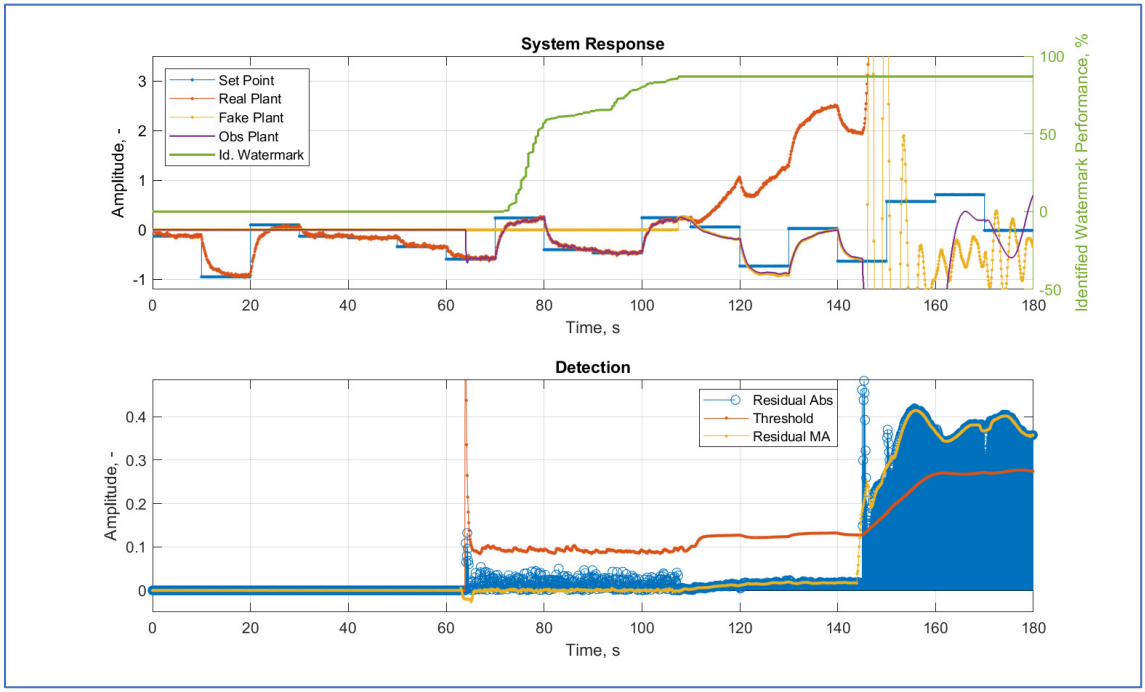



**Figure 60 Parameter change mid-attack**

While a system would continue normal operation due to the watermarking removal process, if an attack is ongoing the attack term suddenly becomes influenced by new dynamics. This is illustrated on Figure 59, where the threshold is immediately violated and thus an attack is detected.

# 9 Overview of Results

The described attack algorithms will now be evaluated for their overall accuracy and their computational cost, after which the results will be used to estimate the effectiveness of the watermarking scheme.

## 9.1 Accuracy of Attack Algorithms

For an overall comparison of the described offline algorithms, all 3 (Simplification Method, Frequency Optimization, Model Inverse) were subjected to the same conditions: 3 levels of varying noise strength (4%/20%/100% of RMS compared to reference input) and 3 dataset lengths (250 samples, 1.5k samples, 10k samples). The results are measured using the already introduced Variance Accounted For, which uses the true noise to estimate the success of the procedure, and the more heuristic one Error ACcounted, which compares the power or the error to the expected noise component.

| Attack Type | | Noise Strength / No. of Samples | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 4% | | | 20% | | | 100% | | |
| | | 0.25k | 1.5k | 10k | 0.25k | 1.5k | 10k | 0.25k | 1.5k | 10k |
| SM | VAF | 0.0% | 85.7% | 96.4% | 0.0% | 73.6% | 85.1% | 0.0% | 92.6% | 90.1% |
| | EAC | 66.1% | 99.5% | 99.8% | 12.3% | 88.8% | 94.8% | 0.0% | 99.7% | 95.7% |
| FO | VAF | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 90.4% | 85.2% | 95.4% | 99.8% |
| | EAC | 0.0% | 41.3% | 63.0% | 62.7% | 0.0% | 96.5% | 21.3% | 14.1% | 99.8% |
| MI | VAF | 0.0% | 92.8% | 90.1% | 0.0% | 99.0% | 98.5% | 71.4% | 65.4% | 98.6% |
| | EAC | 0.0% | 98.3% | 90.3% | 31.5% | 97.8% | 99.3% | 98.9% | 79.2% | 99.3% |

