Detection of Replay Attack in Control Systems Using Multi-Sine Watermarking

Azam Ghamarilangroudi

A Thesis

in

The Department

of

Electrical & Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Electrical & Computer Engineering) at

Concordia University

Montréal, Québec, Canada

March 2020

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Azam Ghamarilangroudi**

Entitled: **Detection of Replay Attack in Control Systems Using Multi-Sine Watermarking**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Electrical & Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Rastko Selmic

_____ External Examiner
Dr. Walter Lucia

_____ Examiner
Dr. Rastko Selmic

_____ Supervisor
Dr. Shahin Hashtrudi Zad

_____ Co-supervisor
Dr. Youmin Zhang

Approved by _____
Dr Yousef R. Shayan, Chair
Department of Electrical & Computer Engineering

____ 3/27/2020

_____
Dr Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Detection of Replay Attack in Control Systems Using Multi-Sine Watermarking

Azam Ghamarilangroudi

Cyber-physical systems (CPSs) consist of networks of sensors, computers and actuators. This research studies a control system within a CPS in which the plant and controller are separated geographically but connected through communication links. The links could be subject to security attacks. Recently, the research focus on attack detection has been growing rapidly. This thesis aims to develop methods based on the dynamic models of CPS for detecting attacks.

This research focuses on detection of "replay attacks". First, it proposes a watermarking scheme based on injecting a sequence of multi-sine waves. The watermarking is designed in such a way that the transient response to watermarking is suppressed. A design process is proposed to reach a compromise between (i) the ease of detection of watermarking effects in the output and (ii) the limiting of output fluctuations due to watermarking (and loss of control quality). One of the benefits of this method is that it only requires frequency response of the closed loop system at a set of frequencies; a model of system is not required.

Power spectral density estimates based on periodograms of the plant output (received by the controller) are used to trace watermarking. Furthermore, replay attack detection by tracing watermarking effects in the residual of Kalman filters is also explored.

A case study involving a laboratory water tank is used to explore the proposed method. The results of linear and non-linear model simulations are presented and is shown that replay attacks can be detected successfully.

Acknowledgments

I would like to express my gratitude to my supervisors Dr Shahin Hashtrudi Zad, and Dr Youmin Zhang for the opportunity to be under their supervisory at Concordia University and their patience and continuous supports of my research. Moreover, I must express my deepest gratitude to my beloved ones in particular my parents Mahboobeh and MohammadHassan, and my brothers Amir, Eman and Amin and my sister Elham for their unconditional love and support. None of these would have been possible without their encouragement. This work is dedicated to them. My special thanks go to my brother, Eman, who has supported me wholeheartedly throughout my studies.

Contents

List of Figures

Chapter 1

Introduction

A Cyber Physical System (CPS) integrates physical processes, computational and control resources, and communication capabilities. CPS is widely used in modern society, becoming frequent in many domains, including energy production, health care, telecommunications, power generation, water and gas distribution networks, smart cities, smart buildings, smart grids, biomedical engineering, medical devices, autonomous vehicles and transportation systems. Some of the expected qualities of CPSs, to name a few, are $autonomy$, $reliability$, $security$ and ***efficiency***. $Autonomy$ refers to designing control rules, for example in a centralized, decentralized or distributed system in a way that the system works properly. One of the usages of $reliability$ is that how a rule/standard for the whole system is defined in which the system works functionally, without any critical failure, besides all of the individual standards of each system. $Security$ means that the communications are safe and can be trusted, and *efficiency* means that how a system/controller is designed in order to minimize the cost function in system while achieving the desired functionally. Also CPS is relates to the Internet of Things (IOT), as IOT forms a foundation for the CPS revolution. CPS is driving the biggest shift in business and technology since World War II. CPSs are physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core. Just as the internet transformed how humans interact with one another, CPS will transform how we interact with the physical world around us.

There are three important topics in CPS: $confidentiality$, $integrity$ and $availability$ (known as CIA [4]). $Confidentiality$ in CPS means that we can rely on the input and output of the system and we prevent any adversary (attack) to penetrate the system to read data. $Integrity$ means preventing any attack attempts to inject any data to input and output of system, and $availability$ means that we always have communication between controller and plant. If $integrity$ fails, it means that an attacker can prevent the data from reaching the plant or controller or it can inject an attack. CPS requires improved tools which enable us to design methodology that supports: 1) specification, modeling and analysis of continuous and discrete models or models of computation, as well as networking, interoperability and time synchronization; 2) scalability and complexity management through interfacing with a synthesis of systems; 3) validation and verification of stochastic models, as well as simulation and certification.

In this part, some definitions of attack are introduced [2].

Replay Attack: In this type of attack, the attacker reads the information of input and output of the system that respectively comes from and goes to the controller, without knowing any information of the system. It manipulates the input and output in a way that it adds a signal u to the control signal (e.g. a multiple of it) and it repeats the output, as controller does not notice any difference in the output of the system.

Covert Attack: In this type of attack, the attacker knows all of the information of the system, reads the input and output data to/from the system, and can inject an input to the system and reciprocally to the output of the system which neutralizes the effect of added input. In this way the controller does not recognize the existence of attack.

Zero Dynamic Attack: In this type of attack, the attacker knows complete knowledge of system plus the initial condition of states, and does not need to read the input and output of the system, and just injects an attack as input on actuator channel in the same frequency of the right-half pole of the non-minimum phase system, which makes system unstable.

Bias injection Attack: In this type of attack, the attacker knows the model of the plant, but it does not need to read the information of input and output of system. The attacker adds a bias in

output and also adds a bias to state (x) as states and output (y) in a way that nothing will appear in detection filter. The need for detection of attacks has been growing significantly, especially due to many existing ways of hacking the systems. In the next section, we review some research done on this subject.

1.1 Literature Review

1.1.1 Classification of Attacks

Different authors propose different methods of attacks while the method of detecting the attacks are also presented in the same paper. For instance in [1], the authors have introduced different types of attack and also Pasqualetti in [2], beside introducing various types of attack, has introduced the methods of detecting and identifying the attacks. We will explore some of these results in the next section along with attack detection. In this part, we mention that different authors may use different definitions of attack. For example, Teixeira et al. in [1] have defined replay attack as an attack that repeats the recorded data in output, but Pasqualetti et al. in [2] have defined replay attack as an attack which injects some signal u in input and with injecting some output that subtracts from output of the system leading to the same output as it was recorded, and stated that the difference with covert attack is that the covert attack is closed loop while the replay attack is open loop.

1.1.2 Detection of Attack

The analysis of vulnerabilities of CPS to external attacks has received increasing attention in the past 10 years. Concerns about security and safety of control systems is not new, as various papers have dealt with system fault detection, isolation and recovery. CPS, however, suffers from specific vulnerabilities which do not have impact on output and classical control system, but affect the boundedness of states in a way that it makes the system unbounded for which appropriate detection and identification techniques are needed to be developed [2, 3]. Different papers study

different attacks and define them [5, 6, 7]. Some papers propose the method of coding for detecting attack [8, 9]. In [9] a method of putting a decoder inside sensor is suggested which properly works in a special condition. Also in some papers such as [9], it is argued that if we give attacker sufficient time, it can estimate the encoder matrix, so he can inject the attack properly such that we cannot detect. In confrontation of attacker and defender, what is important is which one has more information than the other one, and this may determine the winner. Some papers propose the method of injecting an Independent and Identically Distributed (IID) control signal u^* to plant input u which increases the cost function, but increases the ability to detect attack. The goal is to solve the optimization problem of minimizing the cost function versus maximizing the covariance of attack for different attacks such as replay attack and false data injection attack [10, 11, 12, 13]. In [11] it is shown that the probability of detection of attack changes based on the number of inputs which attacker can read; if the attacker can read the data (input) to which watermarking method is applied to, the probability of attack detection decreases as much as it will be equal to false alarm rate. Compared to [10] in which attack can be better detected when watermarking technique is applied, the authors of [11] study the case that attacker can read the input which defender applies the watermarking method.

Some papers illustrate that the persistent excitation condition is used to reach the goal of system identification; for example Wu et al. in [14] discuss pulse compression method in process monitoring. The results show that compared to the case of with no probing signal, the output achieves high resolution, high signal to noise ratio monitoring, and the acquired data can be used for online diagnosis. Yilin Mo in [13], [12] and Weerkkody et al. in [11] have investigated the use of an IID signal, and have showed that detecting the attack will be easier. The most common probing signals for power systems are a rectangular pulse or square wave, periodic waveform, sustained sinusoidal signals, and sustained noise signals [15]. For instance, Hauer in [15] has used square waves to probe specific oscillatory modes. In [16], the author has used a method for generating the cosine wave probing signal and have showed that using that cosine signal helps to identify the system and has compared results theoretically to another case when an IID signal is used as probing signal.

Pierre in [16] has discussed the use of the sum of many very low amplitude sinusoidal waves i.e. multi-sine signals. Briefly, the advantages of using such a signal compared to an IID is: firstly, we have complete control over the frequency content of the signal, and we can choose the frequencies for sine waves in the frequency band of interest; secondly, it does not make sharp transitions compared to an IID signal; thirdly, we need to identify system continuously with specific frequency and amplitude which is the specification of a sine wave, not an IID signal. It has also some disadvantages such as using a periodic signal like multi-sine signal excites only specific frequencies, while a non-periodic signal excites a continuous range of frequencies. So, a key for an IID signal is to excite a large number of frequencies covering the frequency band of interest. Pierre et al. in [16] proposed a way to design a good multi-sine signal in a way that the amount of Signal to Noise Ratio (SNR) for a low-level probing signal is more than other cases, and they have used multi-sine signals for system identification. In [17] Hauer et al. have applied sine wave, square wave and pseudo random signal as probing signals and have compared the results with each other. In [18] the authors have used pseudo random noise and single-mode square wave (SMSW) and at the end, they have compared the results for mid-level signal with low-level signal. In [19], the author has used a persistent excitation signal for regulation/tracking problem.

Morrow et al. in [20] has studied the use of a probing signal such as an IID signal to detect replay attack. The system is from Distributed Flexible AC Transmission System (D-FACTS). In [21] the author have studied also the sub optimal technique with stochastic game approach. In [22] the author has declared that using a dynamic detector, the number of measurements needed for detection of attack is lower than the number of measurements with static detector. [23] declares that if the system does not know the initial state and the attacker knows, the attacker can damage the system while being stealthy. The zero state inducing attack is also proposed, and it happens when the attacker does not change the system sensor output. It is always stealthy. Some papers such as [24] have studied smart sensors which can send innovation signals, residuals, instead of output to the controller. In this way the attacker can read the residual and make an attack signal with linear change in real residual. Here mean and covariance of the attack signal is the same as

the innovated signal produced by a smart sensor. Because there is a false data detector in controller side (which detects attack based on the residual), and the statistical specifications of attacker signal is the same as innovation signal, the detector cannot detect the attack while the attack can be designed in a way that the covariance of the error in remote estimator will increase much more than the case that we do not have any attack. In this way, the attacker can harm the system without the defender realizing it. In [25, 26], producing randomly A,B,C,D of system with fast speed (a seed) and replacing A,B,C,D of system every step, and making the system consistent, when the matrices of the controller and detector change, prevents any attack that needs the model of the system; therefore, attack cannot read and guess the system dynamic as fast as the dynamic of the system is changing. As a result, if the attacker enters the system, it will not succeed. Meanwhile, some papers deal with Denial of Service (DOS) attack. Krotofil et al. [27] have studied the effect of DOS attack that depending on the time it is applied, it can have the worst effect. The paper [28] is based on the nonlinearity specification: in a non-linear system the effect of injecting two inputs in the output of system is not equal to sum of the output effects of each input. Loosely speaking in [29, 30, 31, 32], the authors completely and comprehensively study the subject of CPS and different existing methods that are available for detecting the attacks. Table 1.1 they are categorized. For instance, in power grid context, Liu et al. [33] has investigated false data injection attacks by inserting arbitrary errors into sensor measurements. The authors analyzed two attack outlines in which the attacker is either constrained to some specific meters or limited in the resources required to compromise meters. For each scenario, algebraic conditions are derived to validate the existence of stealthy attach vectors, which do not make any change to the residue. In the same field, Sandberg et al. [34] have represented numerous security measure methods that model the least attempt needed by an attacker to inject false data to Supervisory Control and Data Acquisition (SCADA) systems. To design such a method, the authors have explored the physical topology of the power network, provided situational awareness to the system operator in an effort to interpret data manipulation. Pasqualetti et al. [35] have analyzed attacks on sensors and actuators by considering a generic continuous time control system. The authors have defined

special conditions that provided the probability of detecting such attacks, given a set of known susceptibilities. In [36], Irita et al. propose a detecting method by adding a white Gaussian noise as a code signal to both sensor output and also control output (input of plant), and replay attack can be detected using fault diagnosis matrices even if the code signal is decrypted. This paper proposes a robust detection system created by introducing a replay attack detection method that sacrifices control performance to code signal. The authors have proposed a bargaining game which has agreed on control input noise and considers control performance and detection precision. Based on sensor output and state estimated values, fault diagnosis matrices for detecting replay attack are used.

Table 1.1: A brief taxonomy of CPS approaches from a control-theoretic perspective [29]

Type of System	**Noise**	**Attack Models**	**Detect Mechanism**	**Reference**
Power grid	✓	false data injection on sensors	Residue detector	[33, 34]
Control systems	-	Attacks on sensors & actuators	Detection filters/ Optimization decoders	[37, 35]
Control systems/Sensor network	✓	Dynamic false data injection(sensor attack)	Residue detector	[38, 39]
Control systems	✓	Replay attack	χ^2 detector & correlation detector, Physical watermarking,	[13]
Wireless Network	-	State attacks	Output estimator	[40]
Distributed Network	-	State attacks	Combinatorial estimator	[41]
Consensus Network	-	Malicious or faulty nodes	Detection and identification filters	[42]
Control systems-Power grid	✓	Replay or Covert attack	Detection and system identification	[20]

Mo et al. in [38] have considered a data injection attack on a noisy wireless sensor network. The attack is modeled as a constrained optimal control problem in which the Kalman filter is used to perform state estimation, while a failure detector is employed to detect anomalies in the system. Similarly, Mo et al. in [39] have considered attacks on control systems in a noisy environment. The adversary in this system is aware of the plant model, noise statistics, the controller and state

estimator. The attacker can also manipulate a set of sensors. Necessary and sufficient conditions are derived for the feasibility of a dynamic false-data injection attack where an attacker can cause unbounded errors in the state estimate without substantially increasing the probability of detection by a residue detector. Additionally, an algorithm to perform such an attack is derived. This method involves rendering unstable modes in the unobservable system. Using redundant sensors to measure unstable modes is suggested as a method to improve resilience to such an attack. Pajic et al. [40] have analyzed the impact of malicious nodes in the context of a wireless control network. The authors have designed and assessed the effectiveness of a detector based on an approach that aims at estimating sensor outputs. In a similar work addressing attacks on system states, Sundaram et al. [41] have proposed a combinatorial procedure to compute the initial state of a distributed control system to infer such attacks. Pasqualetti in [42] also has characterized the effect of unidentifiable inputs on the consensus value and has proposed three failure-sensitive filters to detect and identify malicious or faulty nodes. Verrelli et al. in [19] have studied and considered persistent excitation condition on a regulation/tracking problem of a rotor position. In [43], the authors have studied different attacks including DOS, replay and deception attack, and the methods of detecting the attacks such as Bayesian detection with binary hypothesis, weighted least square method, χ^2 detector based on Kalman filter, and Quasi-FDI (Fault Detection and Isolation technique). Bayesian detection with binary hypothesis is widely applied in the data fusion of sensor networks since it is easy to formulate.

1.1.3 Attack Accommodation

Fawzi et al. [37] has focused on the design, implementation, analysis and characterization of robust estimation and control in CPS when they are affected by corrupted sensors and actuators. He has mentioned that if more than half of the sensors are attacked, it is impossible to accurately reconstruct the state of the system. Yuan et al. [44] has designed a security resilient controller for CPSs under Denial-of-Service (DOS) attack. In fact a coupled design framework incorporates the cyber configuration policy of Intrusion Detection System (IDS) and robust control of dynamical

system. Yuan et al. designed algorithms based on value iteration methods and Linear Matrix Inequalities (LMI) for computing the optimal cyber security policy and control laws. Lucia et al. [45] have proposed a method of designing a sequence of $\mathcal{N}$ robust one-step controllable set and then by designing a state feedback or output feedback controller, they have proposed a supervisor above the system which by checking the pre-check and post-check data in each step and comparing to the amount that should be in the zone, is able to detect attack and find the minimum cost function of system. In [44], Yuan et al. have studied DOS attack also. They proposed a resilient controller against this attack while the performance of the system remains in an acceptable level based on an LMI algorithm and H^{∞}. Rebai in [46] has proposed an event-based implementation in order to archive novel security strategy. By solving a sufficient Bilinear Matrix Inequality (BMI) condition, controller gain is deduced. Li et al. in [47] using the method LMI has controlled the system which is under fault/attack.

1.1.4 Transient Response Suppression

In [49] the authors have studied how to drive a transducer in such a way to produce a steady-state tone burst. By beginning and ending at zero crossings of the sine, i.e. the usual turn on, turn off, transient is suppressed. The goal is to produce sound radiation in the surrounding fluid medium without any transient response using a transient suppressed drive.

1.2 Thesis Objectives and Contributions

Watermarking is one of the methods for detection of replay attacks. In literature, random IID signals are proposed for watermarking.

This thesis proposes a watermarking approach using sine waves. The main advantage of this approach is that it only requires the value of frequency response of the system at a finite set of frequencies used in watermarking. This information can be obtained experimentally and a mathematical model of plant is not required.

To enhance effecting of the method, the frequency of sine waves are changed from time to time. The thesis also propose a method of choosing the watermarking signal to suppress the transient response from applying the sequence of sine waves. Sine waves are smooth and do not increase the actuator wear. Furthermore, the output fluctuations resulting from watermarking can be easily adjusted in the proposed design process.

A case study involving a laboratory tank is used to study the application of the proposed method.

1.3 Thesis Outline

In Chapter 2, different attacks, models and mathematics formula, main concepts and definitions used in this thesis are presented. In Chapter 3, replay attack detection via injecting sine wave instead of an IID signal is described. A method for suppressing the transient part of the output of a system resulting from sine wave is presented and attack detection via periodogram is proposed. In Chapter 4, a case study involving a laboratory tank system is presented and models are used and replay attack detection using watermarking, periodogram and Kalman filter is studied. Finally, Chapter 5 concludes the thesis and highlights the future research directions.

Chapter 2

Background

In this chapter we define every attack to some extent in detail. Before that, we review some definitions:

2.1 Definition of Attacks

In this section, we study some definitions about attack.

Disclosure Resources: When the attacker can read information from either U (actuator channel) or Y (sensor channel), it is said that they are disclosure resources.

Disruptive Resources: When the attacker can inject data on channel or modify the availability of channels.

Confidentiality refers to disclosure resources while integrity and availability refers to disruptive resources.

Data Deception Resources: Before introducing this attack, we define the system. Our system is defined as:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \tag{2.1}$$

where $x :\in \mathcal{R}^n$ and $A \in \mathcal{R}^{n\times n}, B \in \mathcal{R}^{n\times m}, C \in \mathcal{R}^{p\times n}$, and $D \in \mathcal{R}^{p\times m}$. If we assume discrete u and y as u_k and y_k, this attack modifies the control signal u_k and output y_k to corrupted signals $\tilde{u}_k$

and $\tilde{y}_k$, the deception attack can be modeled as:

$$\begin{cases} \tilde{u}_k = u_k + \Gamma^u b_k^u \\ \tilde{y}_k = y_k + \Gamma^u b_k^u \end{cases} \tag{2.2}$$

Where $b_k^u \in \mathbb{R}^{|\mathcal{R}_1^u|}$, $b_k^y \in \mathbb{R}^{|\mathcal{R}_1^y|}$ and $\Gamma^u \in \mathbb{B}^{n_u \times |\mathcal{R}_1^u|}$ and $\Gamma^y \in \mathbb{B}^{n_y \times |\mathcal{R}_1^y|}$, $\mathbb{B} := \{0, 1\}$ are binary matrices mapping the data corruption to respective channels [1]. One model of data deception resource attack is bias data injection which is described as following:

Physical Resources: Physical attacks may occur in control systems, Physical attacks are very similar to fault signals as we have the system (2.3)

$$\begin{cases} x_{k+1} = Ax_k + B\tilde{u}_k + Gw_k + Ff_k \\ y_k = Cx_k \end{cases} \tag{2.3}$$

where w_k is disturbance and f_k is fault. Now if we want to specify a physical attack, F is the attack signature and f_k is the attack signal.

Teixeira et al. in [1], have represented the model of attack based on Fig. 2.1.

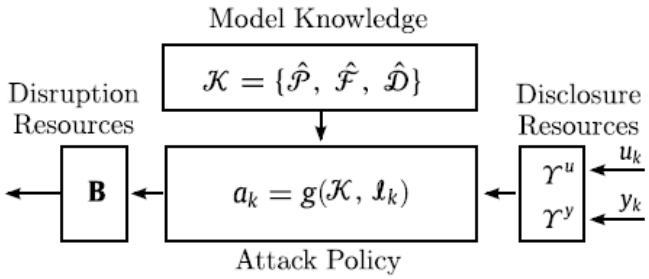


Figure 2.1: Model of attack [1]

The adversary model considered in this paper is illustrated in Fig. 2.1 and is composed of an attack policy and the adversary resources i.e., the system model knowledge, the disclosure resources, and the disruption resources. $\mathcal{K} = \{\hat{\mathcal{P}}, \hat{\mathcal{F}}, \hat{\mathcal{D}}\}$ is a primary model knowledge possessed by the adversary; l_k corresponds to the set of sensor and actuator data available to the adversary at

time k as represented in Eq. (2.4), thus being mapped to the disclosure resources; a_k is the attack vector at time k that may affect the system behavior using the disruption resources addressed by B. The attack policy mapping $\mathcal{K}$ and l_k to a_k at time k is denoted as $a_k = g(\mathcal{K}, l_k)$.

2.1.1 Replay Attack

Reply attack can reset the measurements to reflect the prerecorded nominal operating condition and to hide the effect of state attack on the system dynamics. Reply attack can access all sensors without knowing the dynamic of system. Pasqualetti et al. [2, 3] have described the replay attack as follows:

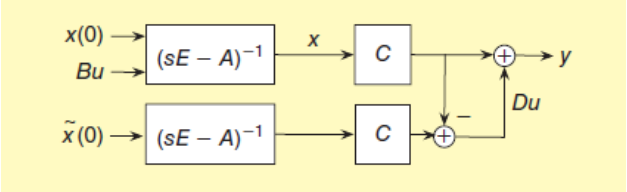


Figure 2.2: Replay attack [2]

As Fig. 2.2 shows, replay attack can be modeled as input $(Bu, -Cx + C\tilde{x})$ when x is the state under attack and $\tilde{x}$ is the state without attack respectively. While in [1] it is described as follows. The disclosure attacks can be modeled as following:

$$PhaseI : \begin{cases} a_k = 0 \\ l_k = l_{k-1} \cup \left\{ \begin{bmatrix} r^u & 0 \\ 0 & r^y \end{bmatrix} \begin{bmatrix} u_k \\ y_k \end{bmatrix} \right\} \end{cases} \tag{2.4}$$

where l_k is the control and measurement data sequence gathered by the adversary from time k_0 to k_r (duration of disclosure resources) and $l_{k0} = 0$ and $r^u \in \mathcal{R}^{n_u \times n_u}$ and $r^y \in \mathcal{R}^{n_y \times n_y}$ are the binary incidence matrices mapping the data channels to the corresponding data gathered by adversary. This type of attack does not affect the physical dynamic of system. Disclosure resource is depicted in Fig. 2.3.

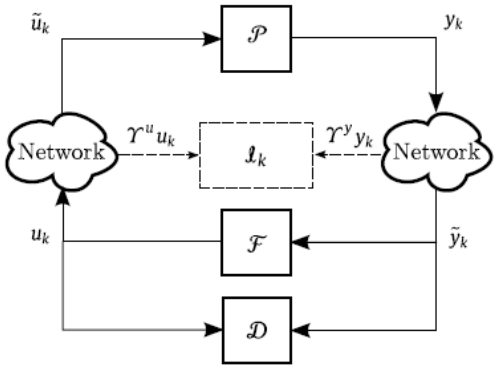


Figure 2.3: Phase I of replay attack [1]

Eq. (2.4) is phase I of attack policy shows that the attack reads the data in this step. Next step is the injection of some data (disruptive resource) as Fig. 2.4 shows.