**Table 3 Accuracy in terms of noise strength and number of samples (100% is best)**

The performance of the algorithms has been summarized in Table 3, where the poorest performances have been highlighted in red (both VAF and EAC red means most likely complete failure of the procedure) and the high quality estimations in green (95+% on both practically guarantee an almost exact match). When in contradiction, VAF can be considered more trustworthy than EAC for theoretical purposes. As expected, all algorithms perform poorly on a small dataset under almost all conditions, but otherwise naturally the Simplification Method works best on low noise, since it ignores it, and for large noise components Frequency Optimization and Model Inverse perform quite well, largely due to the beforementioned full spectrum excitation that is provided by the white noise.

## 9.2 Computational Cost of Attacks

| Execution Time (s) | | | |
|---|---|---|---|
| Attack Type | No. of Samples | | |
| | 0.25k | 1.5k | 10k |
| SM | 0.021 | 0.127 | 3.9 |
| FO | 1.67 | 2.9 | 11.3 |
| MI | 1.4 | 1.2 | 0.9 |

**Table 4 Execution Time of Attacks (lower is better)**

Since an attack on the watermark would be a time-critical thing, the offline algorithms' execution time is measured for one run on single-core mode of execution on a i7-7700HQ Intel CPU. The results can be seen in Table 4. Interestingly, every algorithm has its own bottleneck – the Subspace

Identification of the Simplification Model procedure is limited by one single SVD operation that scales badly with the size of the dataset, whereas the simulations are heavy in Frequency Optimization. The ARMAX procedures are traditionally light, but a curious fact is the reduction of time as the dataset increases. This can be explained by the fact that the procedure by nature is iterative and a larger dataset converges faster to the stopping criteria than a smaller one.

These procedures, if one would need them for an actual time-critical attack, could be sped up immensely by the usage of hardware accelerators such as an FPGA circuit, therefore their success from the previous section should be taken into account.

## 9.3 Effect of Parameter Switching Period on Detection

As was demonstrated in Section 8.9, a switch of the watermark parameters mid-attack would immediately trigger an attack detection due to the mismatch between the attacker's spoofing watermark and the new parameters. For further detection capability analysis, table presents the results of the learning attack algorithm of Section 8.8 when facing a system with switching parameters and the noise strength fixed at 20% as in one of the scenarios from the previous section. The watermark performance threshold is set at 50%, meaning an attack will not begin until this level is reached.

| | Parameter Switching Period (s) | | | | |
|---|---|---|---|---|---|
| | 10 | 20 | 30 | 40 | 60 |
| Best WM Performance: (%) | 0 | 0 | 32.9 | 61.7 | 82.1 |
| Time to Identify (s) | >10 | >20 | >30 | 35.9 | 35.9 |
| Time to Detect (s) | N/A | N/A | N/A | 4.2 | 24.2 |
| Result | No attack | No attack | No attack | Detection at switching | Detection at switching |

**Table 5 Success of attack versus parameter switching period**

The results show that for low periods an attack does not take place due to the lack of data from the attacker perspective – a watermark will never be identified due to the reoccurring switching action of the protection scheme. As the period increases, the attacker gathers data and gradually obtains a good model of the watermark, meaning an attack can begin. However, every attack started is detected at the next switch, as indicated by the fact that the sum of the rows Time to Identify and Time to Detect equal the period itself – meaning the period value in fact sets the time duration an attack can be allowed to occur undetected.

## 9.4 Effectiveness of the Watermarking Scheme

The proposed watermark algorithm has a solid foundation, but, due to the nature of dynamic systems, its strength in terms of offered assurance about the integrity of the signal diminishes with time as more data is available to a possible intruder. The applied algorithm of Section 8.8 can serve as an indicator of that strength and can be used as a heuristic estimator for the application of the switching mechanics of Section 7.5. This, together with the specific requirements for the system can ensure secure operation with a guaranteed detection of any occurring attacks within a limited time window or their avoidance overall.