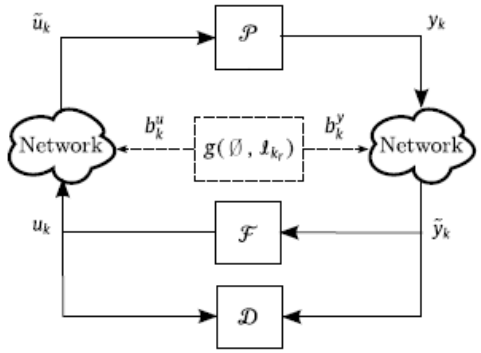


Figure 2.4: Phase II of replay attack [1]

where $\tilde{u}_k$ and $\tilde{y}_k$ are input and output after injecting attack, with $k_0 \leq k \leq k_r$ and $l_{k0} = 0$ and

$$PhaseII: \begin{cases} a_k = \begin{bmatrix} g(k, l_k) \\ r^u(u_{k-T} - u_k) \\ r^y(y_{k-T} - y_k) \end{bmatrix} \\ l_k = l_{k-1} \end{cases} \tag{2.5}$$

where $T = k_r + 1 - k_0$. In replay attack, attacker reads data from $k = k_0$ to k_r, gathering the sequence data l_k and then begins replaying the recorded data at time $k = k_r + 1$ until the end of attack at k_f. b_k^u, b_k^y are attack signals in input and output which are described in Eq. (2.2). In this

type of attack, attacker needs no information about model of system, and if he have access to all channels, he can be stealthy.

2.1.2 Zero Dynamic Attack

In [1] zero dynamic attack is defined as follows:

$$\begin{cases} x^a_{k+1} = Ax^a_k + Ba_k \\ \tilde{y}^a_k = Cx^a_k \end{cases} \tag{2.6}$$

a_k, attack signal is defined as $a_k = \gamma^k g$, where $\gamma \in \mathcal{C}$ are the roots that causes matrix $p(\gamma)$, represented in Eq. (2.7), to lose the rank. In discrete time system the minimum phase zeros are defined $|\gamma| < 1$ and for zero dynamic attack we just consider the non-minimum phase zeros, $|\gamma| > 1$, because they just can cause zero dynamic attack which makes the system unbounded [50].

$$p(\gamma) = \begin{bmatrix} \gamma I - A & -B \\ C & 0 \end{bmatrix} \tag{2.7}$$

The input-zero direction is defined by solving the Eq. (2.8)

$$\begin{bmatrix} \gamma I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2.8}$$

For some initial condition x_0, g will be found. So, we have $a_k = \gamma^k g$ as zero dynamic attack. In [3, 2], it is described for a continuous system as follows:

considering system (2.1), Invariant zeros of system are the complex values $s \in \mathcal{C}$ yields $det(P(s))$ in Eq. (2.7) (replacing γ with s) loses rank ($Rank(P(s)) < n + min(m, p)$). let z be an invariant zero, and let x_0 ,u_0 such that:

$$\begin{cases} (sI - A)x_0 - Bu_0 = 0 \\ Cx_0 + Du_0 = 0 \end{cases} \tag{2.9}$$

where x_0, u_0 are **state zero direction** and **input zero direction**. If we define trajectory $x(t) = x_0 e^{zt}$ and u as $u_0 e^{zt}$ so we have:

$$y(t) = Cx + Du = Cx_0 e^{zt} + Du_0 e^{zt} = e^{zt}(Cx_0 + Du_0) \tag{2.10}$$

The state trajectory x is called **zero dynamic**.

2.1.3 False Data Injection Attack

It is a type of attacks which injects an adversary signal to deceive the detector. Different papers describe it in different ways. Pasqualetti in [2, 3], described it as follows. The attacker corrupts the system dynamics and measurements to render the unstable mode p unobservable from the measurement. Dynamic false data injection attacks require access to some sensors and knowledge of system dynamics to be implemented.

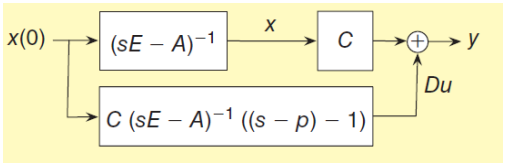


Figure 2.5: false data injection attack [2]

Dynamic false data injection attack acts as it makes change the states in a way that it makes one unstable mode but we do not see the effect of that state in the output of system, so we do not detect attack. Liu et al. [33] described it as following: As Fig. 2.5 shows, for this type of attack attacker needs just perfect information of system, no need to read data of channels (an open loop attack) and also needs to disruptive resources. Considering system (2.1), if we have noise in measurement and assuming $D = 0$, we have:

$$y(t) = Cx + e \tag{2.11}$$

So, we define matrix W as covariance matrix compounds of covariance of each noise in diagonal elements and zero for other elements in matrix. Therefore, we estimate x as $\hat{x} = (C^T W C)^{-1} C^T W y$.

If we have state estimator, residual will be $r = y - \hat{x}$, and for being stealthy, $||y - \hat{x}|| < \tau$ should be satisfied, where τ is a threshold that is defined in system. Now Attacker acts in a way that it adds an amount to y and $\hat{x}$ as follows:

$$\begin{cases} y_a = y + a, \\ \hat{x}_{bad} = \hat{x} + d \end{cases} \tag{2.12}$$

$$||y_a - C\hat{x}_{bad}|| = ||y + a - C\hat{x} - Cd|| = ||y - C\hat{x} + \underbrace{a - Cd}|| = ||y - C\hat{x}|| \tag{2.13}$$

If $||a - Cd|| = 0$ we will have $||y_a - C\hat{x}_{bad}|| = ||y - C\hat{x}|| < \tau$ so the attack is stealthy and can not be detected. If we write the system as Eq. (2.14)

$$\begin{cases} y_a = y + a, \\ ||y - C\hat{x}|| = ||y + a - C(\hat{x} + d)|| = ||y - C\hat{x} + a - Cd|| \leq ||y - C\hat{x}|| + ||a - Cd|| \end{cases} \tag{2.14}$$

Even if we have $||a - Cd|| < \tau_a$, in which $\tau_a = \tau - ||y - C\hat{x}||$, we do not detect the attack, because we still have this condition:$||y - C\hat{x}|| \leq \tau$ [33].

2.1.4 Covert Attack

Pasqualetti et al. [2, 3] described that this type of attacker should know dynamic model of the system and read both channel input and output, and inject data on both channels in a way that y_a which attack injects in output neutralizes the effect of injected input attack u_a.

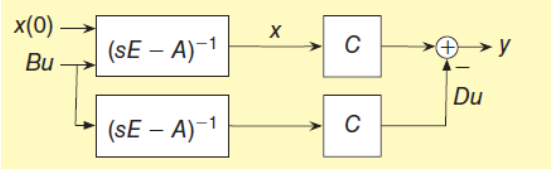


Figure 2.6: Covert attack [2]

If we assume x is the state without attack and $\tilde{x}$ is the state under attack, in covert attack,

attacker injects the signal u as input and $y = Cx - C\tilde{x}$ as output, in which:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = C(x - \tilde{x}) \end{cases} \tag{2.15}$$

Actually the covert attack input is $(Bu, -C\tilde{x})$, where $\tilde{x}$ satisfies $\dot{\tilde{x}} = A\tilde{x} + Bu$ with $\tilde{x}(0) = 0$. In this type of attack, attacker needs the full knowledge of model system, reads both channels (disclosure resources), and injects attacks on both channels (disruptive resources).

2.1.5 Denial of Service Attack

In Denial of service attack, the attacker does not need to know the dynamic of system and read the data, just he prevents the data to reach the actuator or from sensor to controller (availability property).

2.1.6 Eavesdropping Attack

Eavesdropping attack read the data, in each of the channels or both of them (disclosure resource).

The Fig. 2.7 shows properly each attack needed information and operation region.

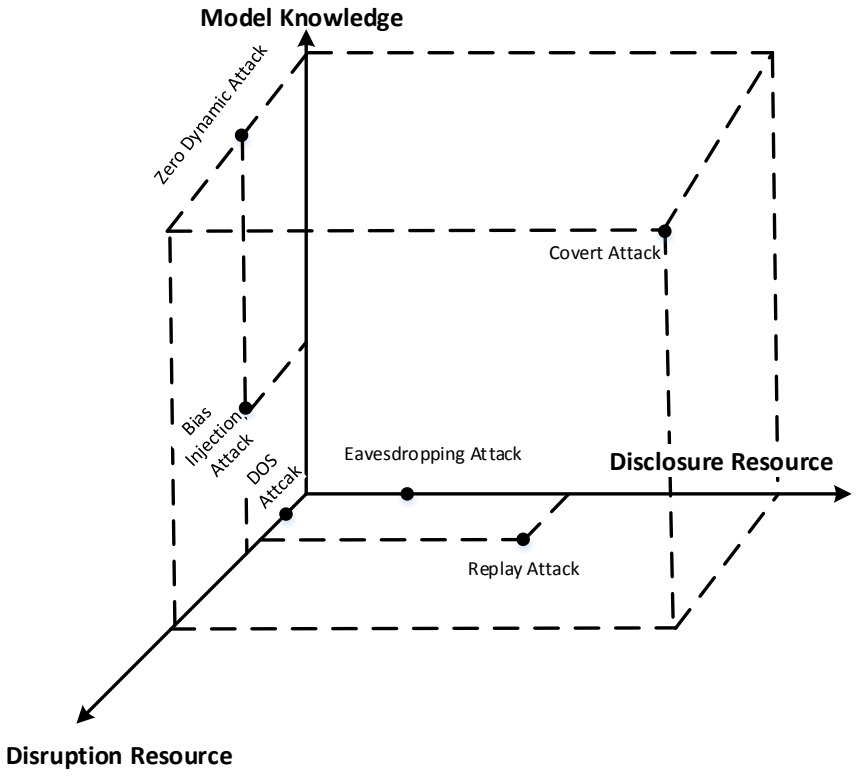


Figure 2.7: 3-D attack space [3]

2.2 Definition of Periodogram

In signal processing, a Periodogram is an estimate of the spectral density of a signal. Periodogram calculates the significance of different frequencies in time-series data to identify any intrinsic periodic signal.

Formula of Power Spectral Density (PSD) is: $P_{xx}(f) = \frac{\Delta t}{N}|\sum_{n=0}^{N-1} x(n)e^{-j2\pi fn}|^2 = \frac{1}{N}|X(f)|^2$ where f is the frequency in which the PSD is calculated around and it is $-\frac{1}{2\Delta t} < f < \frac{1}{2\Delta t}$ while $\Delta t = \frac{1}{fs}$ is sampling time. The integral of the true PSD, $P(f)$, over one period, $\frac{1}{\Delta t}$ for cyclical frequency and 2π for normalized frequency, is equal to the variance of the wide-sense stationary random process: $\sigma^2 = \int_{-\frac{1}{2\Delta t}}^{\frac{1}{2\Delta t}} P(f)df$

According to Parseval's theorem for energy signals

$$\int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} |\hat{x}(f)|^2 df = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(w)|^2 dw \tag{2.16}$$

where $\hat{x}(f)$ is the Fourier transformation of $x(t)$, and is defined based on $\hat{x}(f) = \int_{-\infty}^{\infty} e^{-2\pi i f t}$ $x(t)dt$, and $\omega = 2\pi f$ is frequency in radians per second. The interpretation of this form of the theorem is that the total energy of a signal can be calculated by summing power-per-sample across time or spectral power across frequency. The area under the PSD curve is equal to power of the signal (total signal power), $R(0)$, the autocorrelation function at zero lag. This is also the variance of the signal. The statistical average of a certain signal as analyzed in terms of frequency content, is called spectrum. When the energy of signal is concentrated around a finite time interval, if its total energy is finite, the "Energy Spectral Density" can be computed, as more commonly used. Otherwise for signals whose energies are unlimited, we calculate their power as PSD, a statement of power existing in the signal as a function of frequency. The unit of energy spectral density is $\frac{\omega}{Hz}$. The class of stationary random processes which do not have finite energy and hence do not have the Fourier transform, and such signals have finite average power and hence are characterized by a power density spectrum. PSD of a signal is Fourier transformation of Autocorrelation function. Let S_{xx} be PSD, and $R_{xx}(\tau)$ is autocorrelation. Therefore:

$$R_{xx}(\tau) = \int_{-\infty}^{\infty} S_{xx} e^{i2\pi f \tau} df \tag{2.17}$$

In other words, $S_{xx} = \int_{-\infty}^{\infty} R_{xx}(\tau) e^{-i2\pi f \tau} d\tau$

The method of averaged periodograms, more commonly known as Welch's method, in which a long x[n] sequence is divided into multiple shorter, and possibly overlapping parts. It computes a windowed Periodogram of each one, and computes an average array, i.e. an array which each element is an average of the corresponding elements of all the periodograms. For stationary processes, this reduces the variance of the signal [51], [52], [53].

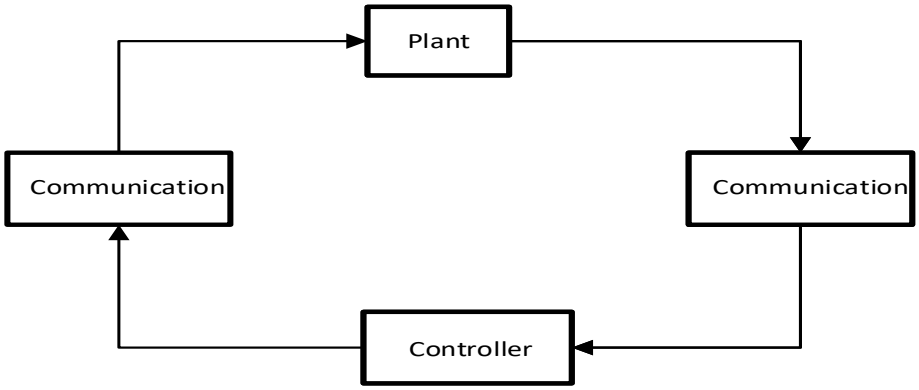
Chapter 3

Attack Detection Using Multi-Sine Watermarking

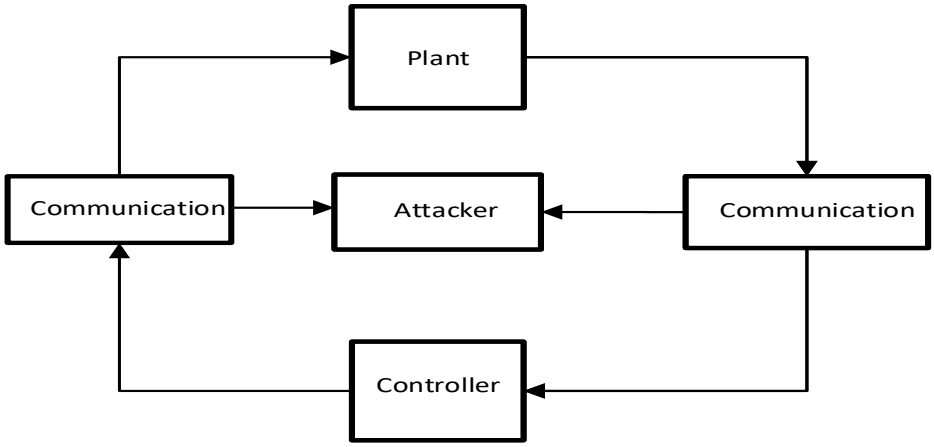
The objective of this thesis is to develop an approach for detecting replay attacks based on watermarking. In this chapter, we begin by introducing the problem and reviewing our assumptions. Next we present our proposed method, develop the design procedure for generating the watermarking signal and explain the process for detecting replay attacks. A case study will be presented in the next chapter to illustrate and assess the method.

3.1 Problem Statement

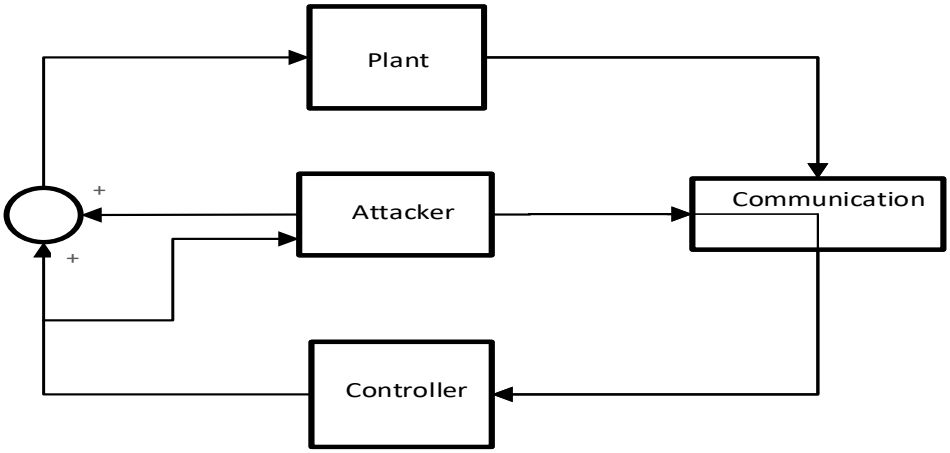
As described in Chapter 2, when replay attack occurs (Fig. 3.1 and Fig. 3.2), the attacker records the output of the system for interval δ, when the system is in steady state (phase 1). Then the attacker replaces the output data with the recorded and its repetitions. At the same, the attacker begins to alter the control signal (phase 2).



(a) Phase 0: Before attack happens

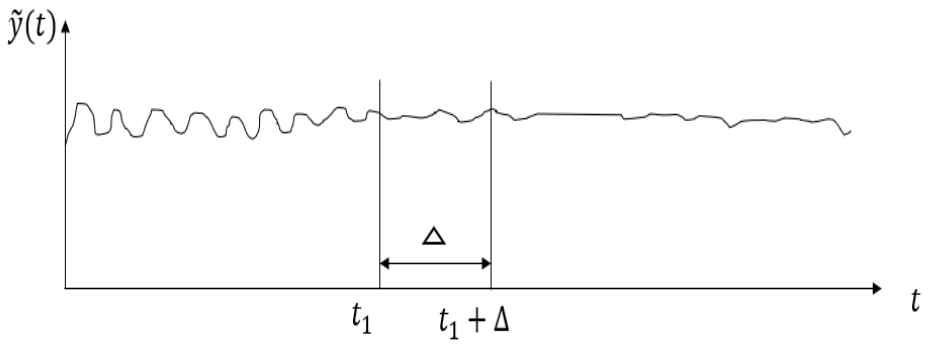


(b) Phase 1: Attacker reads the output data

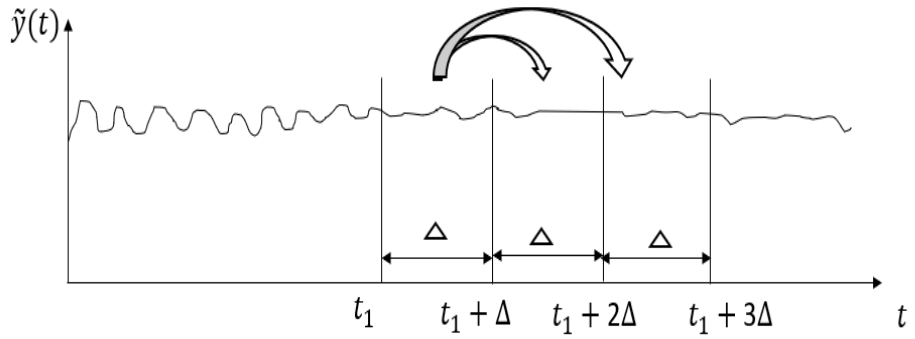


(c) Phase 2: Attacker replays the recorded output and alters the control sihnal

Figure 3.1: Different phases of replay attack



(a) Attacker records the output data



(b) Attacker replays the recorded output

Figure 3.2: Different steps of replay attack

As it is shown in Fig. 3.2, the attacker firstly records the data when the closed loop output reaches the steady state and then apply the attack. Here are some assumptions:

- The plant is single-input-single-output, possibly nonlinear. and under control in a feedback loop.
- The plant is subject to input and output noise.
- The closed-loop system has reached steady state and the plant operates around an operating point.

3.2 Proposed Solution

In this section, we propose and develop a method for detecting replay attack using watermarking. Watermarking starts when the closed loop system reaches steady state. Fig. 3.3 shows the linear model of the system around its operating point. Let U_0 and Y_0 denote the controller and the plant output around operating point.

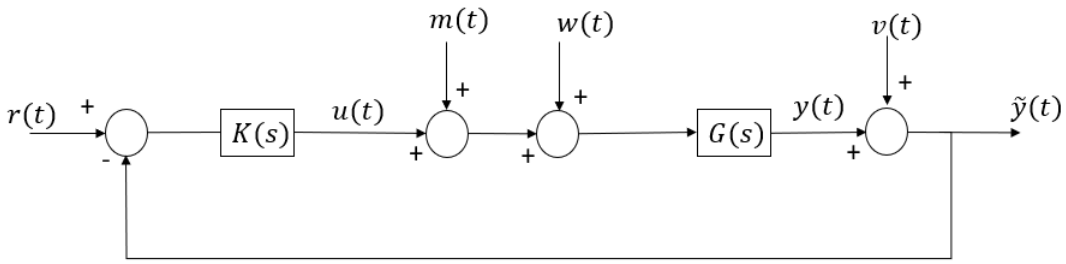


Figure 3.3: Model of a linear system

where $K(s)$ and $G_{my}(s)$ are the controller and plant transfer functions, $r(t)$, $m(t)$, $w(t)$ are reference input, watermarking signal and plant input disturbance. Furthermore, $v(t)$ is the output (sensor) noise. Thus $\tilde{y}(t)$ is the measured output. The watermarking signal, $m(t)$, is added, so that its effect can be traced in the plant. The absence of such effect in the measured output $\tilde{y}(t)$ can be an indication of a replay attack. The signal $m(t)$ must be chosen so that:

1. the effect of $m(t)$ in the output can be easily traced (despite the disturbance, noise, and replay attack)

2. an attacker cannot detect the watermarking signal fast enough to adjust the replay attack.

3. the plant output is not perturbed significantly and its fluctuation remains at an acceptable level.

As mentioned in Chapter 1, random IID signals have been proposed for watermarking in [10]. In this thesis, we propose to use sinusoidal signals. The effect of such a signal in the output of the plant will be a signal too. The power of a sinusoidal signal is concentrated in a narrow frequency

band which makes it easy to be detected. This helps with satisfying item 1 above. Another benefit of using sinusoidal signals compared with random IID signals is that the latter changes abruptly which increases the stress on actuators. A a sinusoidal input, however, changes smoothly. In order to make it difficult for an attacker to adjust its replay attack in response to watermarking (item 2 above), we change the frequency of sinusoidal signal. We refer to each time interval where the frequency of sinusoid is kept constant as a "**frame**". The length of a frame should be long enough so that the effect of watermarking can be detected by the controller. However, the frame length has to be so short that the attacker cannot detect the watermarking and adjust to it. This issue will be discussed later in this section. To address the third issue above (i.e limiting output fluctuations due to watermarking), we will show that using a suitable multi-sine signal can suppress the transient response of the plant due to watermarking. This is particularly useful in transition from one frame to the next frame. As a result, the output fluctuations due to watermarking will reduce to the steady state response. The amplitude of the steady state response can be easily computed analytically and adjusted. This is not the case with the random watermarking signal). Another benefit of our method is that (as will be seen) only the value of frequency response at the closed loop system at the watermarking frequencies are needed (which can be obtained experimentally). The complete model is not needed.

3.2.1 Transient Response Suppression

As mentioned before, in this thesis we propose the use of multi-sine signals for watermarking. The proposed watermarking consists of a sequence of multi-sine signals, each applied for an interval called a frame. In order to minimize the effect of watermarking on the plant output, we choose the multi-sine signals in such a way that they do not generate any transient response in plant output. The absence of transient response also helps with detecting the effect of watermarking in the plant output. Consider Fig. 3.3 and the transfer function from watermarking signal M(s) to plant output Y(s):

$$G_{my}(s) = \frac{Y(s)}{M(s)} = \frac{G(s)}{1 + K(s)G(s)}$$

The watermarking signal is of the form:

$$m(t) = \sum_{i=1}^{n_m} A_i \sin(\omega_i t + \phi_i)$$

We will show that, given a set of frequencies, the amplitudes A_i, phases ϕ_i and the number of sinusoidal signals, n_m, can be chosen such that in the output ($y(t)$), there is no transient response. We present the answer in the form of a solution for the following problem.

Problem: Given a stable system with a strictly proper transfer function $G_{my}(s)$ and initially, at rest, find a multi-sine input signal $m(t) = \sum_{i=1}^{n_m} A_i \sin(\omega_i t + \phi_i)$ with minimum number terms applied at $t = 0$, such that the output does not contain transient response.

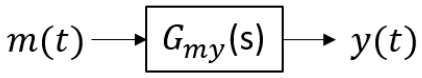


Figure 3.4: model of system from $m(t)$ to $y(t)$

We will present a closed-form solution for the first-order and second-order systems, following a time-domain approach. Next, we will provide a solution based on a frequency-domain approach which is not closed-form but is easier to use for higher order systems.

(a) Time-Domain Approach

(1) First order systems

Suppose $G_{my}(s)$ is a first-order system given by the differential equation:

$$\frac{dy}{dt} + ay(t) = bm(t) \tag{3.1}$$

where "a" and "b" are parameters. We will see that the minimum number of sinusoidal signals in this case is $n_m = 1$. Suppose $m(t) = A_1 \sin(\omega_1 t + \phi_1)$ $(t \geq 0)$. Then $y(t)$ will be

$$y(t) = k_1 e^{-at} + A_1 |G_{my}(jw_1)| \sin(\omega_1 t + \phi_1 + \measuredangle G_{my}(j\omega_1)) \tag{3.2}$$

The first and the second terms are transient and steady state response. respectively, $G_{my}(j\omega_1)$ is the transfer response at frequency ω_1, and $\measuredangle G_{my}(j\omega_1))$ is the phase of $G_{my}(j\omega)$ at ω_1. The transient response will be suppressed ($k_1 = 0$) if and only if the steady-state response satisfies the initial condition; that is $y_{ss}(0) = 0$. Thus, at $t = 0$

$$A_1|G_{my}(j\omega_1)|\sin(\phi_1 + \measuredangle G_{my}(j\omega_1)) = 0 \tag{3.3}$$

Assuming $G_{my}(j\omega_1) \neq 0$, we conclude:

$\phi + \measuredangle G_{my}(j\omega_1) = 2l\pi$ for $l = 0, 1, \cdots$.

In particular, for $l = 0$, we can choose $\phi = -\measuredangle G_{my}(jw_1)$. In this case A_1, w_1 can be chosen arbitrarily. Eq. (3.3) guarantees that there will be no transient part ($k_1 = 0$).

(2) Second order systems

Consider the second order system

$$\begin{aligned} \frac{d^2y}{dt^2} + k_1\frac{dy}{dt} + k_0y(t) &= b_1\frac{dm}{dt} + b_0m(t) \\ y(0) &= 0 \\ \frac{dy(0)}{dt} &= 0 \end{aligned} \tag{3.4}$$

The steady-state solution must satisfy the initial conditions (in order to have transient response suppressed). If $m(t)$ is a single sinusoid, $m(t) = A_1\sin(\omega_1(t) + \phi_1)$, it is easy to see that steady state response $A_1|G_{my}(j\omega_1)|\sin(\omega_1 t + \phi_1 + \measuredangle G_{my}(j\omega_1) = 0$ cannot satisfy both initial conditions. $y(0) = 0$ and $dy/dt(0) = 0$. Hence $m(t)$ must have at least 2 sinusoid signals. Let $m(t) = A_1 sin(w_1 t + \phi_1) + A_2 sin(w_2 t + \phi_2)$. Therefore

$$\begin{aligned} y_{ss}(t) = A_1|G_{my}(j\omega_1)|\sin(\omega_1 t + \phi_1 + \measuredangle G_{my}(j\omega_1)) + A_2|G_{my}(j\omega_2)|\sin(\omega_2 t + \phi_2 \\ + \measuredangle G_{my}(j\omega_2)) \end{aligned} \tag{3.5}$$

Without loss of generality, assume $w_2 > w_1$, and $A1 > 0$ and $A2 > 0$ y_{ss} must satisfy the initial conditions which yields Eq. (3.6)

$$\begin{cases} y_{ss}(0) = A_1|G_{my}(jw_1)|\sin(\phi_1 + \measuredangle G_{my}(j\omega_1)) + A_2|G_{my}(j\omega_2)|\sin(\phi_2 + \measuredangle G_{my}(j\omega_2)) = 0 \\ \dfrac{dy_{ss}(0)}{dt} = a_1\omega_1\cos\alpha_1 + a_2\omega_2\cos\alpha_2 = 0 \end{cases} \tag{3.6}$$

where $\alpha_1 = \phi_1 + \measuredangle G_{my}(j\omega_1)$, $\alpha_2 = \phi_2 + \measuredangle G_{my}(j\omega_2)$. Let $A_1|G_{my}(j\omega_1)| = a_1$, $A_2|G_{my}(j\omega_2)| = a_2$. Therefore

$$a_1 sin\alpha_1 + a_2 sin\alpha_2 = 0 \tag{3.7a}$$

$$a_1 w_1 cos\alpha_1 + a_2 w_2 cos\alpha_2 = 0 \tag{3.7b}$$

From Eq. (3.7) and assuming $a_1 \neq 0$ we get:

$$\sin\alpha_1 = -(a_2/a_1)sin\alpha_2 \tag{3.8}$$

Therefore

$$\begin{cases} cos\alpha_1 = \pm\sqrt{(1 - (a_2^2/a_1^2)\sin^2\alpha_2} \\ \qquad = \pm\sqrt{\dfrac{(a_1^2 - a_2^2\sin^2\alpha_2}{a_1^2}} \\ \qquad = \pm\dfrac{\sqrt{(a_1^2 - a_2^2 + a_2^2 cos^2\alpha_2}}{a_1} \end{cases} \tag{3.9}$$

Combination Eq. (3.7b) and Eq. (3.9) gives:

$$|w_1\sqrt{(a_1^2 - a_2^2 + a_2^2 cos^2\alpha_2)}| = |-a_2 w_2 cos\alpha_2| \tag{3.10}$$

Thus Case 1: $a_1 \neq a_2$. It follows from Eq. (3.10) that

$$w_1^2(a_1^2 - a_2^2 + a_2^2 cos^2\alpha_2) = a_2^2 w_2^2 cos^2\alpha_2 \tag{3.11a}$$

$$w_1^2(a_1^2 - a_2^2) = a_2^2(w_2^2 - w_1^2)cos^2\alpha_2 \tag{3.11b}$$

$$cos\alpha_2 = \pm(w_1/a_2)\sqrt{(a_1^2 - a_2^2)/(w_2^2 - w_1^2)} \tag{3.11c}$$

$$cos\alpha_2 = \pm\sqrt{((a_1/a_2)^2 - 1)/(w_2/w_1)^2 - 1} \tag{3.11d}$$

Therefore

$$(a_1/a_2)^2 \leq (w_2/w_1)^2 \tag{3.12}$$

Since $w_2 > w_1$, we can conclude:

$$a_2 < a_1 \tag{3.13}$$

In summary, the solution in this case is given by Eq. (3.13), Eq. (3.11d) and Eq. (3.8).

Case 2: if $a_1 = a_2$

It follows from Eq. (3.7a) that $\sin\alpha_1 = -\sin\alpha_2$, which results in $\alpha_1 = -\alpha_2$, or $\alpha_1 = \pi + \alpha_2$. For the first case, $\cos\alpha_1 = \cos\alpha_2$, and using Eq. (3.7b), $(a_1w_1 + a_2w_2)\cos\alpha_1 = 0$. This implies $\cos\alpha_1 = 0$.

$$\alpha_1 = \frac{\pi}{2}, \quad \alpha_2 = -\frac{\pi}{2} \tag{3.14a}$$

$$\text{or} \quad \alpha_1 = -\frac{\pi}{2}, \quad \alpha_2 = \frac{\pi}{2} \tag{3.14b}$$

For the second case, $\alpha_1 = \pi + \alpha_2$, $\cos\alpha_1 = -\cos\alpha_2$, so $(a_1w_1 - a_2w_2)\cos\alpha_1 = 0$, thus $\cos\alpha_1 = 0$ (since $a_1w_1 - a_2w_2 = a_1(w_1 - a_2w_2) \neq 0$.) Therefore, in this case, $\alpha_1 = \frac{\pi}{2}$, $\alpha_2 = -\frac{\pi}{2}$ or $\alpha_1 = -\frac{\pi}{2}$, $\alpha_2 = -\frac{3\pi}{2}$ (same as the first case)

(3) Third order systems

For the third-order system, we show 2 $\sin$ signals is enough to suppress the transient part. For this, the steady state response must satisfy $y(0) = 0$, $\frac{dy(0)}{dt} = 0$ and $\frac{d^2y(0)}{dt^2} = 0$. This

results in the following:

$$a_1 \sin\alpha_1 + a_2 \sin\alpha_2 = 0 \tag{3.15a}$$

$$a_1\omega_1 \cos\alpha_1 + a_2\omega_2 \cos\alpha_2 = 0 \tag{3.15b}$$

$$a_1\omega_1^2 \sin\alpha_1 + a_2\omega_2^2 \cos\alpha_2 = 0 \tag{3.15c}$$

Here, a_i and d_i are defined similar to the case of second-order systems. From Eq. (3.15a) and Eq. (3.15c) we can conclude:

$$\begin{cases} \sin\alpha_1 = -\dfrac{a_2}{a_1}\sin\alpha_2 \\ \sin\alpha_1 = -\dfrac{a_2\omega_2^2}{a_1\omega_1^2}\sin\alpha_2 \end{cases} \tag{3.16}$$

For satisfying Eq. (3.16), $\sin\alpha_1$,$\sin\alpha_2 \neq 0$, we can conclude $\omega_1 = \omega 2$ which is impossible by assumption. Therefore it must be the case that $\sin\alpha_1 = 0$ and $\sin\alpha_2 = 0$. Hence Hence,

$$\begin{cases} \alpha_1 = k_1\pi \quad k_1 : integer \\ \alpha_2 = k_2\pi \quad k_2 : integer \end{cases} \tag{3.17}$$

We only need to consider four cases $\alpha_i = 0, \pi \quad (i = 1, 2)$ which

Case (1): α_1=$\alpha_2 = 0$, which is not acceptable since it violates Eq. (3.15b).

Case (2): α_1=$\alpha_2 = \pi$, which is not acceptable since it violates Eq. (3.15b).

Case (3): $\alpha_1 = 0$, $\alpha_2 = \pi$, which from Eq. (3.15) results in $\frac{a_1}{a_2} = \frac{\omega_2}{\omega_1}$.

Case (4): $\alpha_1 = \pi$, $\alpha_2 = 0$, which again from Eq. (3.15) results in $\frac{a_1}{a_2} = \frac{\omega_2}{\omega_1}$.

(b) Frequency-Domain Approach

Suppose $G_{my}(s)$ is an n-th order system described by differential equation

$$\frac{d^n y}{dt^n} + a_{n-1}\frac{d^{n-1}y}{dt^{n-1}} + \cdots + a_0 y(t) = b_{n-1}\frac{d^{n-1}m}{dt^{n-1}} + \cdots + b_0 m(t) \tag{3.18}$$

Thus

$$G_{my}(s) = \frac{b(s)}{a(s)} = \frac{b_{n-1}s^{n-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} \tag{3.19}$$

The watermarking signal is a multi-sine signal $m(t) = \sum_{i=1}^{n_m} A_i \sin(\omega_i t + \phi_i)\ \ (t > 0)$ and therefore:

$$M(s) = \frac{p_m(s)}{(s^2 + \omega_1^2) \cdots (s^2 + \omega_{n_m}^2)} \tag{3.20}$$

where $p_m(s)$ is a polynomial of degree $2n_m - 1$ or lower, as we have:

$$0 \leq deg(p_m(s)) \leq 2n_m - 1 \tag{3.21}$$

The response of $G_{my}(s)$ to input $M(s)$ is:

$$\begin{aligned} y(s) =& G_{my}(s)M(s) \\ =& \frac{b(s)p_m(s)}{a(s)(s^2 + \omega_1^2) \cdots (s^2 + \omega_{n_m}^2)} \end{aligned} \tag{3.22}$$

A necessary and sufficient condition to suppress the transient response (which is due to the poles of $G_{my}(s)$) is that the poles of $G_{my}(s)$ can be canceled by zeros of $M(s)$; in other words, $p_m(s)$ must be chosen as

$$p_m(s) = c(s)a(s) \tag{3.23}$$

For some polynomial $c(s)$. this implies that:

$$deg(a(s)) \leq deg(p_m(s)) \leq 2n_m - 1 \tag{3.24}$$

Thus

$$n_m \geq \frac{n+1}{2} \tag{3.25}$$

Therefore the minimum number of sinusoidal signals in $m(t)$ is:

$$n_{m,min} = \lceil \frac{n+1}{2} \rceil \tag{3.26}$$

where $\lceil \quad \rceil$ is the ceiling function. Once n_m is chosen based on Eq. (3.25), $c(s)$ in Eq. (3.23) is chosen so that Eq. (3.21) is satisfied. The watermarking signal $M(s)$ is obtained from Eq. (3.20) and $m(t)$ is obtained using partial fraction expansion.

Example: Second order system. Suppose $n = 2$, and

$$G_{my}(s) = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0} = \frac{b(s)}{a(s)} \tag{3.27}$$

From Eq. (3.23) it follows that:

$$n_m \geq \frac{n+1}{2} = \frac{3}{2} \tag{3.28}$$

So, the smallest n_m is 2 and

$$m(t) = A_1 \sin(\omega_1 t + \phi_1) + A_2 \sin(\omega_2 t + \phi_2) \quad (t > 0) \tag{3.29}$$

From Eq. (3.21), $deg(p_m(s)) \leq 2n_m - 1 = 3$. i.e, $p_m(s)$ is at most third-order. It follows from Eq. (3.23) that:

$$\begin{cases} p_m(s) = c(s)a(s) \\ \qquad\quad = (c_1 s + c_0)(s^2 + a_1 s + a_0) \end{cases}$$

Any choice of c_1 and c_0 results in suppression of transient response. Of course the trivial case of $c_1 = c_0 = 0$ should be excluded. The values of c_1 and c_0 determine the amplitude of steady state response. In summary:

$$
\begin{aligned}
M(s) =& \frac{(c_1 s + c_0)(s^2 + a_1 s + a_0)}{(s^2 + \omega_1^2)(s^2 + \omega_2^2)} \\
Y(s) =& \frac{b(s) p_m(s)}{a(s)(s^2 + \omega_1^2)(s^2 + \omega_2^2)} \\
=& \frac{(b_1 s + b_0)(c_1 s + c_0)}{(s^2 + \omega_1^2)(s^2 + \omega_2^2)}
\end{aligned}
$$

3.2.2 Amplitude of Sine Waves

The multi-sine watermarking signal results in fluctuations in plant output given by

$$
\sum_{i=1}^{n_m} A_i |G_{my}(jw_i)| \sin(\omega_i t + \phi_i + \measuredangle G(jw_i)) \tag{3.30}
$$

The amplitude A_i should be chosen so that

(i) The output perturbations are small enough that do not degrade the quality of output regulation, and

(ii) The output perturbations are large enough to be detected and distinguished from noise. The issue detection, it will be described in detail in section (3.2.4).

Let $\alpha = \sum_{i=1}^{n_m} A_i |G_{my}(jw_i)|$, as upper bound of amplitude of sine signal in output, β=sensor accuracy, δ denotes the maximum acceptable output fluctuations (due to watermarking, noise, disturbance.) Suppose δ can be defined as $\delta = \delta_m + \delta_d$ in which δ_m is the maximum fluctuation due to watermarking, and δ_d pertains to the rest (noise, etc.). To meet (i) and (ii), we require that $\beta < \alpha < \delta_m$. Furthermore, the power of watermarking signal should be sufficiently high so that the corresponding effects can be detected in the presence of noise and other disturbances. The ratio of the power of the output fluctuations due to watermarking to that of noise and disturbance is represented as SNR (Signal to Noise Ratio):

$$
SNR = \frac{\frac{1}{2} \sum_{i=1}^{n_m} A_i^2 |G_{my}(jw_i)|^2}{(\sigma_v^2 + \sigma_w^2 \frac{1}{2\pi} \int_{-\infty}^{\infty} |G_{my}(jw)|^2 dw)} \tag{3.31}
$$

The SNR can be used to compare the effects of watermarking and disturbance. Here, σ_w and σ_v are the variances of input and output disturbance signals. If the plant input and output have the same units (as in the case in our case study in chapter 4), we can simplify calculations by using:

$$\eta = \frac{\frac{1}{2}\sum_{i=1}^{n_m} A_i^2 |G_{my}(jw_i)|^2}{\sigma_v^2 + \sigma_w^2} \tag{3.32}$$

A_i's should be chosen to result in smallest η that permits the detection of watermarking effects.

3.2.3 Frequencies of Sine Waves and Frame Size

Let us consider the watermarking signal over a frame:

$$m(t) = \sum_{i=1}^{n_m} A_i \sin(\omega_i t + \phi_i)$$

Here t is measured with respect to the start of the frame. Let $f_i = \frac{\omega_i}{2\pi}$ and $T_i = \frac{1}{f_i}$ denote the frequency in (Hz), and period of each component respectively. To simplify the design, we choose frequencies so that for some integers $n_1, n_2, \cdots, n_{n_m}$

$$\frac{f_1}{n_1} = \frac{f_2}{n_2} = \cdots = \frac{f_{n_m}}{n_{n_m}}$$

We assume integers are relatively prime (i.e. gcd$((n_1, n_2, \cdots, n_{n_m}) = 1$). This ensures that $m(t)$ is a periodic signal with period $T_{combined} = n_1 T_1 = n_2 T_2 = \cdots = n_{n_m} T_{n_m}$. The size of each frame for watermarking is chosen to be a multiple of $T_{combined}$. This will ensure that at the end of frame, when $m(t)$ is cut, no transient response is generated. Without loss of generality assume that f_i's are in increasing order: $f_1 \leq f_2 \leq \cdots \leq f_{n_m}$. Also let T_f denote the frame size for signal $m(t)$. As mentioned before, T_f is chosen to be a multiple of $T_{combined}$: $T_f = kT_{combined}$ for some positive integer k. As will be explained, we will use the periodogram of plant output (as an estimate of power spectral density, PSD) to detect the presence of sinusoids in the watermarking signal. We

have to be able to distinguish f_i's. The frequency resolution of PSD is:

$$\Delta f = \frac{1}{T_f}$$

Thus, for two frequencies f_1 and f_2 we must have:

$$\begin{aligned} T_f \geq & \frac{1}{f_2 - f_1} \\ = & \frac{1}{f_1(\frac{n_2}{n_1} - 1)} \quad \text{since} \quad \frac{n_2}{n_1} = \frac{f_2}{f_1} \\ = & \frac{T_{combined}}{n_2 - n_1} \quad \text{since} \quad T_{combined} = \frac{n_1}{f_1} \end{aligned}$$

Since $f_1 < f_2$ and $n_1 < n_2$, the above condition is satisfied since T_f is chosen as a multiple of P (i.e. $T_f = kT_{combined} \geq P \geq \frac{T_{combined}}{n_2 - n_1}$). In the special case of $n_2 - n_1 = 1$, it is better to avoid the borderline case of $T_f = T_{combined}$ and choose $T_f \geq 2T_{combined}$. In order to be able to choose $T_f = P$, it is better to choose f_1, f_2 such that $n_2 - n_1 \neq 1$, for example $\frac{n_2}{n_1} = \frac{5}{3}$ or $\frac{3}{1}$. Choosing low values for frequencies f_i results in large combined period $T_{combined}$ of watermarking signal and large frame size T_f. This will provide more time to the attacker to detect watermarking and probably adjust to it. On the other hand, increasing the frequencies will result in small η at large frequencies f_i's $|G_{my}(jw)|$ is smaller, unless large watermarking signal (A_i) is used. Another drawback of large frequencies is that due to modeling uncertainty at high frequencies, frequency response values are less accurate at high frequencies. This results in lower accuracy in design calculations for watermarking.

3.2.4 Detection of Watermarking Signal

The multi-sine watermarking signal results in fluctuations in plant output given by Eq. (3.30). The set of frequencies $(\omega_1, \cdots, \omega_{n_m})$ is changed from frame to frame. Suppose that the closed-loop system is in steady state, operating at a setpoint (i.e. the reference input, $r(t)$, in Fig (3.3) is a constant). Then to detect the signal of Eq. (3.30) and distinguish it from output disturbance,

one could record the output over the corresponding frame and determine its PSD to confirm the presence of sine signals at frequencies $\omega_1, \cdots, \omega_{n_m}$. In this thesis, periodogram is used to estimate PSD. There are also other methods such as modified periodogram, and parametric methods for estimating PSD. A detailed analysis of these methods in order to determine the most suitable one for watermarking application is left for future research.

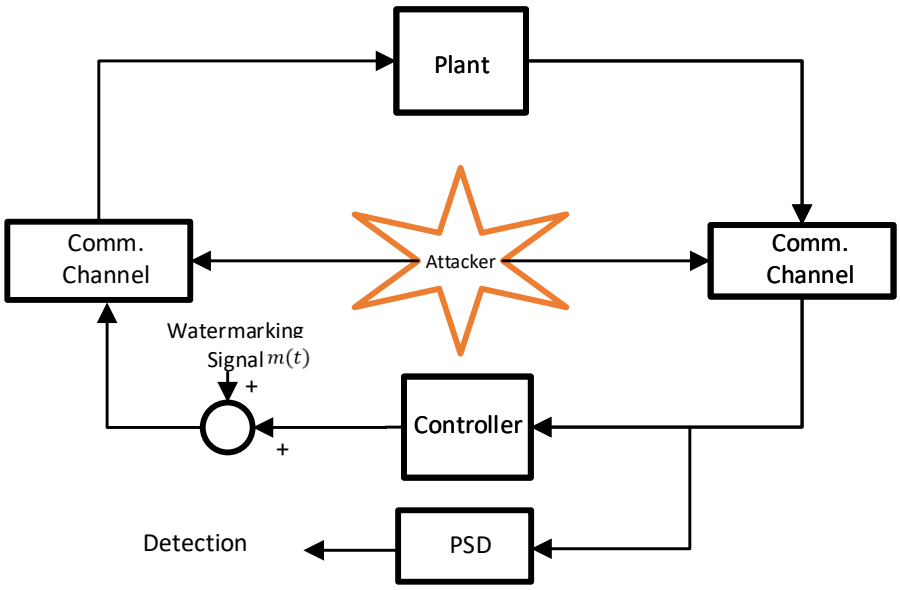


Figure 3.5: Model of control system with PSD detector

If the reference input r(t) is not a setpoint and varies with time, then the plant output will change accordingly and the PSD of output has power in frequencies other than the watermarking frequencies. In such a case, frequencies $\omega_1, \cdots, \omega_{n_m}$ may not easily be seen unless the amplitudes $A_i|G_{my}(jw_i)|$ are sufficiently large. Large fluctuations due to watermarking are not desirable. In this case, an alternative solution would be to use a Kalman Filter and monitor its residual signal.

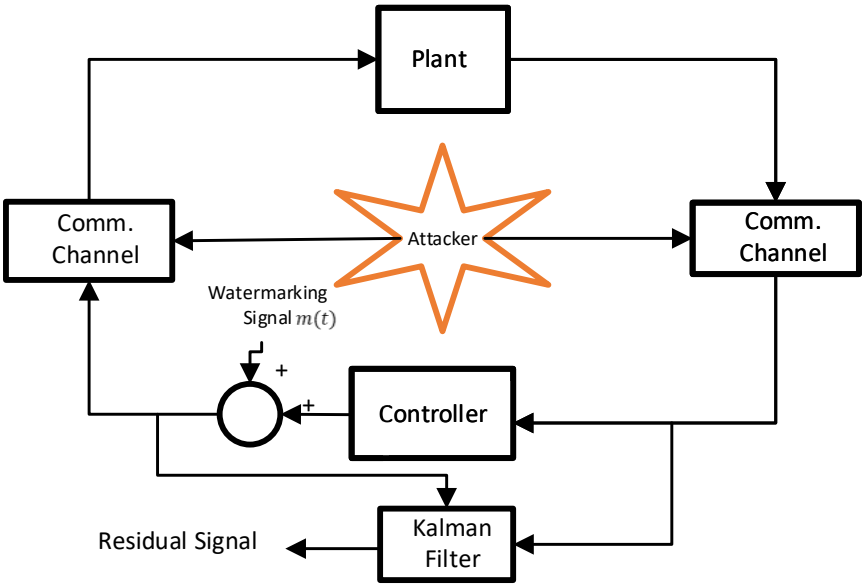


Figure 3.6: Model of control system with Kalman filter detector

In this setup, in the absence of attack (and faults), the residual signal only contain noise. In the presence of replay attack, the residual signal will include the effects of watermarking signal. The frequencies of watermarking signal can be detected in the residual signal using a PSD estimator. The detection of the frequencies would also indicate that changes in residual signal is not because of a fault.

3.2.5 Conclusion

In this chapter, we presented a method for detecting replay attacks. The method is based on watermarking using multi-sine signals. A method for suppression of transient response in plant output was discussed and choosing frequencies, amplitudes and frame sizes were discussed. In Chapter 4, a model of a laboratory tank is presented and a detailed analysis of the application of the proposed method is provided.

Chapter 4

Replay Attack Detection in a Tank System

In this chapter, we apply the watermarking procedure for replay attack detection described in Chapter 3 to a tank system. We begin by introducing a nonlinear model of the tank system and its feedback control system. Next following the proposed process in Chapter 3, we design a watermarking signal. Finally, we present the simulation results and discuss various design aspects and their impact on the final results.