# 10 Conclusion and Further Research

An algorithm from another publication on the same topic was reviewed and several weaknesses were identified. A thorough analysis found one weak assumption and a design flaw, which make the algorithm susceptible to a Man-In-The-Middle attack with full state estimation after the recording of some minimal data. Once synchronization has occurred between the attacker and sender, full injection attack can take place without any other data needed from the sender.

A different Linear Watermarking scheme based on dynamic filters was proposed. A quick overview of the concepts of Additive and Multiplicative Watermarking was offered and their respective weaknesses identified. A hybrid approach named a "dye-pack" watermark was selected, and a full procedure was developed to apply a reversible watermark to a signal before transmission, based on a stateful generator with cryptographically secure parameter source. The same method in inverse is used at the receiver to restore the signal to its original form without any traces of the watermark, making it usable to a traditional controller from a control loop. An attack detector familiar with the dynamics of the plant, the control signal, and present noise and uncertainties, validates the received data by comparing the residual of an observer to a dynamic threshold.

A smart attacker was introduced, an upgrade from the classic Replay/Reroute/Inject attacks. A well prepared foreign entity can be expected to be fully prepared with extensive knowledge about the plant (as has already happened in reality [29]), and thus only the secret parameters and possible disturbances on the plant are unknown. To estimate the resilience of the watermarking scheme, multiple attack angles were considered, and 3 different parameter identification and state estimation algorithms were developed in the Time and Frequency domain, then tested with various settings in terms of recorded data and noise present. An online algorithm was created with automated decision-making process to demonstrate the possibility of attack at earliest convenience. Based on all findings, a parameter switching strategy is proposed to uphold the strength of the watermarking scheme by introducing fresh secrets based on a cryptographically secure source.

The algorithm was tested with a low-order watermark to provide proof of concept. Increasing the complexity by using higher order filters would inevitably make them harder for identification, both in terms of computational complexity and data required for unbiased estimation, while introducing a minimal load increase to the watermark generator due to the simplicity of state-space calculations. Whereas the presented scheme utilizes a watermark removing algorithm before the controller, an adjusted watermark parameter generator could shift the watermark into a different spectrum allowing for the watermarked value to be fed directly to the controller, possibly saving time and separating logically the control loop and the attack detector utility.

The online attack highly resembles the best secretary problem – more data would allow for a better estimation; however, this might reduce the attack window before a change in parameters. This problem could further be pursued to maximize an attacker's ability and thus improve the parameter switching strategy accordingly.

The proposed scheme is self-synchronising and practically invisible to the operating control system. An interesting aspect for future research would be its resilience to dropped packets (as in dynamic asymptotic systems any errors converge to zero with the progression of time), perhaps with slight modifications to the threshold. Another approach would be time-varying watermark generators, which due to their nature would make identification harder. A future version of this algorithm can include watermark parameters based on plant states and/or outputs, to improve the integrity confidence.

One big issue with an Attack Detector based on an observer is the possibility of system faults, which in this scenario would be hard to distinguish from an attack. A number of Fault Detectors could be set up in parallel with an Attack Detector, however great care must be taken to ensure that a hacker cannot masquerade his attack as a fault to trigger them. That would not only help them to avoid detection, but also possibly cause a system outage by abusing a certain maintenance protocol, which theoretically is still a successful attack on the availability of the plant.

In conclusion, Linear Watermarking can be successfully applied to control systems for integrity verification. Attention must be paid to the system-specific conditions and protection requirements to ensure optimal performance of the scheme.

# Bibliography


[1] R. Routledge, *Discoveries and inventions of the nineteenth century*, 13th ed. 1900.

[2] K. Stouffer, J. Falco, and K. Scarfone, "GUIDE to industrial control systems (ICS) security," *Stuxnet Comput. Worm Ind. Control Syst. Secur.*, pp. 11–158, 2011.

[3] "Control Systems Introduction," *Tutorialpoint.Com*, 2013. [Online]. Available: https://www.tutorialspoint.com/control_systems/control_systems_introduction.htm.

[4] "Getting Started with the Arduino MKR Vidor 4000." [Online]. Available: https://www.arduino.cc/en/Guide/MKRVidor4000.