4.1 Plant Model

The plant is a single water tank used in a flow control system (Fig. 4.1). The parameters are chosen according to those provided in [54]. From mass balance,

$$A\frac{dH}{dt} = Q_1 - Q_2 \tag{4.1}$$

where A is the tank area, Q_1 and Q_2 are input and output volum flow, and H is water level. The output flow is

$$Q_2 = a_z S(2gH)^{1/2} \tag{4.2}$$

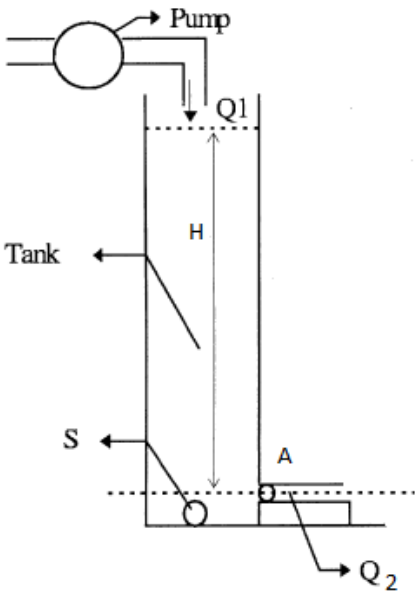


Figure 4.1: Schematic diagram of tank system

where $a_z = 0.45$, outflow coefficient (correcting factor, dimensionless)

$S = 5 \times 10^{-}5 \quad m^2$, the cross sectional area of output pipe

$A = 0.0154 \quad m^2$ is the area of tank, and $g = 9.81 \ \ m/s^2$ is gravity

With the above numerical values, the nonlinear equation of the tank becomes:

$$\begin{cases} \dfrac{dH}{dt} = -6.49 \times 10^{-3}\sqrt{H} + 64.9Q_1 \\ Q_2 = 0.997 \times 10^{-4}\sqrt{H} \end{cases} \tag{4.3}$$

The tank is part of a flow control system in which the output flow (Q_2) is measured and regulated by adjusting input flow (Q_1). The operating point value of output flow is chosen here to be:

$$\begin{aligned} Q_{2_o} &= 5.46 \times 10^{-5} \quad m^3/s \\ &= 54.6 \quad ml/s \end{aligned}$$

This corresponds to the water level of:

$$H_o = \frac{1}{2g}(\frac{Q_{2_o}}{a_z S})^2 = 0.3 \quad m$$

Suppose $h(t), q_1(t)$ and $q_2(t)$ denotes deviations from operating point values:

$$\begin{aligned} h =& H - H_o \\ q_1 =& Q_1 - Q_{1_o} \\ q_2 =& Q_2 - Q_{2_o} \end{aligned}$$

The linearized model around the operating point will be:

$$\begin{cases} \frac{dh}{dt} =& -\frac{\alpha_z S}{A}\sqrt{\frac{g}{2H_o}}h(t) + \frac{1}{A}q_1(t) \\ q_2 =& \alpha_z S\sqrt{\frac{g}{2H_o}}h(t) \end{cases} \tag{4.4}$$

Let $w(t)$ and $v(t)$ denote input and output disturbance, both assumed to be Gaussian white noise, with zero mean and variances $\sigma_\omega^2 = 2 \times 10^{-14}(m^3/s)^2$ and $\sigma_v^2 = 8.25 \times 10^{-15}(m^3/s)^2$. These correspond to standard deviations of $0.14 \ \ ml/s$ and $0.09 \ \ ml/s$.

After substituting in Eq. (4.4), with parameter values and adding disturbance we get:

$$\begin{cases} \frac{dh}{dt} =& -5.9 \times 10^{-3}h(t) + 64.9q_1(t) + 64.9w(t) \\ q_2(t) =& 9.1 \times 10^{-5}h(t) + v(t) \end{cases} \tag{4.5}$$

The transfer function of tank will be $G = \frac{5.88\times10^{-3}}{(s+6\times10^{-3})}$ or $G = \frac{q_2(s)}{q_1(s)} = \frac{1}{(169s+1)}$. Next, a PI controller $\frac{(5.5s+0.1)}{s}$ is designed and step response characteristics for closed loop system will be derived as:

Rise Time: 41.48 s

Settling Time: 206.46 s

Overshoot: 11.85 %

The closed loop poles are located at $-0.0192 \pm j0.0149$. The block diagram of the feedback system is given in Fig. 4.2. The linear model in Simulink is given in Fig. 4.3 and Fig. 4.4.

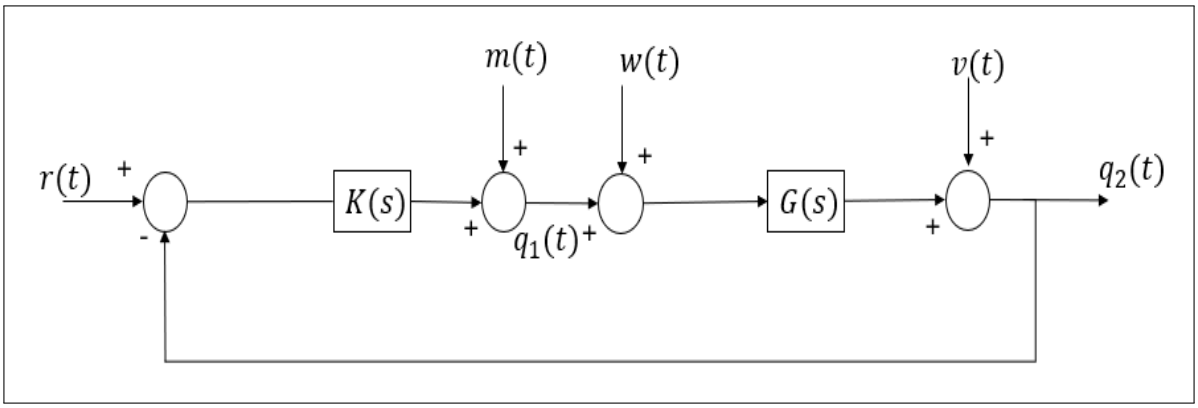


Figure 4.2: Flow feedback control system

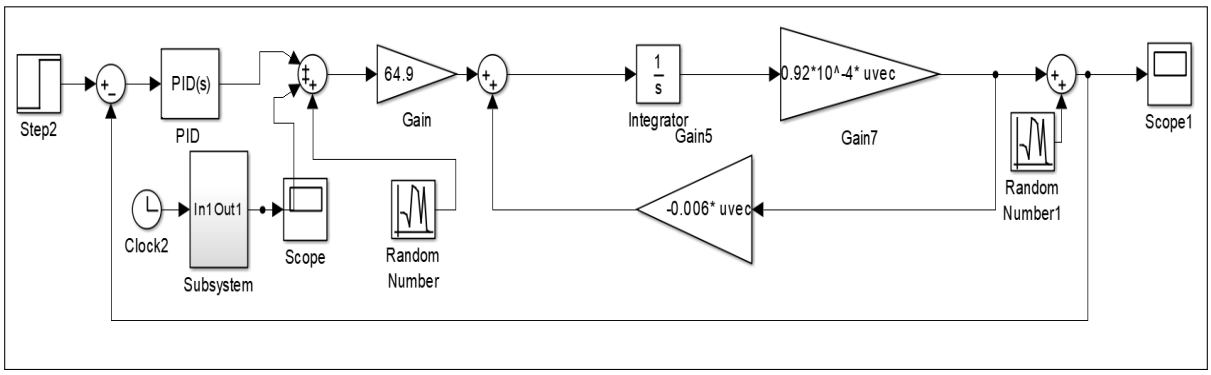


Figure 4.3: Linearized model of system in Simulink

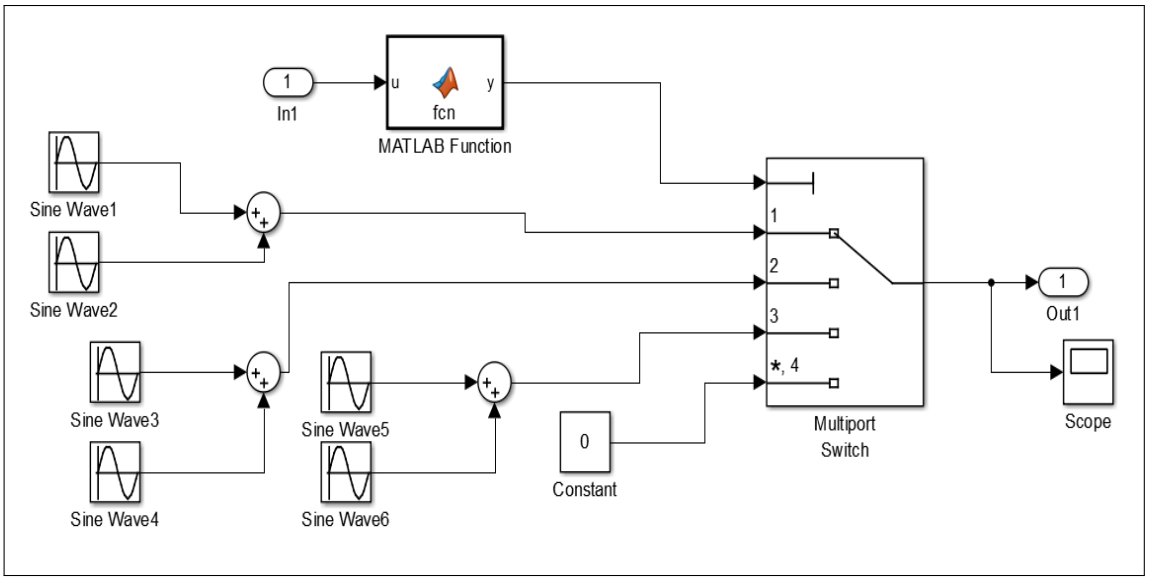


Figure 4.4: Sine wave section in Simulink

4.2 Watermarking Signal

The closed-loop transfer functions

$$G_{rq_2}(s) = \frac{q_2(s)}{r(s)}, \quad G_{mq_2}(s) = \frac{q_2(s)}{m(s)} \tag{4.6}$$

are both second order transfer functions.

Since G_{mq_2} is second order, following the discussion in sec. (3.2.1), the multi-sine signal $m(t)$ must include at least two frequencies, so that transient responses can be suppressed. Let

$$m(t) = A_1 \sin(\omega_1 t + \phi_1) + A_2 \sin(\omega_2 t + \phi_2)$$

The frequency responses of $G_{rq_2}(j\omega)$ and $G_{mq_2}(j\omega)$ are plotted in Fig. 4.5.

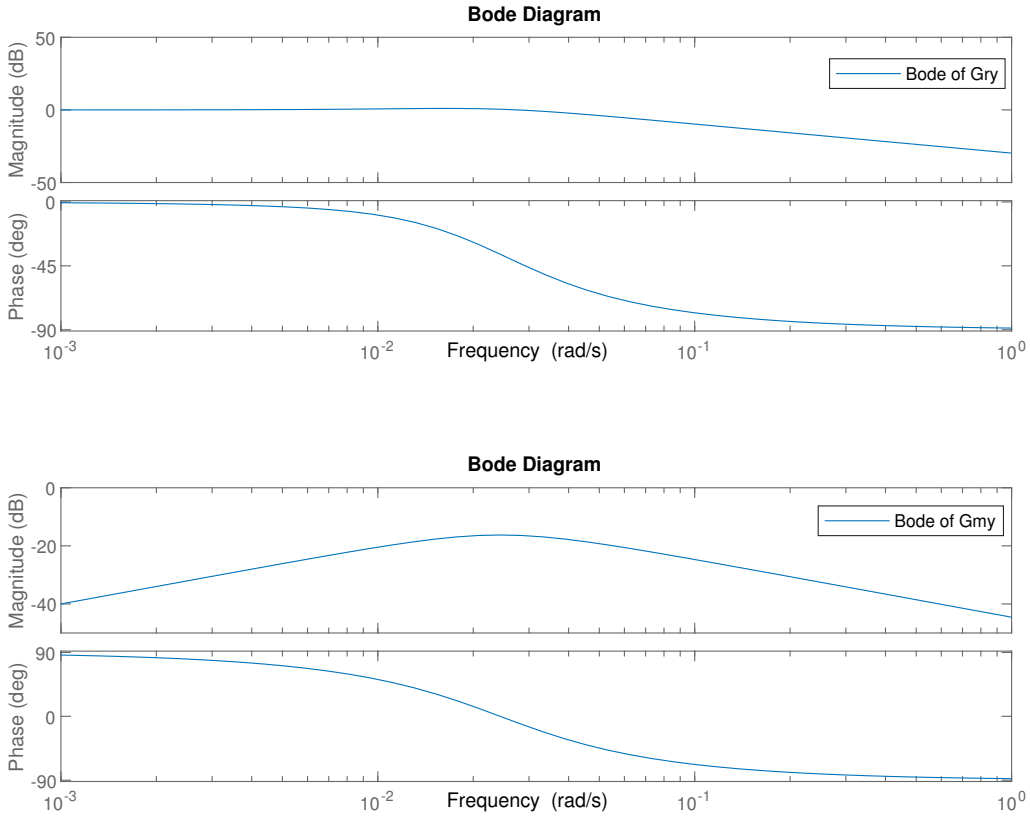


Figure 4.5: Bode diagrams of G_{rq_2} and G_{mq_2}

The frequencies are chosen from frequencies in which $G_{mq_2}(j\omega)$ has relatively high values: $0.01 \leq \omega \leq 0.6$. For the purpose of this study, three frames are considered:

Frame 1:

$$w_1 = 0.01, \quad w_2 = 0.03, \quad \frac{n_2}{n_1} = 3 \implies T_{combined} = T_1 = 3 \times T_2 = 628 \ \ s$$

Frame 2:

$$w_3 = 0.07, \quad w_4 = 0.1167, \quad \frac{n_4}{n_3} = \frac{5}{3} \implies T_{combined} = 3 \times T_3 = 269 \ \ s$$

Frame 3:

$$w_5 = 0.2, \quad w_6 = 0.6, \quad \frac{n_6}{n_5} = 3 \implies T_{combined} = T_5 = 3 \times T_6 = 31.4 \ \ s$$

For frame 1, following section (3.2.1) we choose:

$\phi_1 = \frac{\pi}{2} - \measuredangle G_{mq_2}(jw_1)$ and $\phi_2 = -\frac{\pi}{2} - \measuredangle G_{mq_2}(jw_2)$

Following the discussion in Section 3.2.2, an upper bound for output fluctuations due to watermarking is:

$$\begin{aligned}\alpha &= A_1|G_{mq_2}(j\omega_1)| + A_2|G_{mq_2}(j\omega_2)| \\ &= 2A_1|G_{mq_2}(j\omega_1)| \\ &= 2A_2|G_{mq_2}(j\omega_2)|\end{aligned}$$

Amplitude A_1 and A_2 are chosen so that $A_1|G_{mq_2}(jw_1)| = A_2|G_{mq_2}(jw_2)|$. Maximum permissible fluctuation of this laboratory tank is due to watermarking in water level is assumed to be $2\ cm$ or 6.7% of operating point. Based on Eq. (4.4), max fluctuation of output flow (when $h = 0.02\ m$) is $\delta_m = 1.8\ ml/s$ or 3.3% of operating point value (5.4×10^{-5}). We choose A_1 and A_2, so that $\alpha < \delta_m$ or $\alpha < 1.8 \times 10^{-6}$. The amount of α in each frame is fix.

We have:

$$|G_{mq_2}(j0.01)| = 0.095$$

Thus

$$2A_1 \times 0.095 < 1.8 \times 10^{-6}$$

or

$$A_1 < 9.5 \times 10^{-6}$$

Once A_1 is chosen, A_2 is determined from:

$$A_2 = \frac{|G_{mq_2}(j0.01)|A_1}{|G_{mq_2}(j0.03)|}$$

This in turn, determines the η factor

$$\eta = \frac{\frac{1}{2}A_1^2|G_{mq_2}^2(jw_1)| + \frac{1}{2}A_2^2|G_{mq_2}^2(jw_2)|}{\sigma_v^2 + \sigma_w^2}$$

The values of sine wave amplitudes for the other two frames are found similarly. In the following, we will examine watermarking and its detection for three different choices of η.

4.3 Detection with Periodogram

4.3.1 Case 1: $\alpha = 6.72 \times 10^{-8}$ and $\eta = 0.04$, linear system

First, we choose $\alpha = 6.72 \times 10^{-8}$, which means $A_1|G_{mq_2(jw_1)}| = 3.36 \times 10^{-8}$ or $A_1 = 3.54 \times 10^{-7}\ m^3/s$. The value of A_2 and the amplitude for sine waves for frame 2 and 3 are computed similarly (based on the same value of α). Firstly, we apply just sine wave, without reference input, $r(t) = 0$ and without noise, and then the effect of noise and reference input in further is considered. Firstly, just in frame 1 the output of applying phasing and no phasing signals are compared to each other, and then it is expanded to 3 frames. The Fig. 4.6 shows just for frame 1 of applying sine signals. In Fig. 4.7 and Fig. 4.8 for one and two combined period(s), the outputs are shown and as can be seen, the curves which phasing are applied to, do not have any transient part compared to the curve which ϕ has not been applied. These tests are applied on linear system based on Eq. (4.4) and in the following figures the results are presented. The sinusoids "with phasing" are as following:

$3.548 \times 10^{-7} \sin(0.01t + 0.66) + 2.26 \times 10^{-7} \sin(0.03t - 1.3)$

$4.128 \times 10^{-7} \sin(0.07t + 2.58) + 6.72 \times 10^{-7} \sin(0.11t - 0.33)$

$1.14 \times 10^{-6} \sin(0.2t + 2.94) + 3.41 \times 10^{-6} \sin(0.6t - 0.06)$

and the sine signals "without phasing" are:

$3.548 \times 10^{-7} \sin(0.01t) + 2.26 \times 10^{-7} \sin(0.03t)$

$4.128 \times 10^{-7} \sin(0.07t) + 6.72 \times 10^{-7} \sin(0.11t)$

$1.14 \times 10^{-6} \sin(0.2t) + 3.41 \times 10^{-6} \sin(0.6t)$

For each frame, t is the time from the start of the corresponding frame. The watermarking signal and in particular, the phase shifts ϕ_1 and ϕ_2 have been chosen to suppress the transient response.

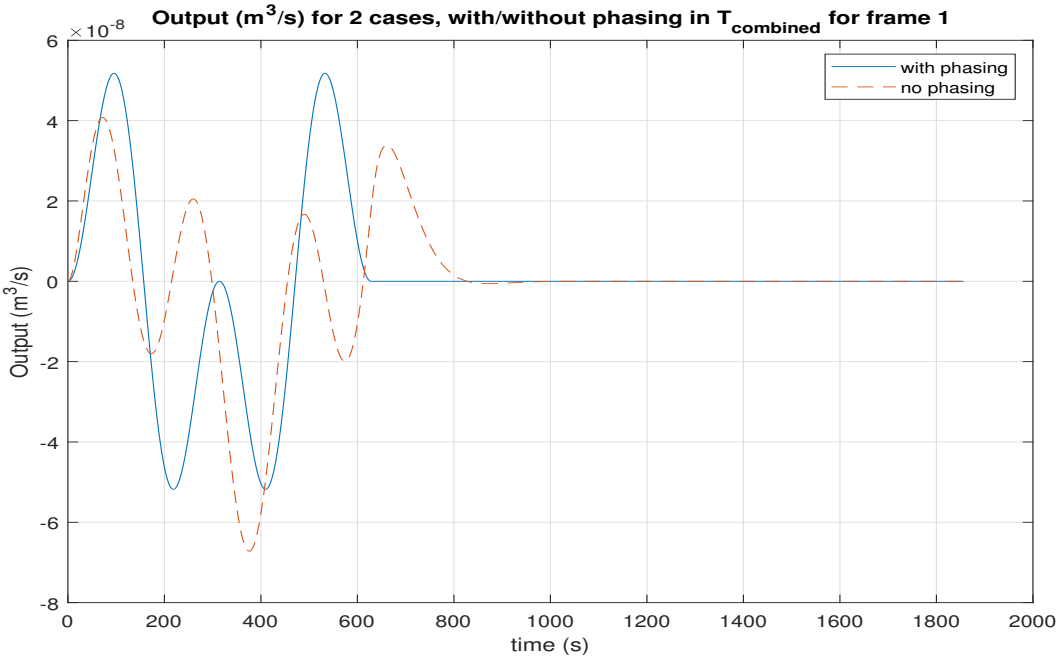


Figure 4.6: Output of system, watermarking for one frame, $T_f = T_{combined}$ of sine signal, without noise

Fig. 4.6 shows that for input with phasing, output does not have transient, after t= 629 s, there is trivial fluctuation in output which is absent in the case of watermarking signal with phasing.

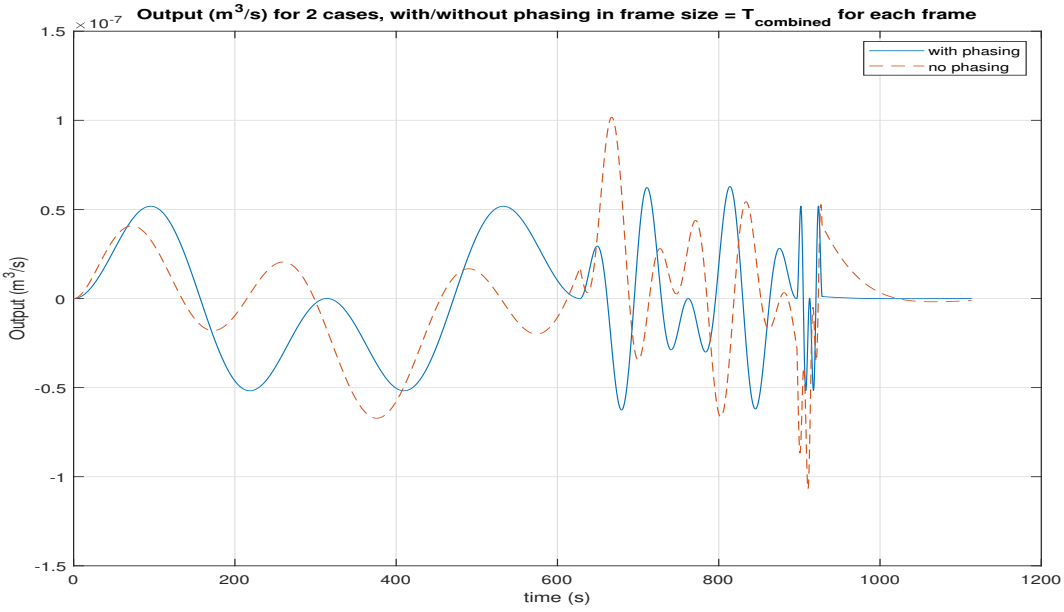


Figure 4.7: Output of system, in frame size $T_f = T_{combined}$, 3 frames, without noise

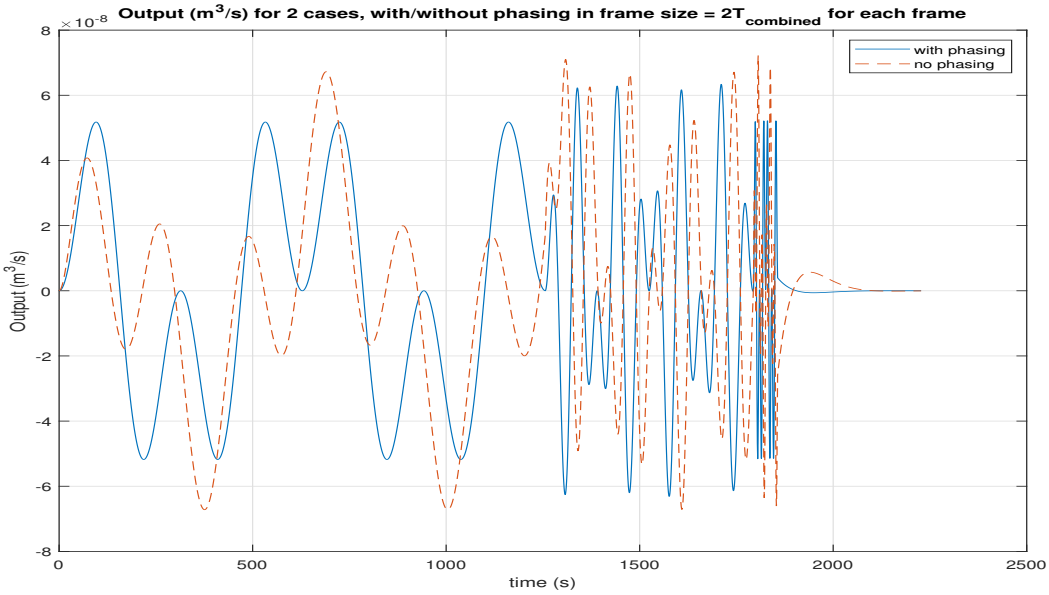


Figure 4.8: Output of system in frame size $T_f = 2\ T_{combined}$ in 3 frames, without noise

periodograms: As mentioned in previous chapter, one way for controller to track the effects of watermarking in the plant output is through the output signal's periodogram. Specifically, the controller can obtain the periodogram of output segments corresponding to each frame. The result of our example are provided in Fig. 4.9 for the case $T_f = T_{combined}$ and Fig. 4.10 for the case $T_f = 2T_{combined}$.