[5] D. Pugliesi, "Functional levels of a manufacturing control operation." [Online]. Available: https://en.wikipedia.org/wiki/Distributed_control_system#/media/File:Functional_levels_of_a_Distributed_Control_System.svg.

[6] "Fundamentals, System Design, and Setup for the 4 to 20 mA Current Loop." [Online]. Available: https://www.ni.com/nl-nl/innovations/white-papers/08/fundamentals--system-design--and-setup-for-the-4-to-20-ma-curren.html.

[7] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017.

[8] G. Lucas and G. Kurtz, *Star Wars Episode IV: A New Hope*. 1977.

[9] "Three Aspects to Consider When Securing Industrial Automation Control System Networks." [Online]. Available: https://www.moxa.com/en/articles/three-aspects-for-securing-industrial-automation.

[10] "Layered security approach (is only as good as ….)." [Online]. Available: https://www.plixer.com/blog/layered-security-approach/.

[11] D. Dzung, M. Naedele, T. P. Von Hoff, and M. Crevatin, "Security for industrial communication systems," in *Proceedings of the IEEE*, 2005, vol. 93, no. 6, pp. 1152–1177.

[12] Nasanbuyn, "MITM Diagramm." [Online]. Available: https://commons.wikimedia.org/wiki/File:MITM_Diagramm.png.

[13] P. Hespanhol, M. Porter, R. Vasudevan, and A. Aswani, "Dynamic Watermarking for General LTI Systems," Mar. 2017.

[14] M. Yilin and B. Sinopoli, "Secure Control Against Replay Attacks," in *2009 47th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2009*, 2009.

[15] "Fourier Series 3D - interactive demonstration (Discrete Fourier Transform in 3D, FFT)," 2016. [Online]. Available: https://www.tomasboril.cz/fourierseries3d/.

[16] F. Y. Shih, *Digital watermarking and steganography: Fundamentals and techniques, (second edition)*. 2017.

[17] V. Arnold, "Arnold's cat map." [Online]. Available: https://en.wikipedia.org/wiki/Arnold%27s_cat_map.

[18] L. Pérez-Freire, P. Comesaña, J. R. Troncoso-Pastoriza, and F. Pérez-González, "Watermarking security: A survey," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006, vol. 4300 LNCS, pp. 41–72.

[19] M. Crypto, "Advanced encryption standard (AES)." [Online]. Available: https://en.wikipedia.org/wiki/Advanced_Encryption_Standard.

[20] J.-P. Aumasson, *Serious cryptography : a practical introduction to modern encryption*. 2017.

[21] M. E. Mooren, "Watermarking for attack detection in networked control systems," Delft University of Technology, 2019.

[22] J. Dumas, J. ROCH, É. Tannier, and S. Varrette, *Foundations of Coding*. Hoboken, NJ, USA: Wiley, 2015.

[23] R. M. G. Ferrari and A. M. H. Teixeira, "Detection and Isolation of Replay Attacks through Sensor Watermarking," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7363–7368, Jul. 2017.

[24] R. M. G. Ferrari and A. M. H. Teixeira, "Detection and isolation of routing attacks through sensor

watermarking," in *Proceedings of the American Control Conference*, 2017, pp. 5436–5442.

[25] R. M. G. Ferrari, T. Parisini, and M. M. Polycarpou, "A robust fault detection and isolation scheme for a class of uncertain input-output discrete-time nonlinear systems," *Proc. Am. Control Conf.*, pp. 2804–2809, 2008.

[26] T. Puleva and E. Haralanova, *Control Theory Part 2*, First. Sofia, Bulgaria: Techical University of Sofia, 2016.

[27] J. J. . M. Lunenburg, "Inversion-based MIMO feedforward design beyond rigid body systems," *Eindhoven Univ. Technol. Tech. Rep.*, no. November, 2009.

[28] Y. Mo, S. Weerakkody, and B. Sinopoli, "Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs," *IEEE Control Syst.*, vol. 35, no. 1, pp. 93–109, Feb. 2015.

[29] D. E. Whitehead, K. Owens, D. Gammel, and J. Smith, "Ukraine cyber-induced power outage: Analysis and practical mitigation strategies," *70th Annu. Conf. Prot. Relay Eng. CPRE 2017*, no. October 2016, 2017.