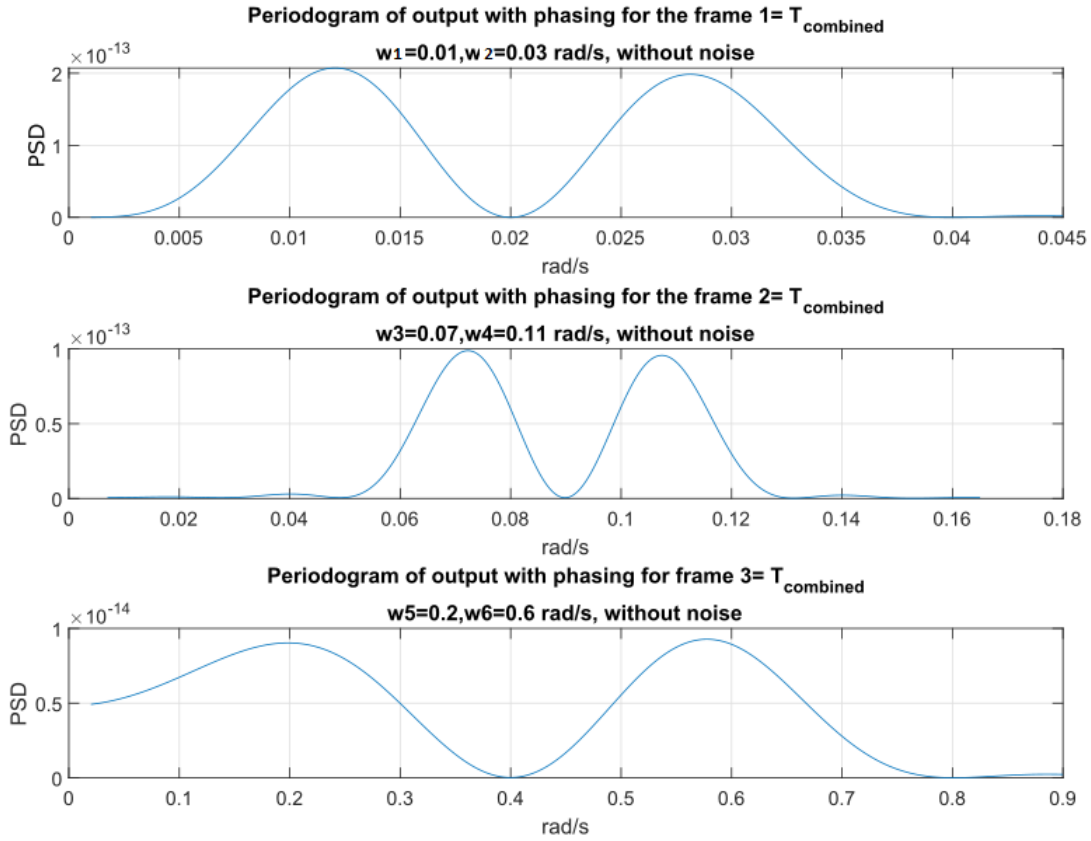


Figure 4.9: The periodogram of 3 frames, each frame $T_f = T_{combined}$, without noise

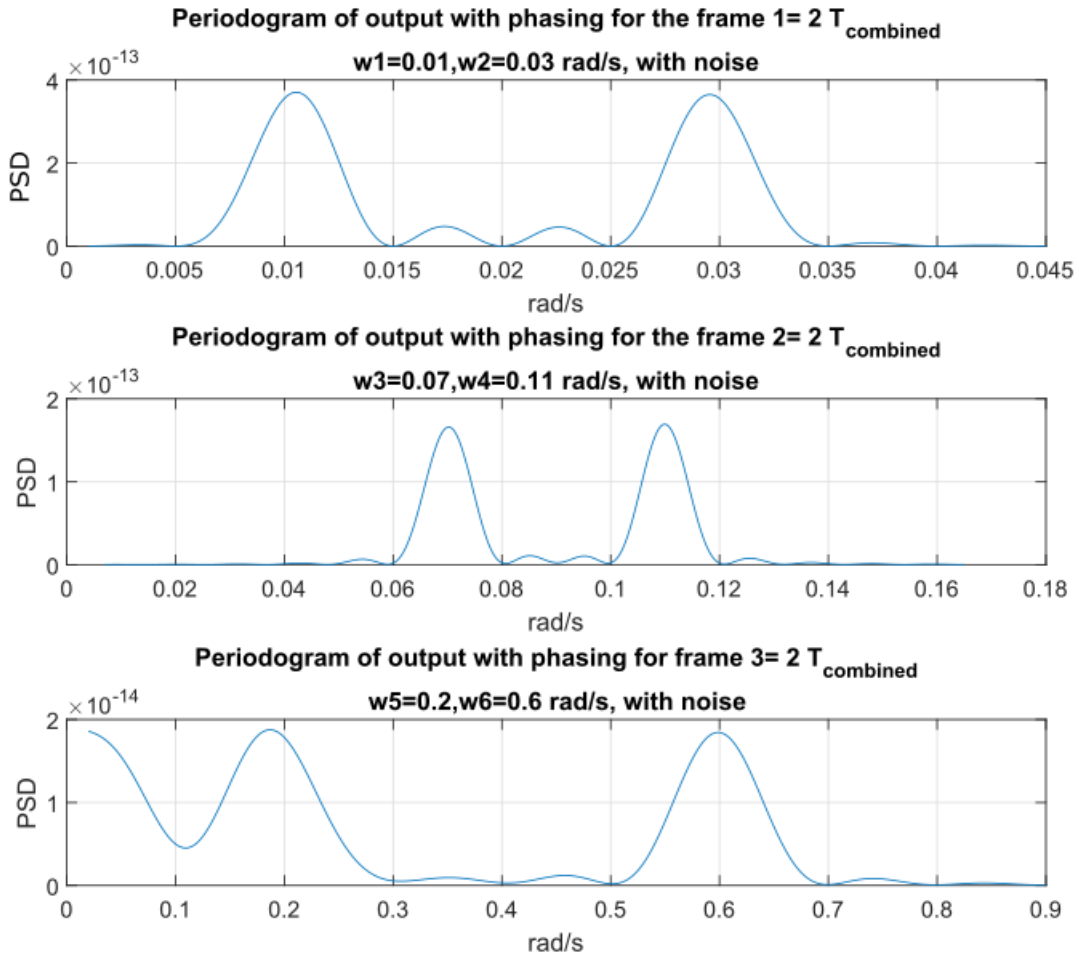


Figure 4.10: The periodogram of 6 sine signals, 3 frames, each frame $T_f = 2\ T_{combined}$, with noise

As can be seen in Fig. 4.9 and Fig. 4.10, periodograms of $T_f = 2T_{combined}$ are sharper compared to other one, which is because the resolution of periodogram is inverse of recording time ($\Delta f = \frac{1}{T_f}$). As can be observed, the frequencies of 3 frames are detected. Next we repeat the simulation with the same initial conditions, reference input. The output signal and the periodograms of the output for each frame are shown in Fig. 4.11 and Fig. 4.12. In order to see the effect of phasing, two simulations are performed, one without and one with phasing. We observe that in both cases, the watermarking frequencies can be identified in the output.

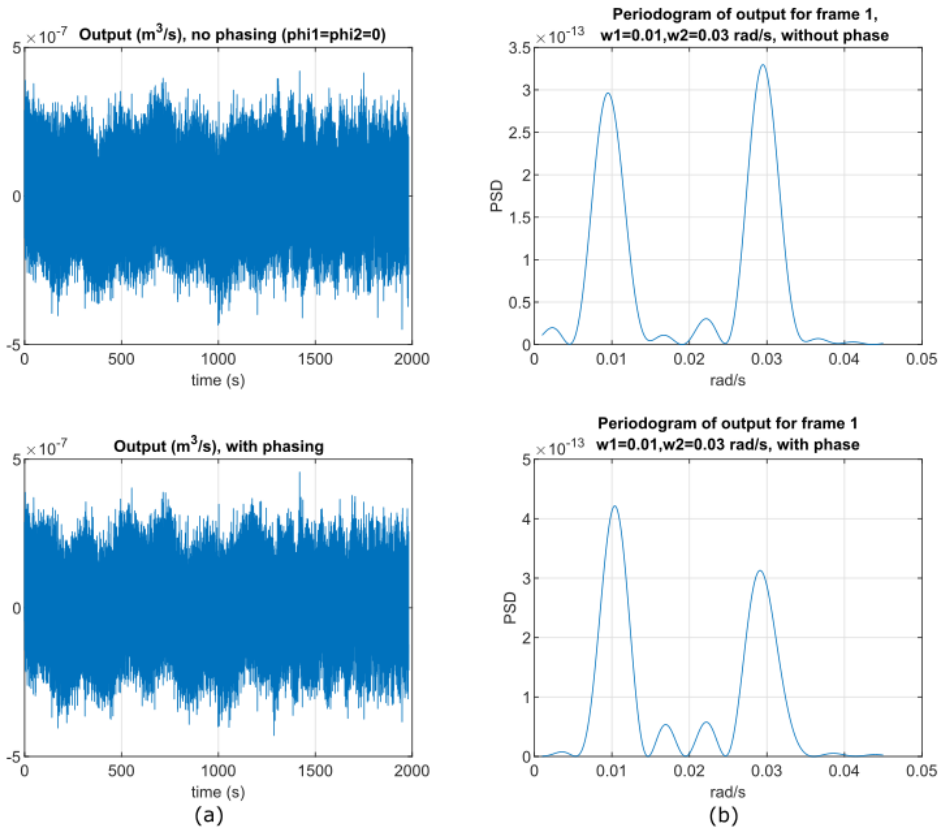


Figure 4.11: (a): The output of system with process and sensor noise, without reference input, (b) the periodogram of frame 1, $T_f = 2\ T_{combined}$, with noise, with/without phasing

Furthermore, the periodogram of with phasing and without phasing cases are very similar. It seems that the benefit of transient suppression is in time domain and in limiting output fluctuations. In this simulation, the ratio of the worst case fluctuation due to watermarking to 3σ value of input and output disturbance are:

$$\frac{A_1|G_{mq_2}(jw_1)| + A_2|G_{mq_2}(jw_2)|}{3\sigma_v} = 0.25$$

and

$$\frac{A_1|G_{mq_2}(jw_1)| + A_2|G_{mq_2}(jw_2)|}{3\sigma_w} = 0.16$$

This shows that the amplitude of fluctuations from watermarking is much smaller than the disturbance.

Since the output signal is stochastic, its periodogramis stochastic. In Fig. 4.12, teh periodograms of three frames are provided for another output sample. The watermarking frequencies are easily detectable.

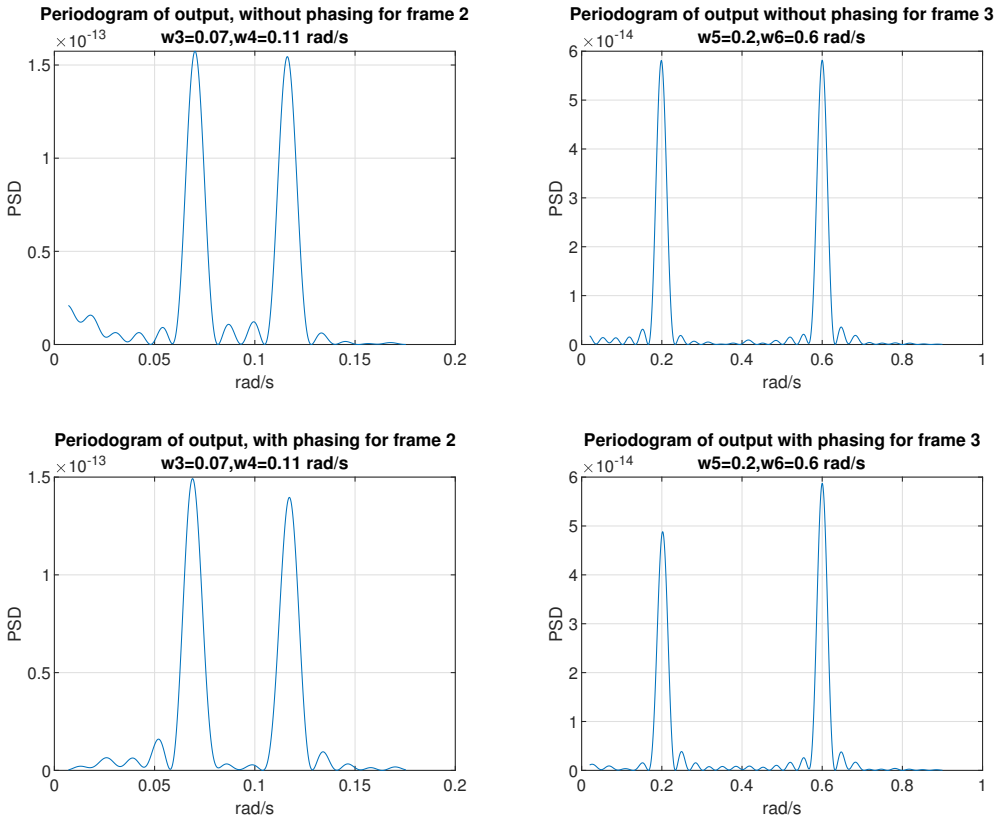
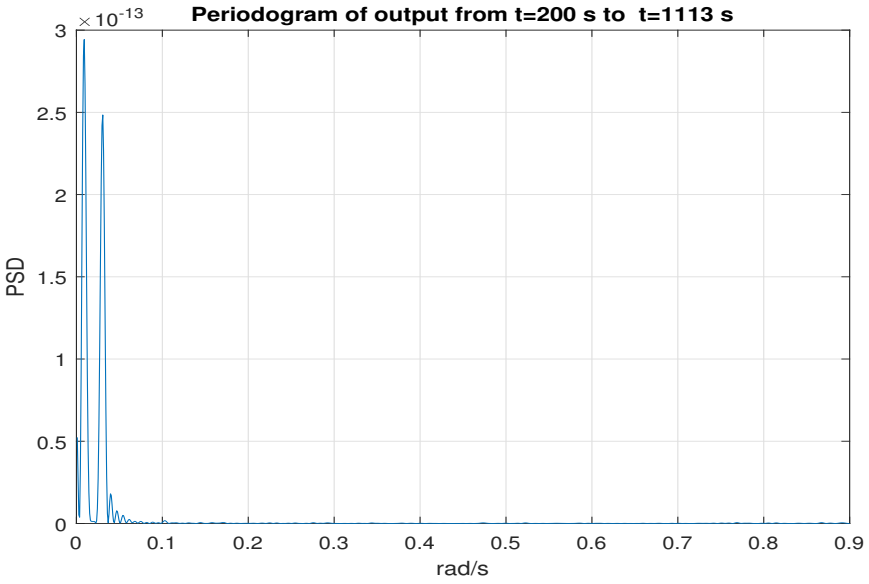


Figure 4.12: Another sample of Periodogram of 4 sine signals, frames 2 and 3, each frame $T_f = 2\ T_{combined}$, with noise, with/without phasing

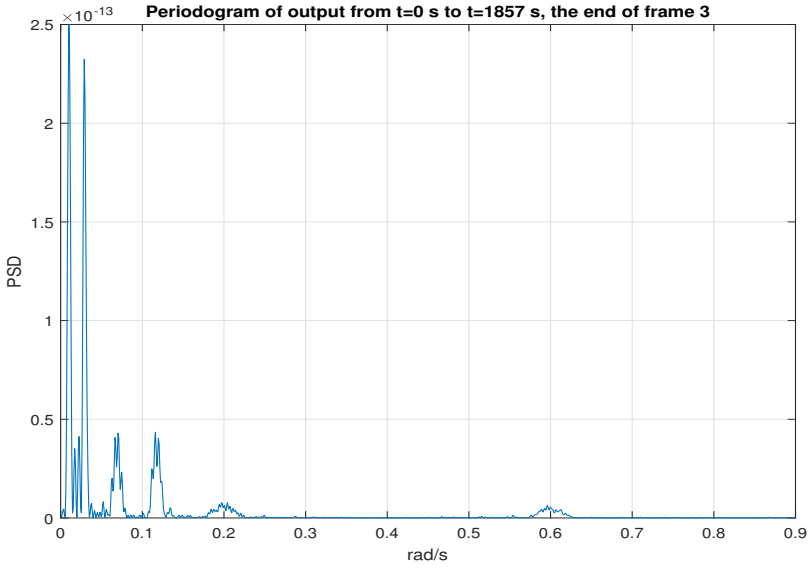
Attacker's analysis:

Now let us look at analysis of the output from an attacker's point of view. In a replay attack, the attacker does not need to know much about the plant , and simply records and replays the system output. Now suppose the attacker intends to examine the output for trace of watermarking. The attacker knows neither the frequencies used in watermarking (not even the signal type), nor the start time and the size of frames. In the following, we examine a few cases. Here, watermarking is done using three frames and for each frame $T_f = 2T_{combined}$. The watermarking with phasing is used and simulation is done in the presence of process and sensor noise. The frames are:

Frame 1: $0 \leq t \leq 1256$, Frame 2: $1256 \leq t \leq 1794$, Frame 3: $1794 \leq t \leq 1857$

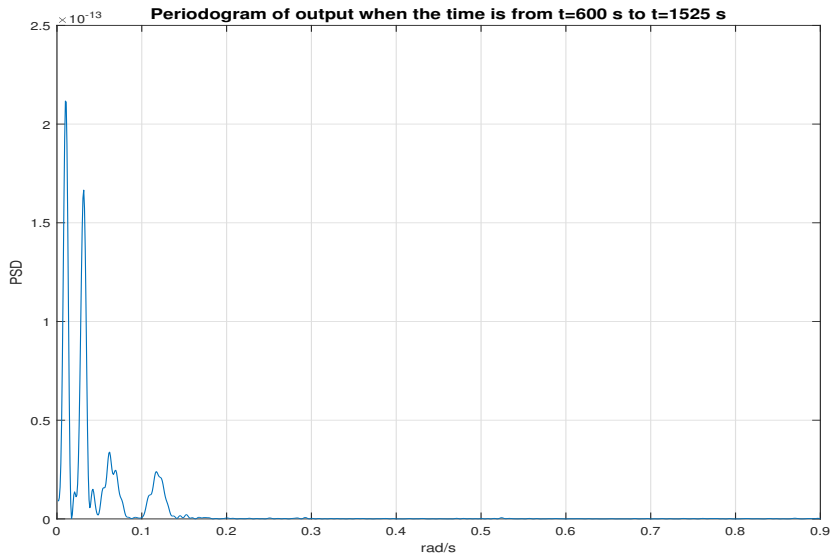


(a) Periodogram of output, from t=200 s to t=1113 s

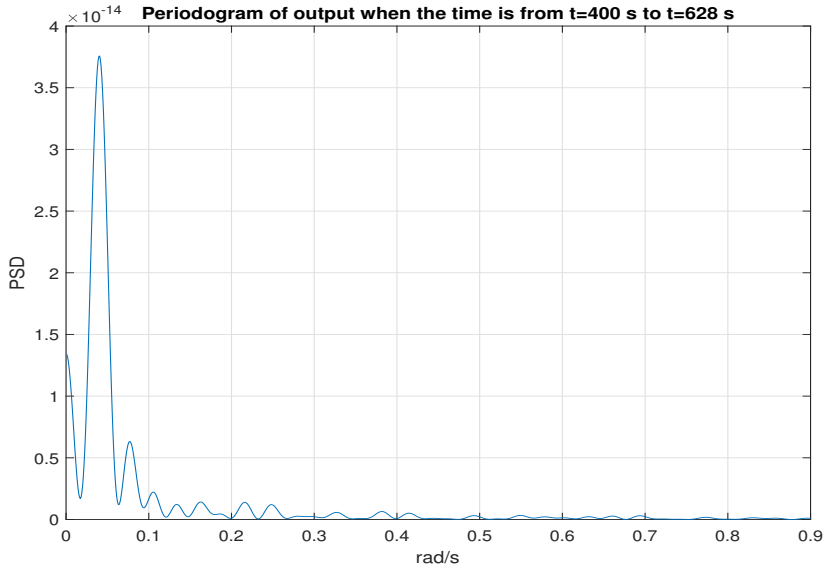


(b) Periodogram of output for 3 frames, from t=0 s to t=1857 s

Figure 4.13: Periodogram of output, with phasing signal in $T_f = 2\ T_{combine}$



(a) Periodogram of output, from t=600 s to t=1525 s



(b) Periodogram of output, from t=400 s to t=628 s

Figure 4.14: Periodogram of output, with phasing signal in $T_f = 2\ T_{combine}$

In Fig. 4.13a, the attacker records and examines the output for $200 \leq t \leq 1080$ which almost corresponds to frame 1. The watermarking frequencies of 0.01 rad/s and 0.03 rad/s are detectable;

however, by the $t = 1080\ \ s$, frame 1 is almost over and frame 2 with new watermarking frequencies will start. Therefore, the result of analysis is not useful for attacker. In Fig. 4.13b, the attacker examines the output from $t = 0\ \ s$ to $t = 1857\ \ s$. More frequencies are detected; however, similar to previous case, the result may not be useful for attacker. In Fig. 4.14b, the output is examined at shorter period $t = 600\ \ s$ to $t = 1525\ \ s$. This period of time covers second half of frame 1 and the first half of frame 2. Four frequencies are detectable; however, it is not known if all belong to a single frame or multiple frames. In this case, by the time that the frequencies are detected, half of frame 2 has passed, and it would be too late for attacker to artificially add sine waves to the output signal sent to the controller. Finally, In Fig. 4.14, the attacker examines a much smaller slice of output between $t = 400\ \ s$ and $t = 628\ \ s$. This interval falls in $\frac{1}{3}$ of the period of watermarking of sine wave of frame 1. As a result, only frequency 0.03 rad/sec is detected and 0.01 goes undetected by the attacker.

4.3.2 Case 2: $\alpha = 6.72 \times 10^{-8}$ and $\eta = 0.04$, nonlinear system

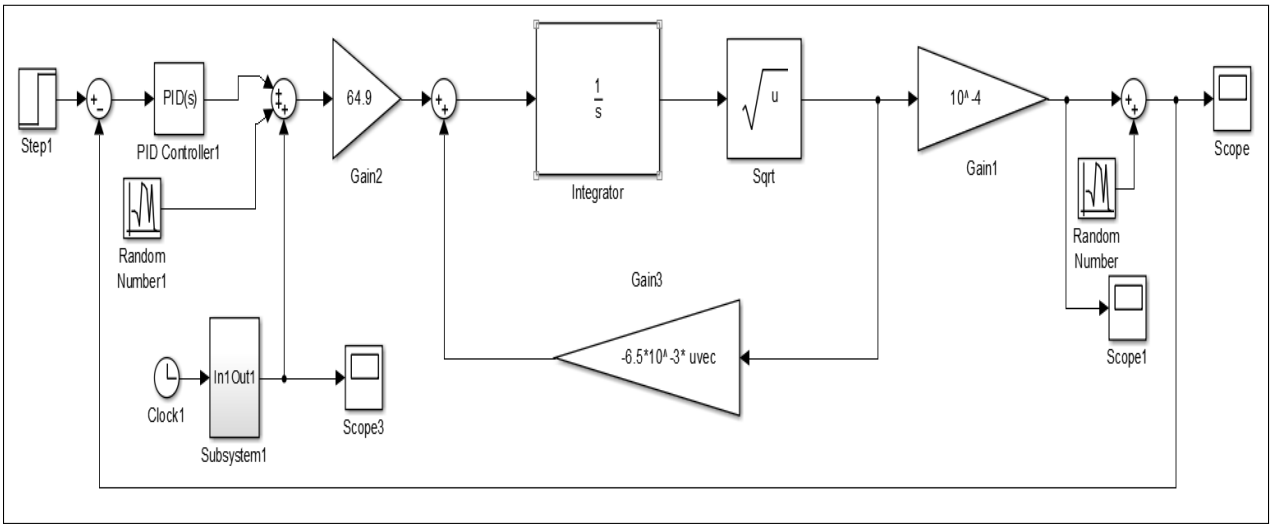


Figure 4.15: Nonlinear model of system in Simulink

In this subsection, we repeat the simulation of the previous case with two changes. First we use the nonlinear model of the plant. Secondly, we assume the plant initially is not at its operating

point. $Q_2(0) = 5.38 \times 10^{-5}$, and the initial level $H(0) = 0.29\ \ m$ (resulting $Q_2(0) = 5.385 \times 10^{-5}$ and initial condition for block of $\frac{1}{s}$ in PID controller is 5.4×10^{-5}. Also the reference input $r(0) = 5.3 \times 10^{-5}$ and a step input applied to system at $t = 0\ \ s$, and r(t) becomes 5.4×10^{-5} for $t > 0$. The plant output and measured output are shown in Fig. 4.16. Watermarking signal (with phasing) is done from $t = 0\ \ s$ (with $T_f = 2\ \ T_{combined}$), as was done in case 1. Note that it takes about $200\ \ s$ for the output to settle and to reach steady state.

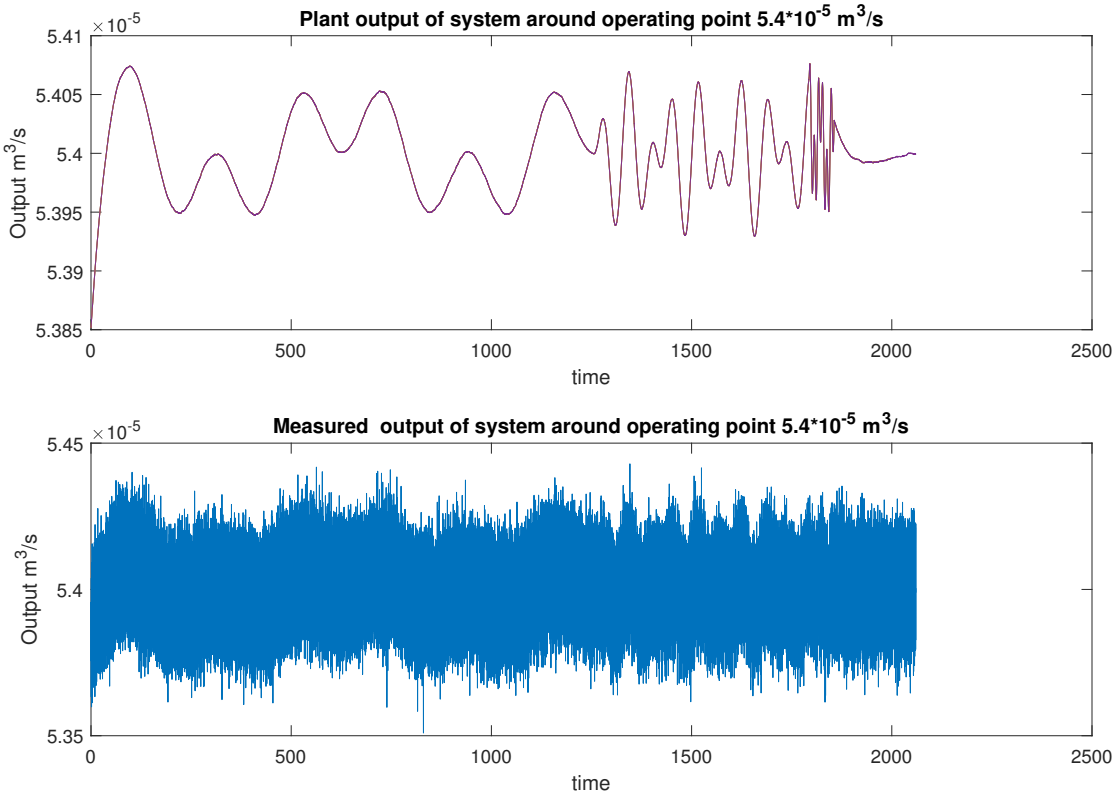


Figure 4.16: The plant and measured output for nonlinear system with noise, with reference input, watermarking signal with phasing in frame size $T_f = 2\ \ T_{combined}$ for each frame, starting from $t = 0\ \ s$

In Fig. 4.17 the periodogram of 3 frames can be seen. We observe that the result is very similar to periodogram of Fig. 4.10 of linear system. The similarity is to be expected since deviations of system from the operating point are small and therefore the linear and nonlinear simulations have close results.

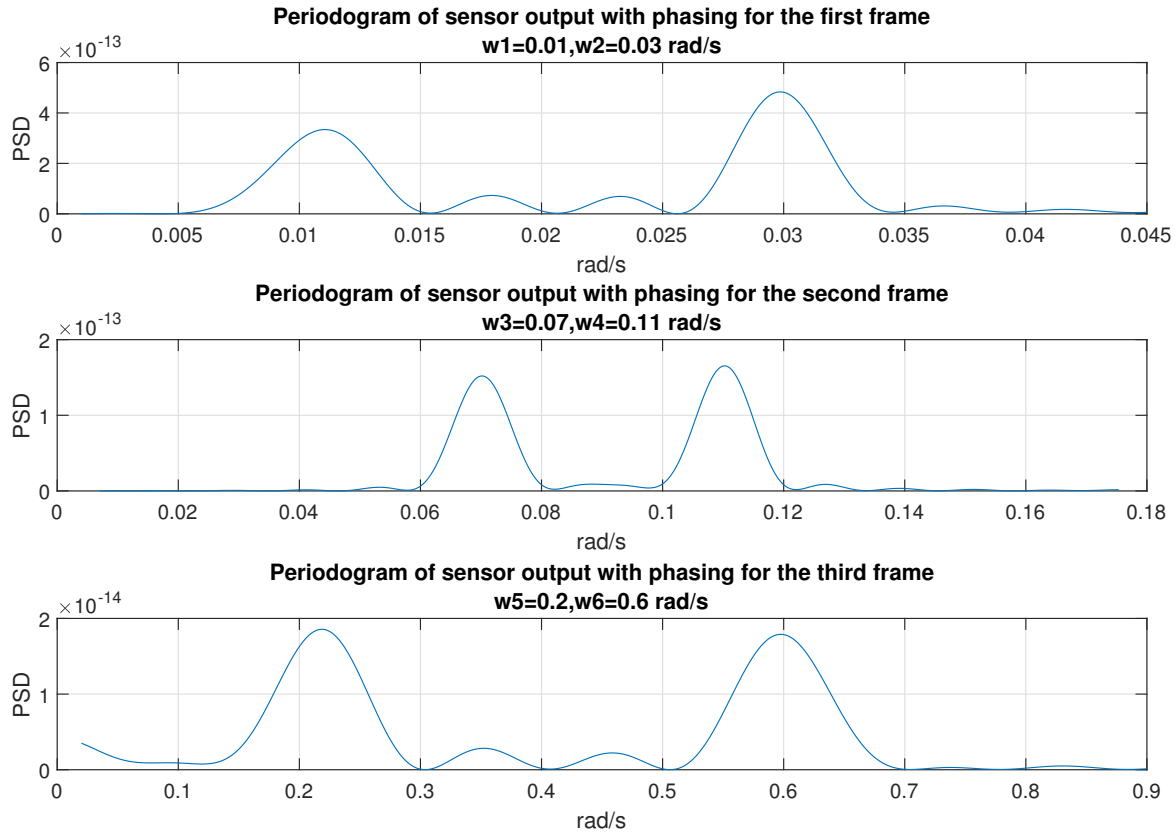


Figure 4.17: Periodogram of 3 frames, each frame $T_f = 2\ T_{combined}$, for the nonlinear system

4.3.3 Case 3: $\alpha = 2.3544 \times 10^{-8}$ and $\eta = 0.0049$, linear system

In this case, we use the linear model to simulate a replay attack and explore the detection of attack by controller using the periodogram of output frames.

Similar to case 1, the reference input is zero. The plant is subject to process and sensor noise. Watermarking is similar to case 1 with one difference, the value of η. Using trail and error we have found the smallest amplitude of sine waves for which the effects of watermarking can be detected from the output using periodogram. The value for η is 0.0049, which corresponds to $\alpha = 2.3544 \times 10^{-8}$, and $|A_i G_{mq2}(j\omega_i)| = \frac{2.3544 \times 10^{-8}}{2} = 1.1772 \times 10^{-8}$. We will discuss watermarking with and without phasing. All frame sizes are twice the combined period: $T_f = 2T_{combined}$. The sinusoids with phasing are as following:

$1.24 \times 10^{-7} \sin(0.01t + 0.66) + 7.92 \times 10^{-8} \sin(0.03t - 1.3)$

$1.44 \times 10^{-7} \sin(0.07t + 2.58) + 2.35 \times 10^{-7} \sin(0.11t - 0.33)$

$3.997 \times 10^{-7} \sin(0.2t + 2.94) + 1.1953 \times 10^{-6} \sin(0.6t - 0.06)$

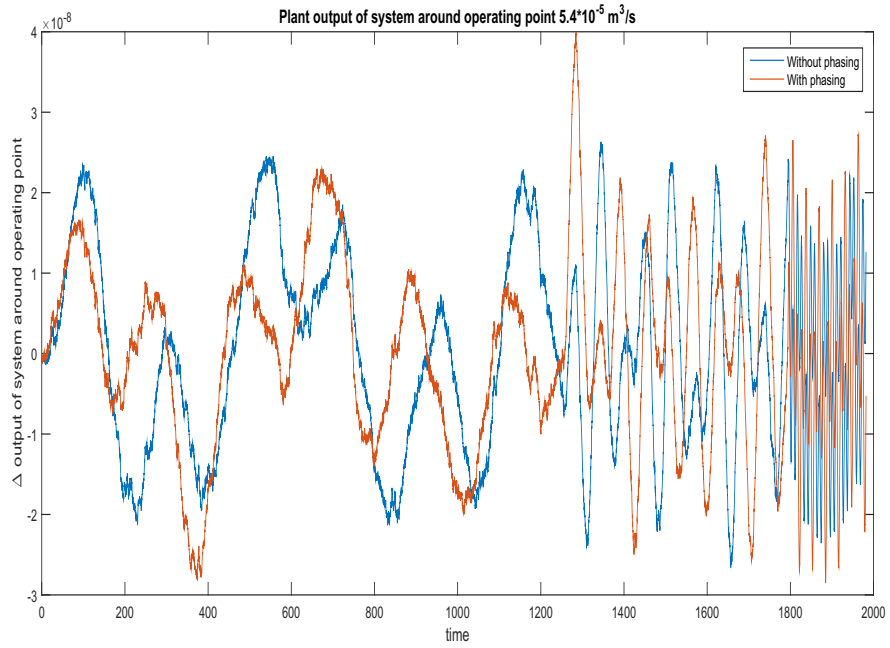
and the sine signals without phase are:

$1.24 \times 10^{-7} \sin(0.01t) + 7.92 \times 10^{-8} \sin(0.03t)$

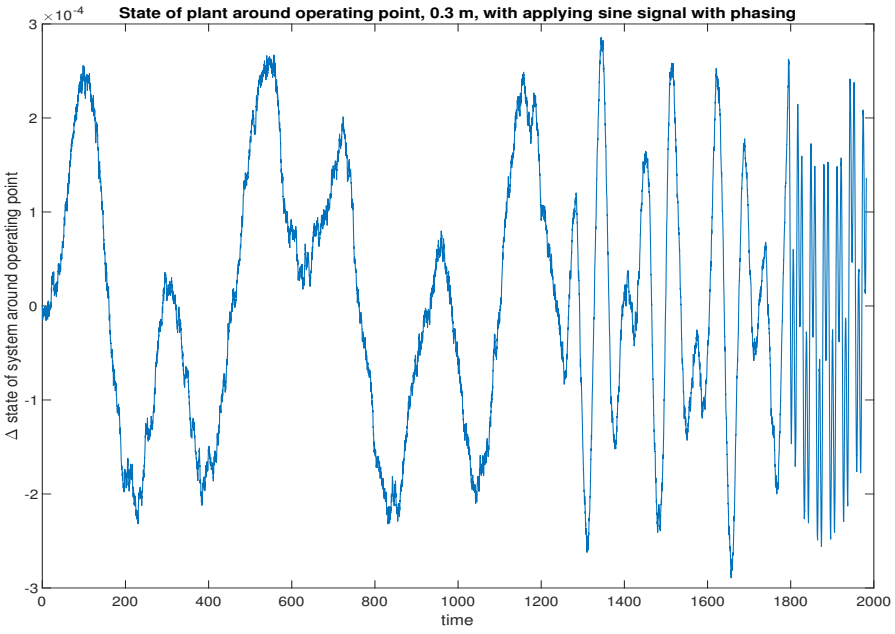
$1.44 \times 10^{-7} \sin(0.07t) + 2.35 \times 10^{-7} \sin(0.11t)$

$3.99 \times 10^{-7} \sin(0.2t) + 1.1953 \times 10^{-6} \sin(0.6t)$

For each frame, time t is from the start time of the corresponding frame. In Fig. 4.18, the effect of noise is considered, and the plant output under noise is depicted.



(a) Plant output of linearized system around the operating point with/without phasing in Simulink



(b) Water level (h) of linearized system around the operating point with applying sine signal with phasing in Simulink

Figure 4.18: Plant output and state of linearized system around the operating point with/without phasing, with noise in Simulink

Next, we describe the attack scenario. Fig. 4.19 shows the block diagram of the closed-loop system of the linearized system. Here, $q_2^a(t)$ denotes the actual value of output and $q_2(t)$ is the measured value (i.e. sensor reading). We assume the sensor is ideal with the exception of the noise: $q_2(t) = q_2^a(t) + v(t)$. Furthermore, $q_2^c(t)$ denotes the information received by the controller. In the absence of attack: $q_2^c(t) = q_2(t)$. During a replay attack, the attacker replaces $q_2(t)$ with prerecorded segments of $q_2(t)$. In our case, we assume that $q_2(t)$ is recorded between $t = 600\ s$ and $t = 800\ s$. Then after $t = 800\ s$, the output segments from $t = 600\ s$ to $t = 800\ s$ is played back (supplied as $q_2^c(t)$). As an illustrative example in Fig. 4.20, the output signal received by the controller, $q_2^c(t)$, is depicted. In this simulation the noise is assumed zero to improve the clarity of the signal. In Fig. 4.19 the controller uses $q_2^c(t)$, calculates a control command, and adds the watermarking signal, and the output will be $u_c(t)$. The overall signal will be $U(t) = U_0 + u_c$ where U_0 is controller output at the operating point. We assume that the attacker replace $U(t)$ with $1.2U(t)$ (i.e. amplifies the control signal). Thus in the linearized model, the signal added by the attacker, $u_{attack}(t)$ is

$$\begin{cases} u_{attack} = (1.2U(t) - U_0) - u_c(t) \\ \qquad\qquad = 0.2U_0(t) + 0.2u_c(t) \end{cases}$$

The control output at the operating point is $U_0 = 5.4 \times 10^{-5}\ m^3/s$. The results of simulation with/without noise, and watermarking with phasing are shown in Fig. 4.21 and Fig. 4.22

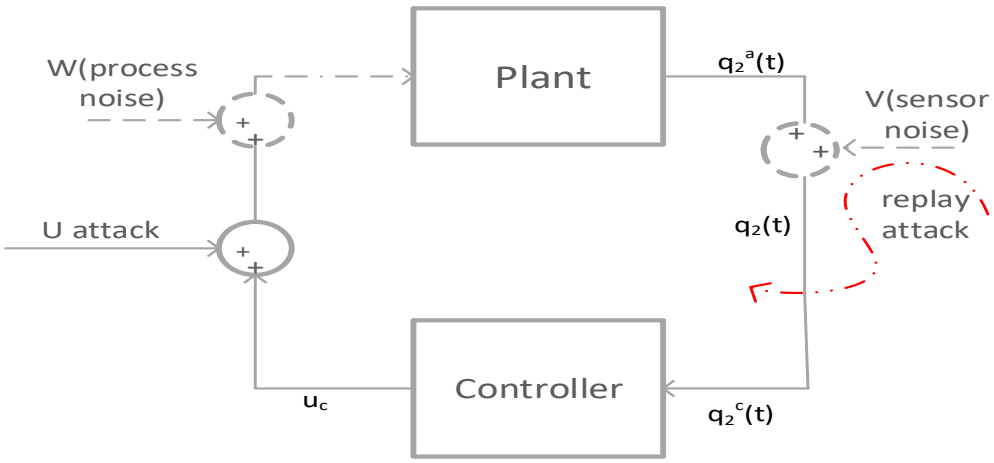


Figure 4.19: Replay attack

In Fig. 4.20, the output under attack is shown in which watermarking signal frame size is $T_f = 2\ T_{combined}$ of frame 1, without noise.

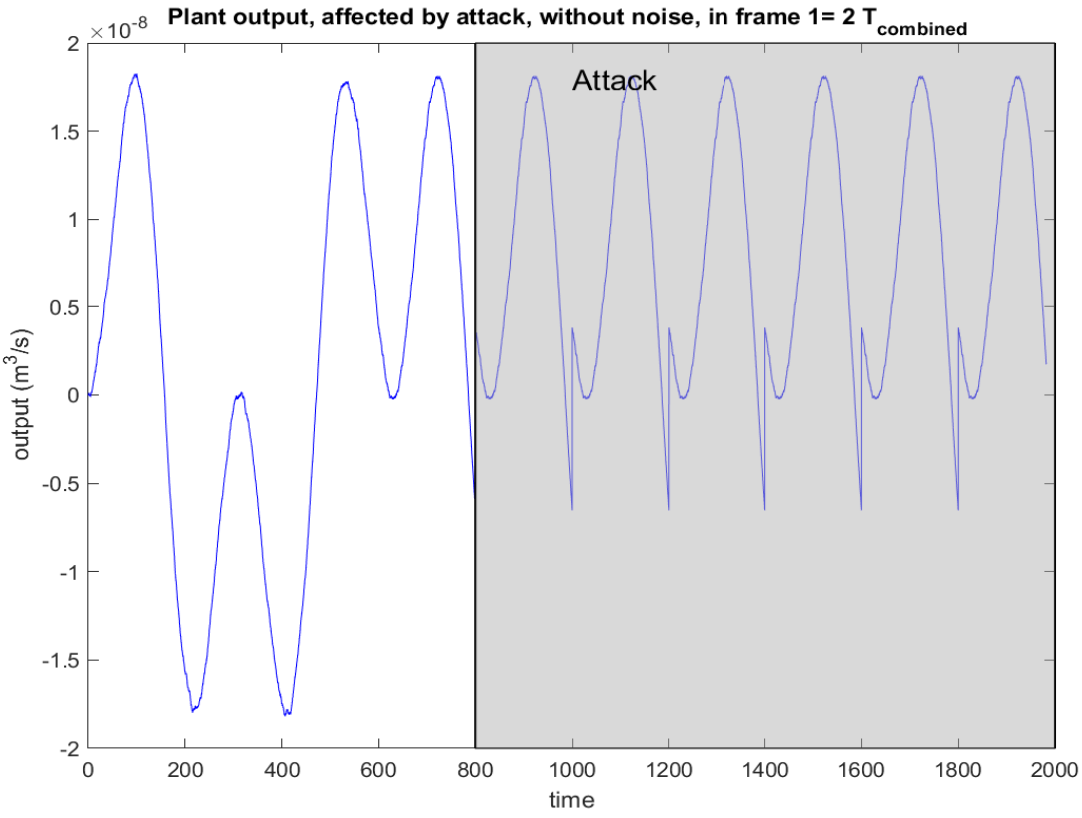


Figure 4.20: Fake plant output with attack, without noise

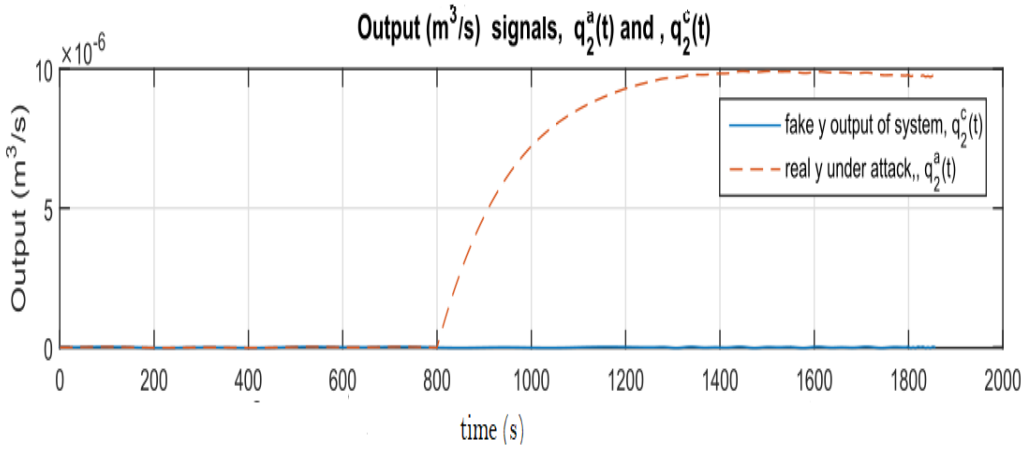


Figure 4.21: Real and fake output of plant, $q_2^a(t)$ and $q_2^c(t)$, with attack, without noise

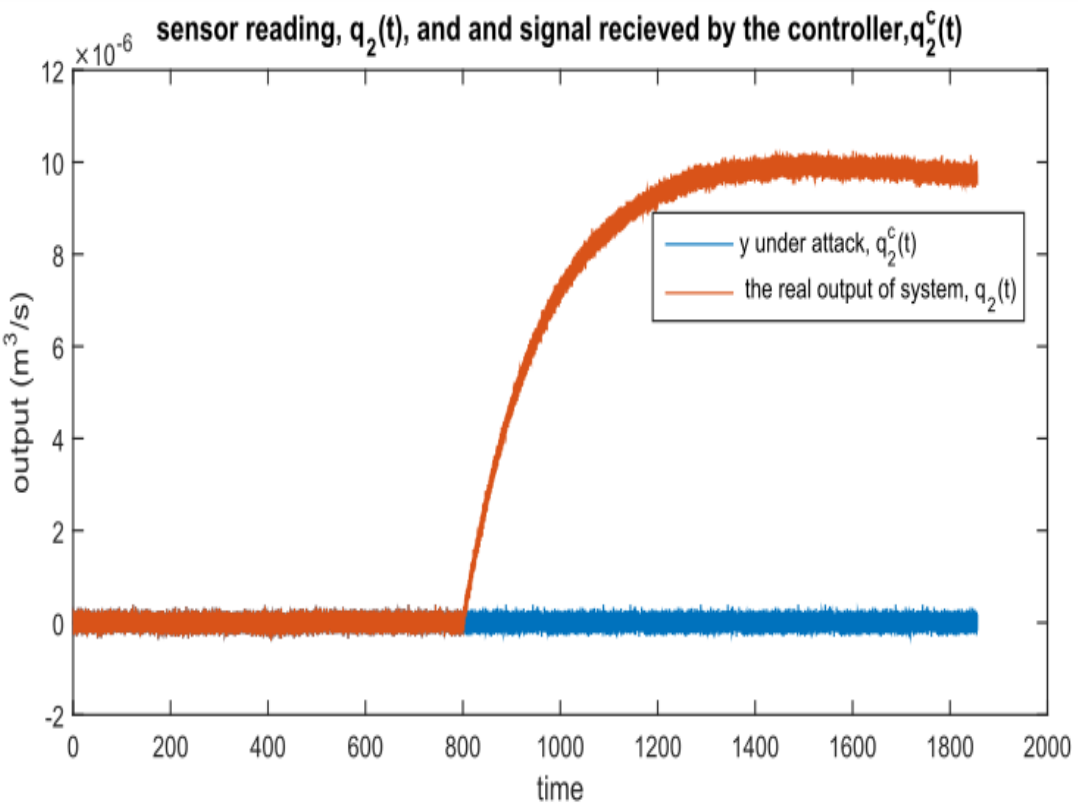


Figure 4.22: Real and fake output of sensor,$q_2(t)$ and $q_2^c(t)$ with attack, with noise

Now let us examine the periodogram of the signal received by controller ($q_2^c(t)$) and see how it can be used to detect the replay attack. From the point of view of Fig. 4.23 and Fig. 4.24 and Fig. 4.25 depict the periodograms of sensor readings for the three frames. The watermarking frequencies can be easily detected from the periodograms.

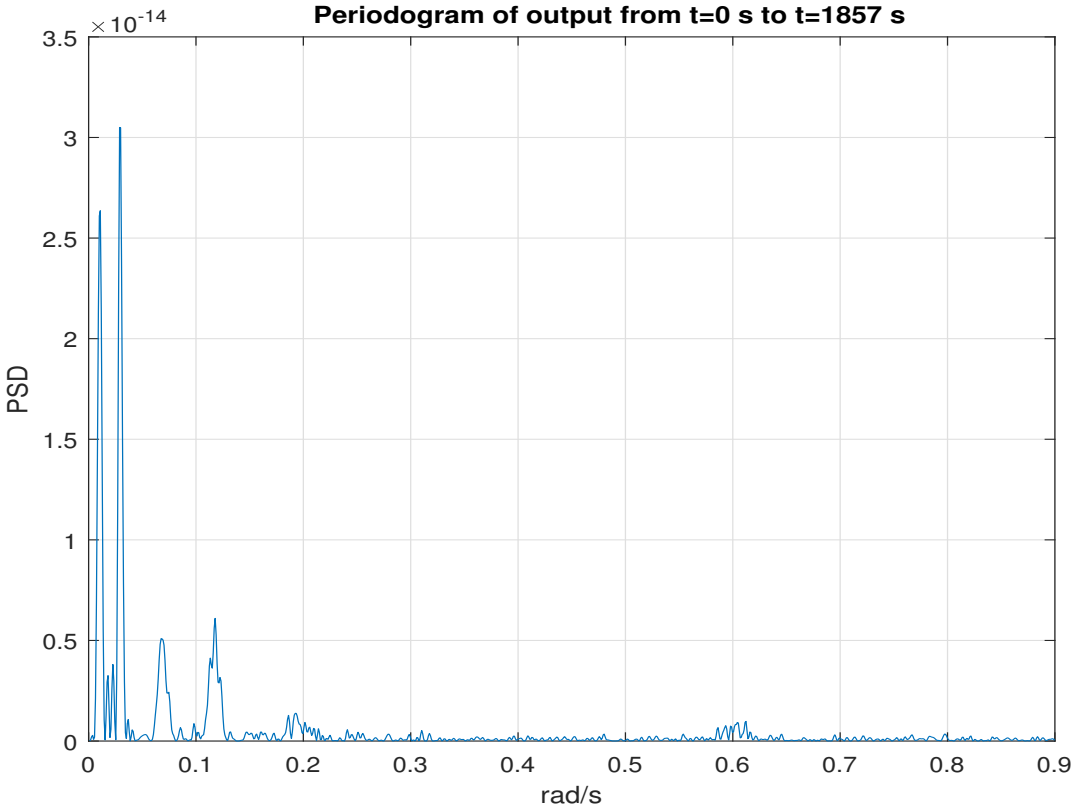


Figure 4.23: periodogram in frame size $T_f = 2\ \ T_{combined}$ in 3 frames

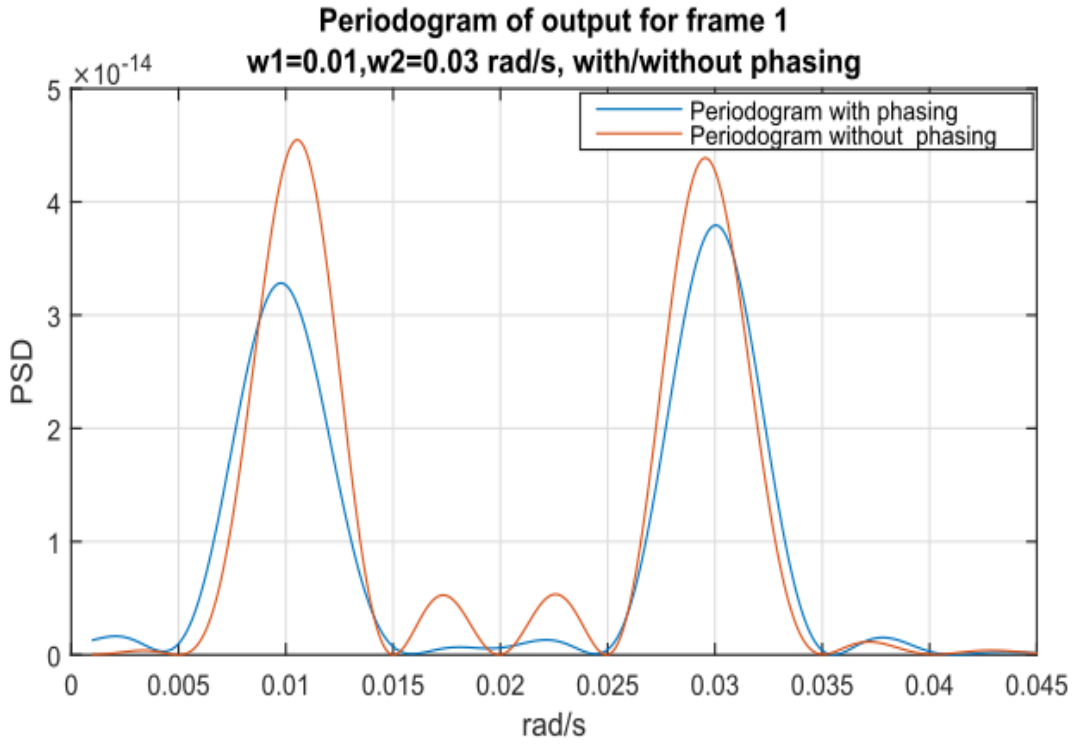
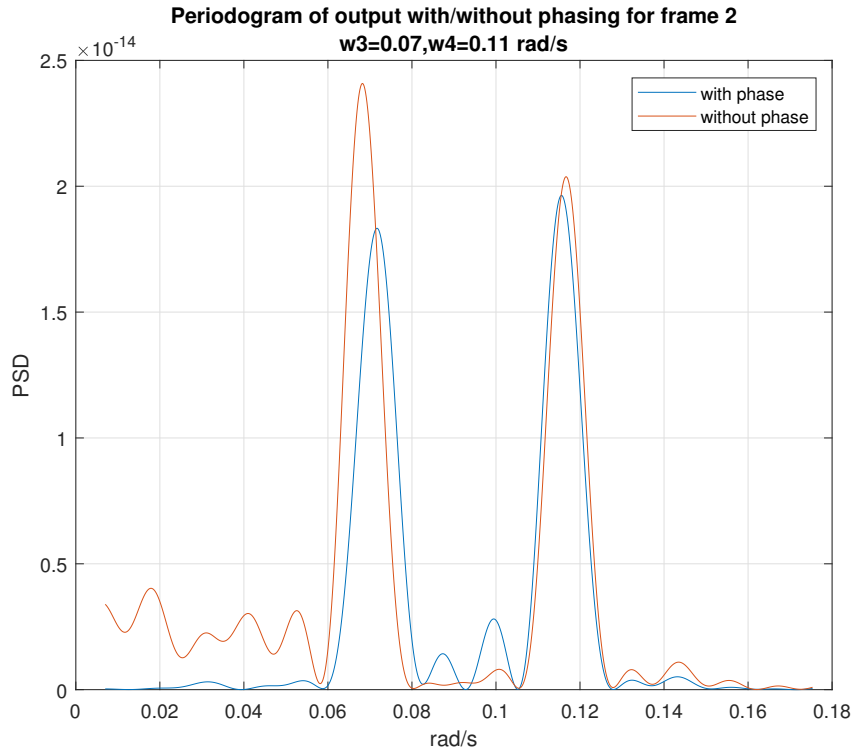
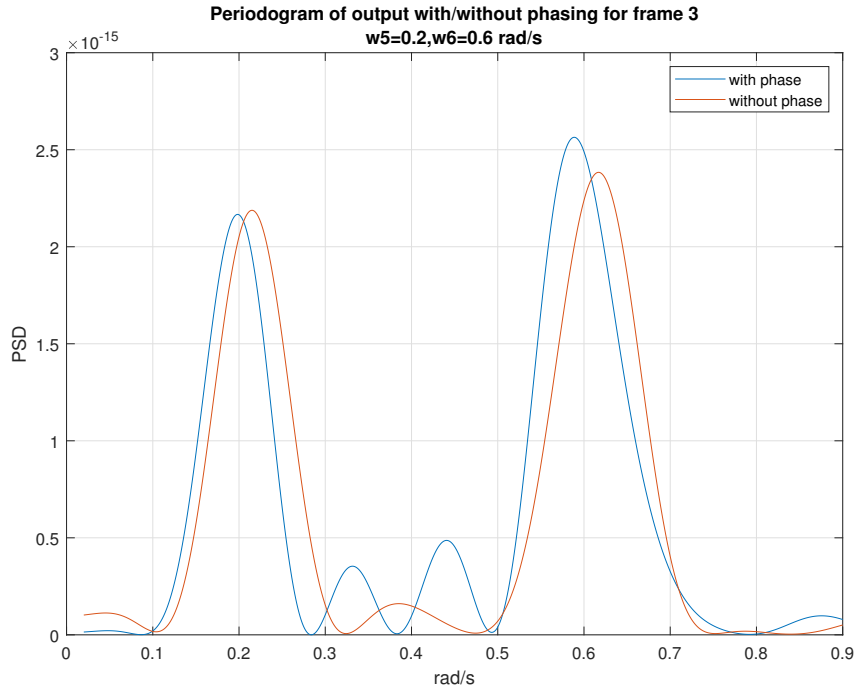


Figure 4.24: Periodogram of output for frame 1, $T_f = 2\ T_{combined}$ with and without phasing, with noise



(a) Periodogram of output for frame 2, $T_f = 2\ T_{combined}$ with and without phasing

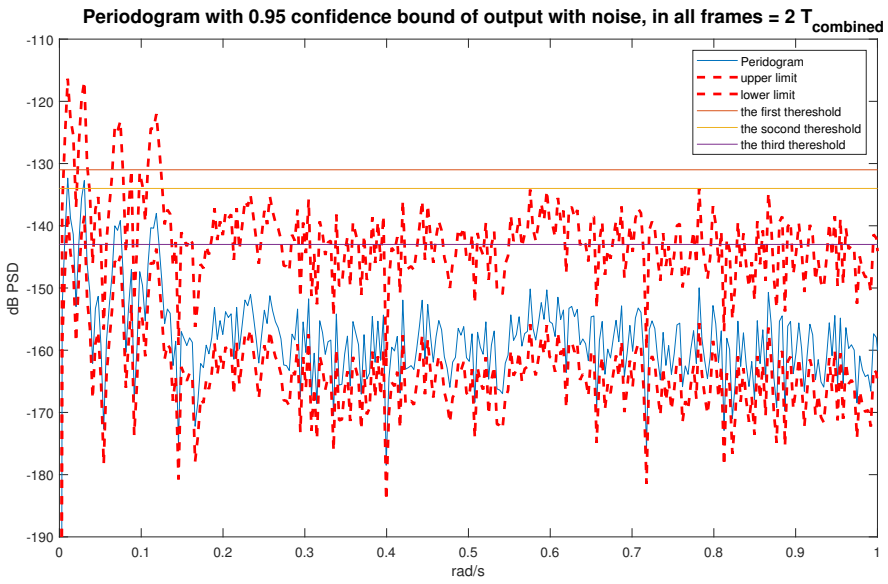


(b) Periodogram of output for frame 3, $T_f = 2\ T_{combined}$ with and without phasing

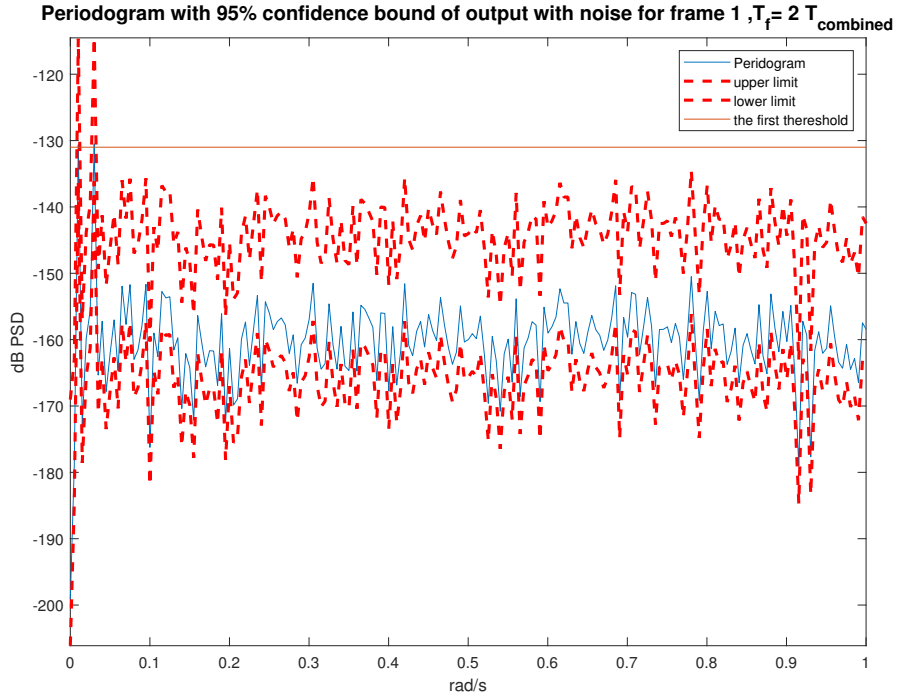
Figure 4.25: Periodogram of output for frame size $T_f = 2\ T_{combined}$ for frame 2 and 3, with and without phasing, with noise

Fig. 4.26 and Fig. 4.27 provide periodograms in dB (for watermarking with phasing). Additionally 0.95%- confidence lower and upper bounds are provided. We observe that the magnitude of periodograms at the watermarking frequencies (shown with a horizontal line) are about 15dB larger than the other frequencies.

Next the periodogram of output signal received by controller ($q_2^c(t)$) under attack are shown in Fig. 4.28 and Fig. 4.29. In Fig. 4.28b, the watermarking frequencies can be seen. This can be justified by the fact that the attack starts at $t = 800\ \ s$, well into the first frame and the signal played back is the sensor readings from frame 1. Fig. 4.29a and Fig. 4.29b show the periodogram of frames 2 and 3. We see that the watermarking frequencies of frames 2 and 3 are not presented which could indicate an attack.

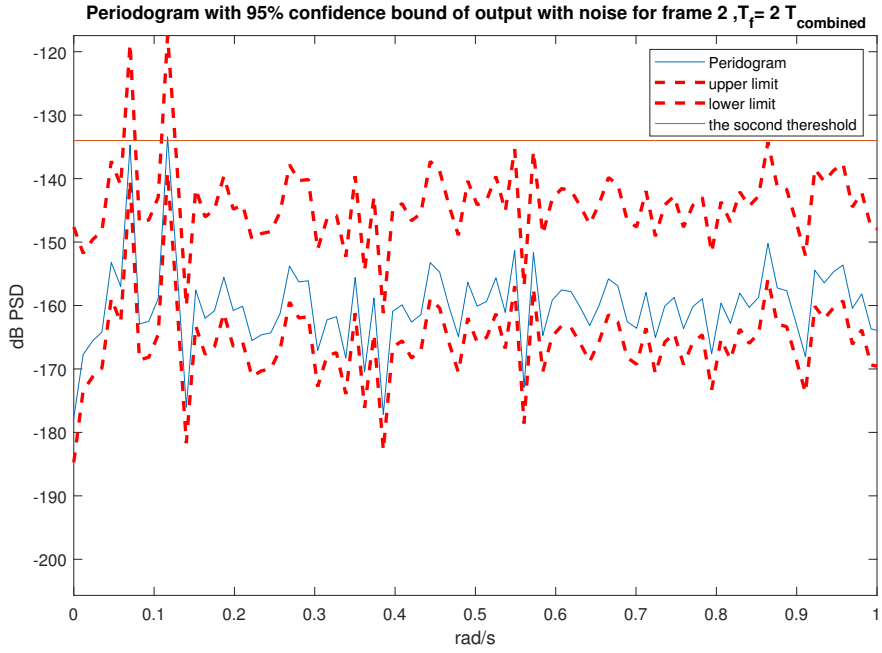


(a) Periodogram with 95% confidence bound of output without attack, in frame size $T_f = 2\ T_{combined}$ for 3 frames



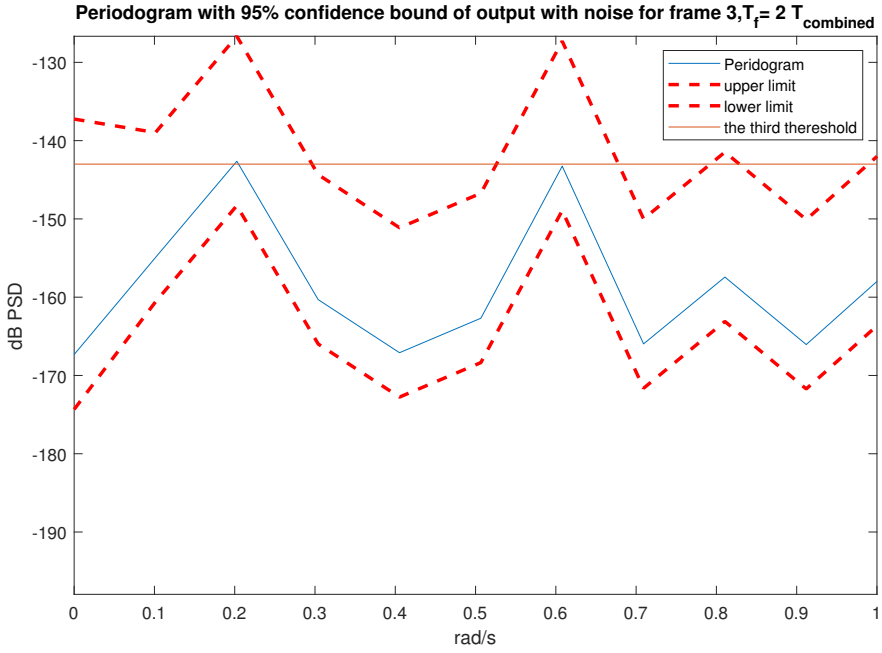
(b) Periodogram with 95% confidence bound of output without attack, in frame 1, $T_f = 2\ T_{combined}$

Figure 4.26: Periodogram with 95% confidence bound of output without attack, with noise, with phasing, for frame 1 and all frames $T_f = 2\ T_{combined}$



(a) Periodogram with 95% confidence bound of output without attack, for frame 2,

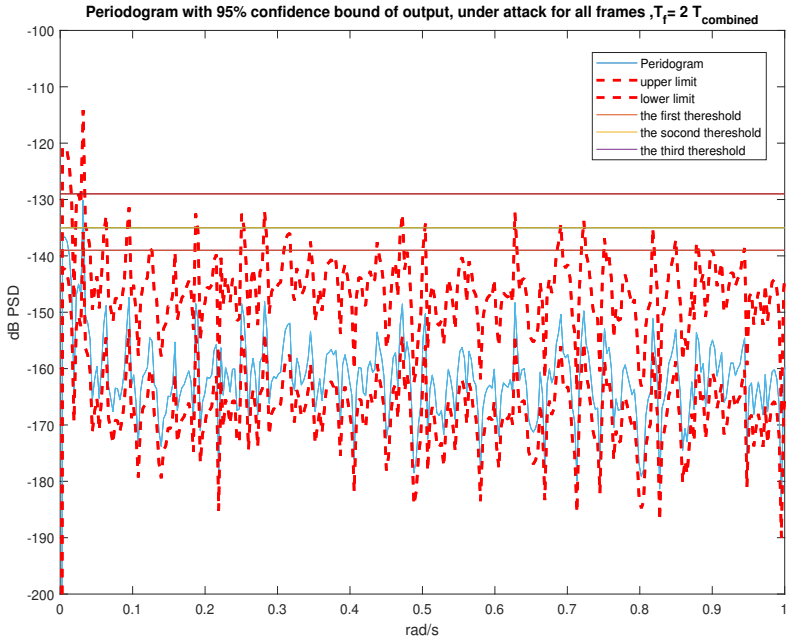
$$T_f = 2\ T_{combined}$$



(b) Periodogram with 95% confidence bound of output without attack, for frame 3,

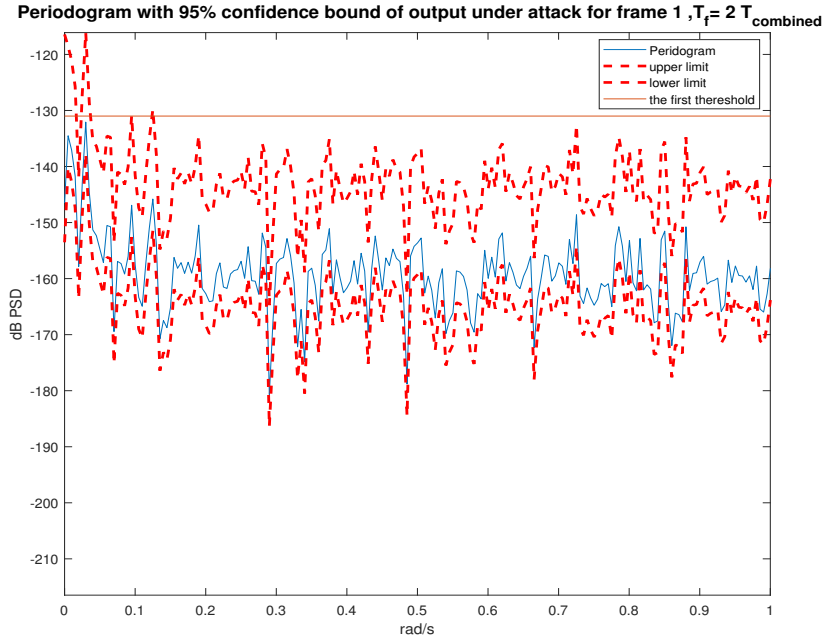
$$T_f = 2\ T_{combined}$$

Figure 4.27: Periodogram with 95% confidence bound of output without attack, with noise, with phasing, for frames 2 and 3, $T_f = 2\ T_{combined}$ for each frame



(a) Periodogram with 95% confidence bound of output under attack, for 3 frames,

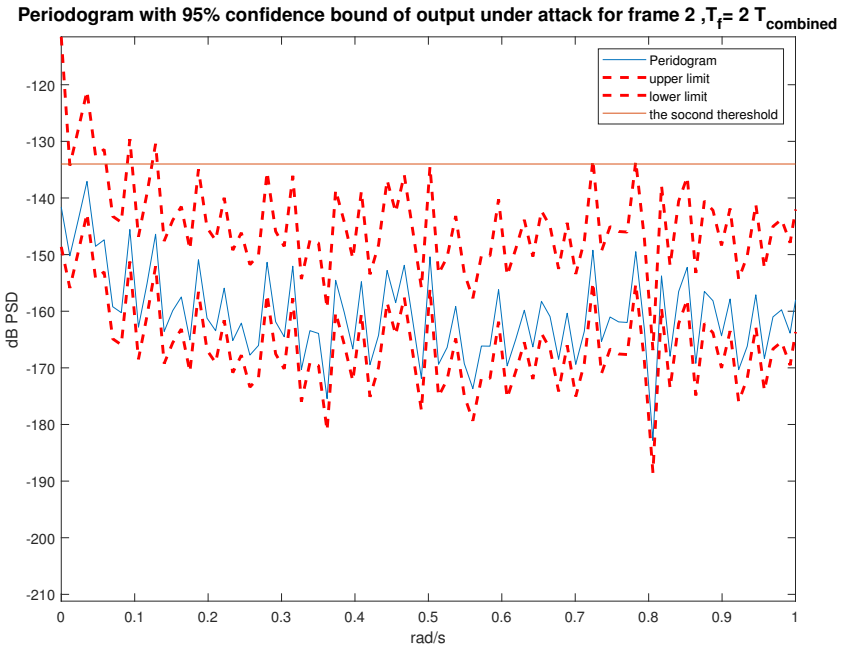
$$T_f = 2 \ T_{combined}$$



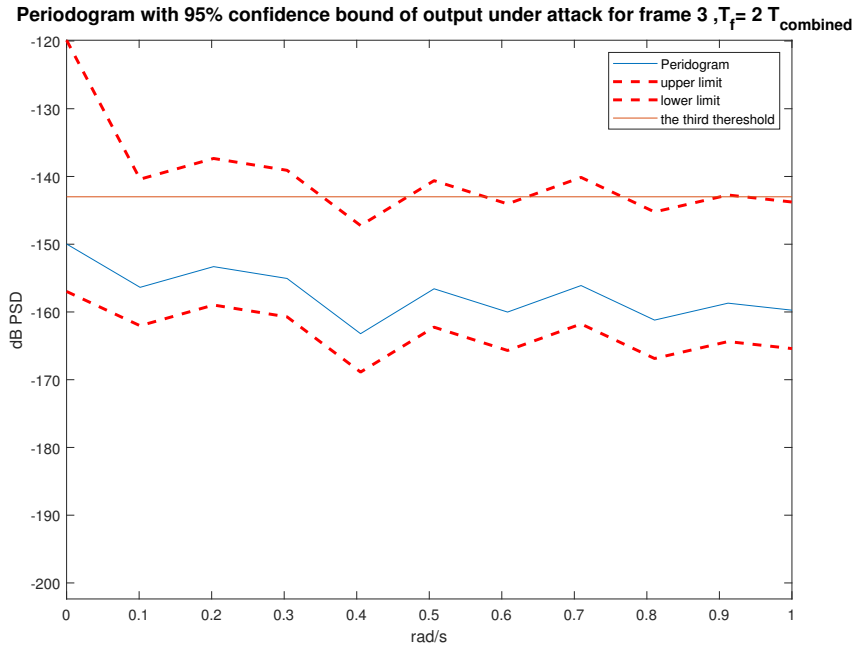
(b) Periodogram with 95% confidence bound of output under attack, for frame 1,

$$T_f = 2 \ T_{combined}$$

Figure 4.28: Periodogram with 95% confidence bound of output under attack, with phasing, with noise for frame 1 and 3 frames, $T_f = 2 \ T_{combined}$



(a) Periodogram with 95% confidence bound of output under attack, for frame 2, $T_f = 2\ T_{combined}$



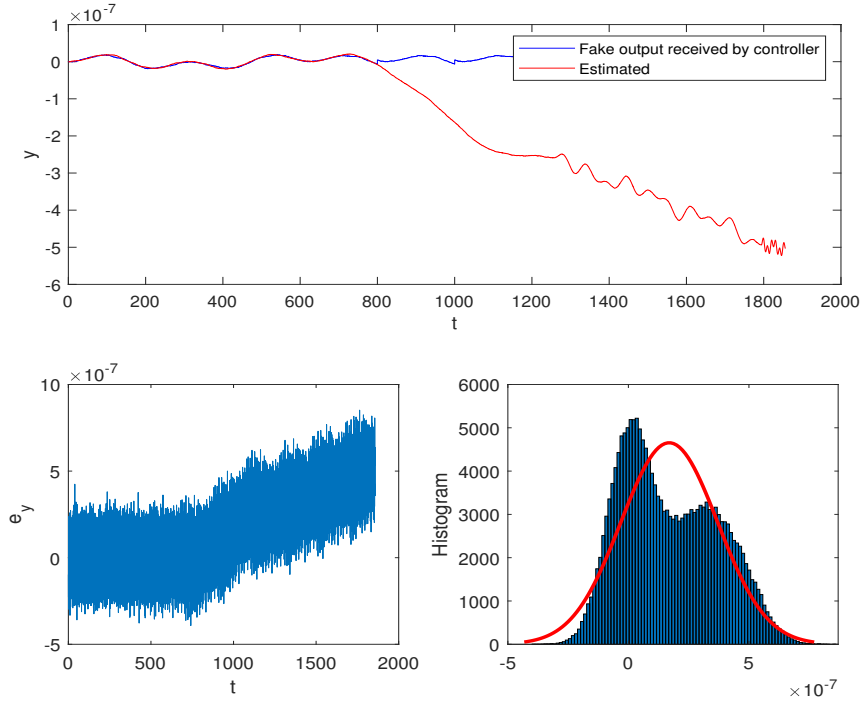
(b) Periodogram with 95% confidence bound of output under attack, for frame 3, $T_f = 2\ T_{combined}$

Figure 4.29: Periodogram with 95% confidence bound of output under attack, with phasing, with noise for frames 2 and 3, $T_f = 2\ T_{combined}$

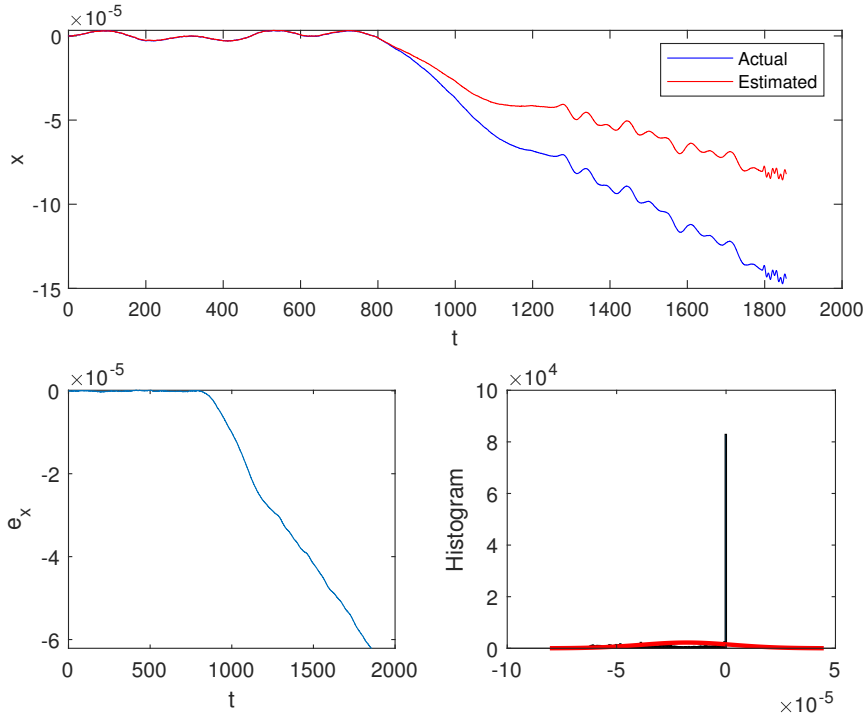
4.4 Detection with Kalman Filter

In this section, the detection of replay attack via a Kalman filter is studied. In [10], Mo et al. showed a replay attack cannot be necessarily detected using Kalman filter and proposed watermarking with random signals as a solution. Here, we propose watermarking using multi-sine waves estimating state x and output y via Kalman filter. As can be seen in the following case study, when there is no attack, Kalman filter detects nothing. When attack occurs, the Kalman filter immediately detects it since e_x (error of estimating state) and e_y (error of estimating state) become noticeable. Thus we can find out that an attack is happening. For this purpose, 3 different cases are examined. In each case, x, y, e_x, e_y and histogram of e_x and e_y are shown for 100 samples. All of the figures are drawn with frame size $T_f = 2\ T_{combined}$ for each frame.

- Case 1: In the linearized system, the reference input which is input deviation around the operating point is zero, and just watermarking sine signals with phasing and process and measurement noise are the inputs of system. The system is assumed to be in operating point and watermarking signals are applied at $t = 0\ s$. Attack scenario is similar to Section 4.3.3. Fig. 4.30 shows that with watermarking signal, Kalman filter detects the attack. The inputs to Kalman filter are output signal received by controller, and input signal generated by controller (which includes the watermarking signal).



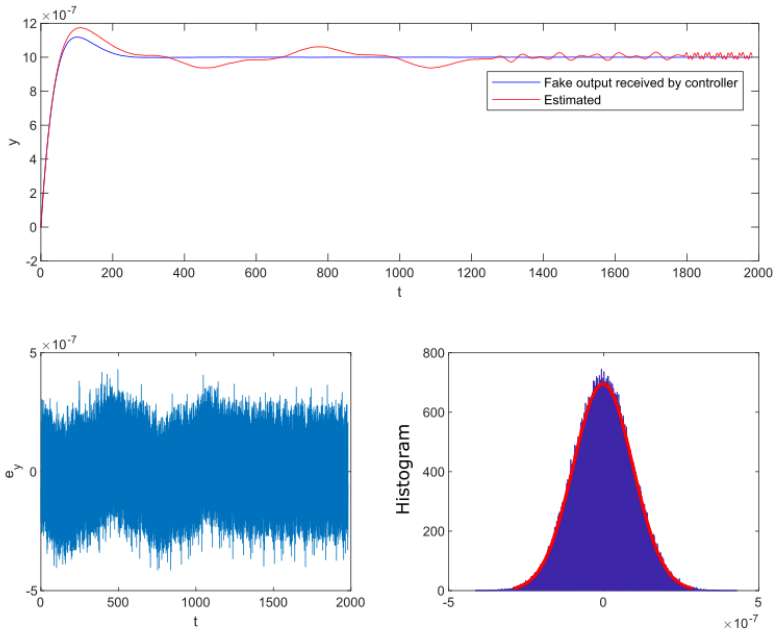
(a) Kalman filter result on y output



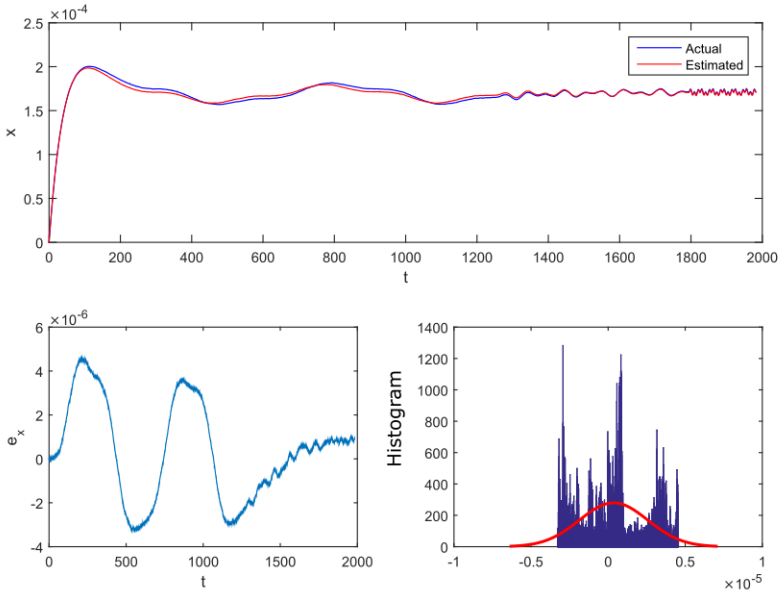
(b) Kalman filter result on state x (water level h)

Figure 4.30: Case 1: Kalman filter result for x, y in case of watermarking with sine wave with phasing

- Case 2: In this case, it is assumed that the reference input is a small value 0.1×10^{-5}, and process and measurement noise present, without any watermarking sine wave.



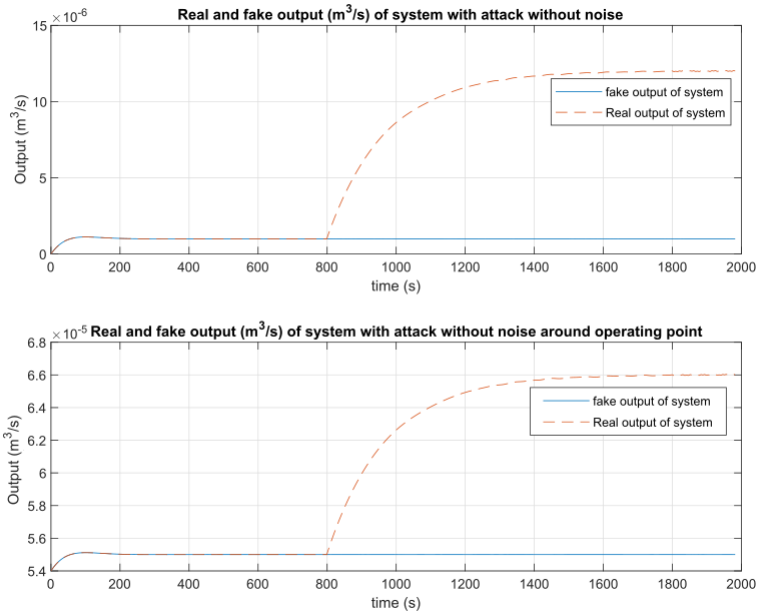
(a) Kalman filter result on y output



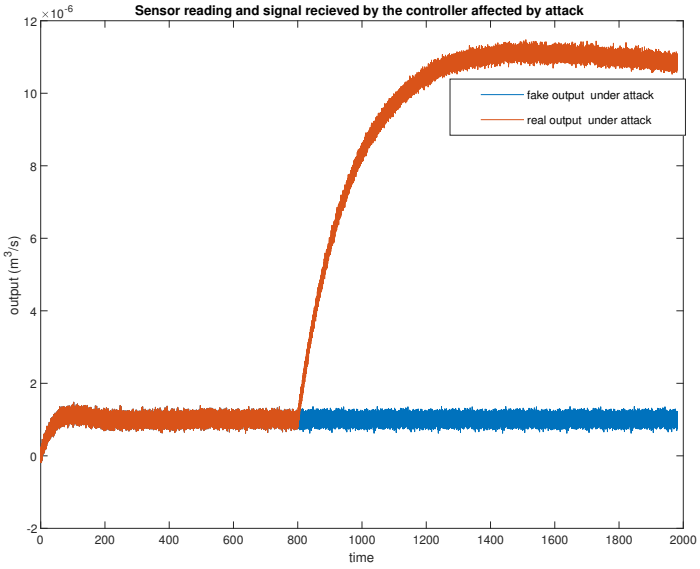
(b) Kalman filter result on state x (water level h)

Figure 4.31: Case 2: Kalman filter result for x, y in case of reference input and no watermarking

As can be seen, in this case, no attack is detected as it was expected. In the absence of watermarking, Kalman filter cannot detect the attack which occurs at $t = 800\ \ s$. The output in both cases with and without noise are shown in the following figures (Fig. 4.32).



(a) Real and fake output of system with attack, without noise



(b) Real and fake output of system with attack, with noise

Figure 4.32: Real and fake output of system under attack with/without noise

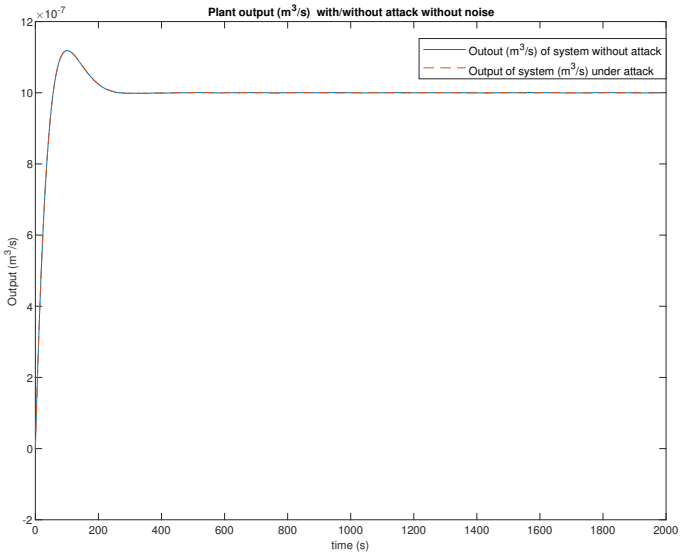
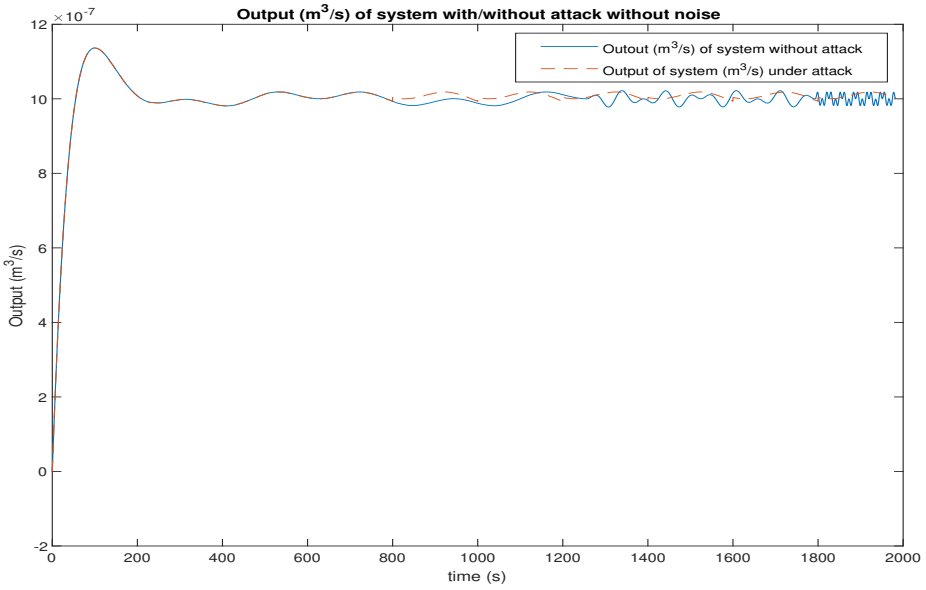


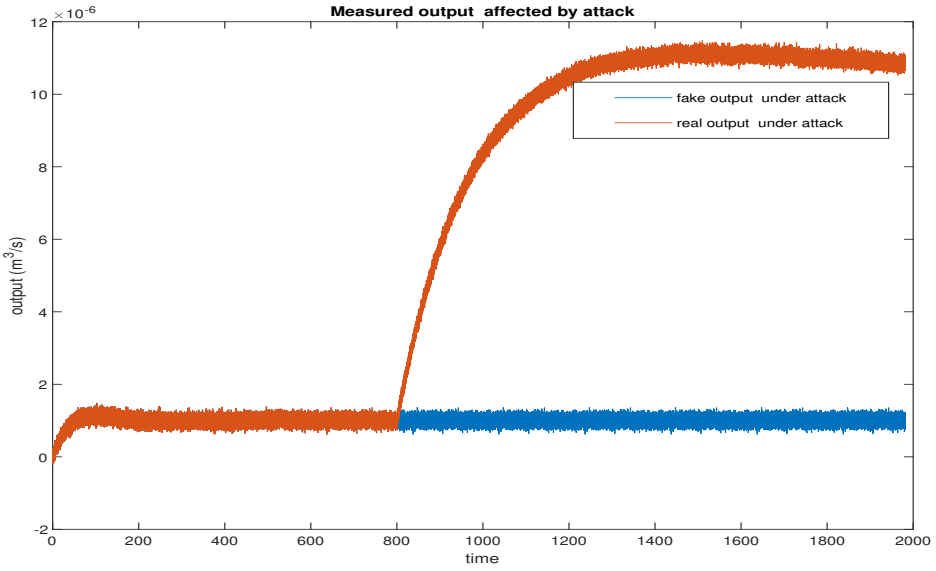
Figure 4.33: Fake output of plant with attack and real output of plant without attack

In Fig. 4.33, the repeated output signal which is produced by the attacker and the output without attack are shown.

- Case 3: In this case, it is assumed that reference input $r(t) = 0.1 \times 10^{-5}$ for $t > 0$ and also watermarking sine wave with phasing is injected to the closed loop system and noise is present. The results are presented in Fig. 4.34 and Fig. 4.35. Kalman filter results (Fig. 4.35) show that an attack is occurring".

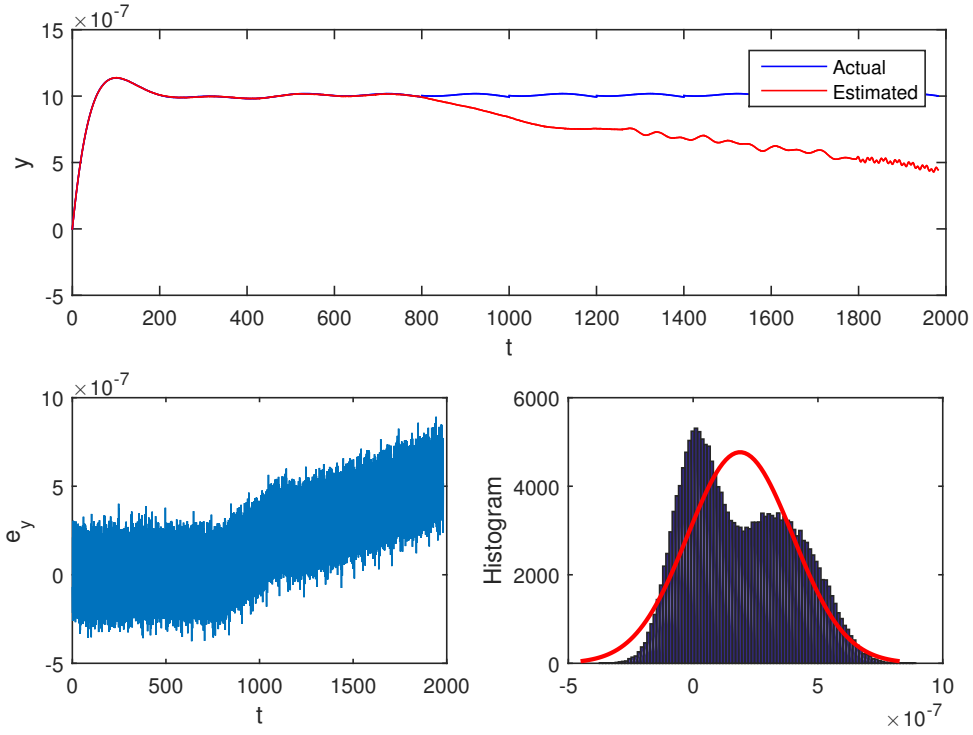


(a) Fake output of plant with attack and real output of plant without attack, with watermarking sin signal

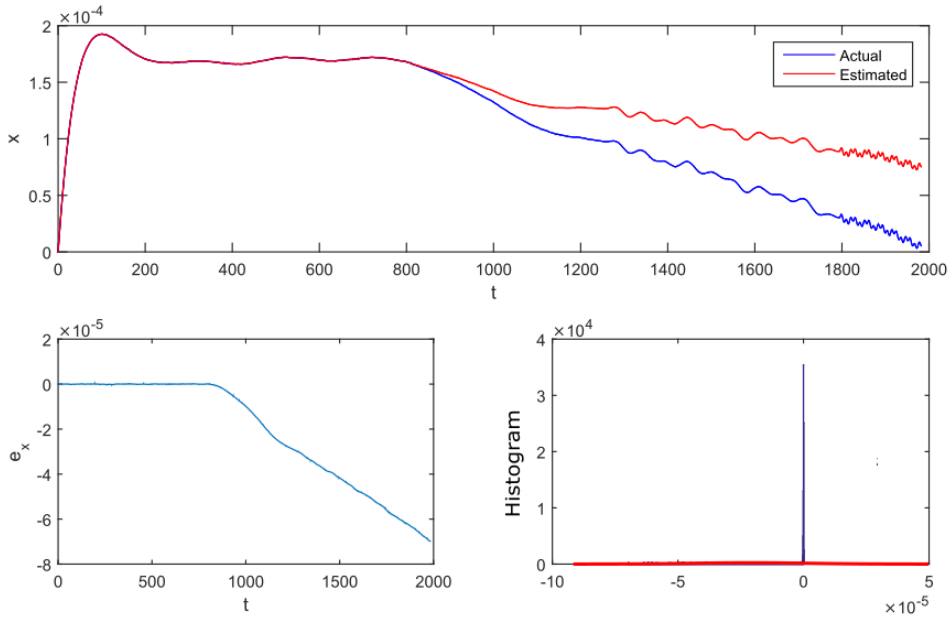


(b) Real and fake output of system with attack with watermarking sine wave, with noise

Figure 4.34: Plant output with/without attack and sensor output with attack, with watermarking sine wave



(a) Kalman result for y output



(b) Kalman result for x (hight of tank)

Figure 4.35: Case 3: Kalman result for y and x in case of input reference and watermarking with phasing

4.4.1 Conclusion

In this chapter, the model of a laboratory tank was studied (linear and non-linear), and based on its parameters, multi-sine watermarking signals were designed, and the effect on plant output was used using periodograms. Then replay attack was described and simulated, and the periodogram of output under attack was studied. From the results, it would be seen that the periodogram of output under attack does not show properly the watermarking frequencies. Therefore using this method we could easily recognize an attack was happening. Next, the use of Kalman filter along with watermarking was studied .If there was no watermarking, after attack started, the residual signals were near zero. But if watermarking was applied, during attack, the residuals become large and from that we could conclude that an attack was happening.

Chapter 5

Conclusion and Future Works

5.1 Conclusion

In this thesis, the problem of detecting replay attack in networked control systems was considered. Existing solutions to the attack detection problem were explained and their advantages and disadvantages were highlighted. Due to the importance of the research topic and the drawbacks of the available methods in the literature, a novel method was proposed in order to detect replay attack in networked control systems.

The proposed method is watermarking using multi-sine waves. The main advantage of this method is that it only requires the frequency response of the closed loop system at the watermarking frequencies. This information can be obtained experimentally. This also means that model of the system is not required. Another feature of this approach is that no assumption is made on the control law: it can be a PID, LQG or any other types of controller. Multi-sine wave are smooth and do not wear the actuators and the fluctuations that they cause in plant output can be easily calculated and limited.

5.2 Future Work

Some suggestions for future research in this area are outlined below:

- Studding modified periodogram, parametric periodogram methods for power spectral density instead of non-parametric Periodogram to determine the best approach to track watermarking signals.

- Instead of sinusoids, a white Gaussian IID signal also can be proposed. Instead of a perfect white Gaussian IID noise, a periodic white Gaussian IID noise, which in some intervals of time is zero, can be studied. Current results are with LQG controller. The research can be expanded to the other controllers such as PID controller.

- Using periodogram of the residual signal of Kalman filter to study the fault and attack situations.

Bibliography

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

[2] F. Pasqualetti, F. Dorfler, and F. Bullo, "Control-theoretic methods for cyberphysical security: Geometric principles for optimal cross-layer resilient control systems," *IEEE Control Systems*, vol. 35, no. 1, pp. 110–127, 2015.

[3] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[4] M. Bishop, *Computer security: art and science*. Addison-Wesley Professional, 2003.

[5] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 55, 2014.

[6] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: risk assessment, detection, and response," in *Proceedings of the 6th ACM symposium on information, computer and communications security*. ACM, 2011, pp. 355–366.

[7] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, and X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674–1683, 2018.

[8] G. A. Tsiamis, K. Gatsis, "State estimation codes for perfect secrecy," *2017 IEEE 56th Annual Conference on Decision and Control, CDC*, pp. 176–181, 2017.

[9] F. Miao, Q. Zhu, M. Pajic, and G. J. Pappas, "Coding schemes for securing cyber-physical systems against stealthy data injection attacks," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 106–117, 2017.

[10] Y. Mo, R. Chabukswar, and B. Sinopoli, "Detecting integrity attacks on scada systems," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1396–1407, 2014.

[11] S. Weerakkody, Y. Mo, and B. Sinopoli, "Detecting integrity attacks on control systems using robust physical watermarking," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 3757–3764.

[12] Y. Mo, T. H.-J. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber–physical security of a smart grid infrastructure," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 195–209, 2012.

[13] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*. IEEE, 2009, pp. 911–918.

[14] N. E. Wu and X. Wang, "A pulse-compression method for process monitoring," in *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*, vol. 3. IEEE, 2001, pp. 2127–2130.

[15] J. F. Hauer and J. G. DeSteese, "A tutorial on detection and characterization of special behavior in large electric power systems," Pacific Northwest National Lab.(PNNL), Richland, WA (United States), Tech. Rep., 2004.

[16] J. W. Pierre, N. Zhou, F. K. Tuffner, J. F. Hauer, D. J. Trudnowski, and W. A. Mittelstadt, "Probing signal design for power system identification," *IEEE Transactions on Power Systems*, vol. 25, no. 2, pp. 835–843, 2009.

[17] J. F. Hauer, W. A. Mittelstadt, K. E. Martin, J. W. Burns, H. Lee, J. W. Pierre, and D. J. Trudnowski, "Use of the wecc wams in wide-area probing tests for validation of system performance and modeling," *IEEE Transactions on Power Systems*, vol. 24, no. 1, pp. 250–257, 2009.

[18] N. Zhou, J. W. Pierre, and J. F. Hauer, "Initial results in power system identification from injected probing signals using a subspace method," *IEEE Transa ctions on Power Systems*, vol. 21, no. 3, pp. 1296–1302, 2006.

[19] C. M. Verrelli, P. Tomei, and E. Lorenzani, "Persistency of excitation and position-sensorless control of permanent magnet synchronous motors," *Automatica*, vol. 95, pp. 328–335, 2018.

[20] K. L. Morrow, E. Heine, K. M. Rogers, R. B. Bobba, and T. J. Overbye, "Topology perturbation for detecting malicious data injection," in *2012 45th Hawaii International Conference on System Sciences*. IEEE, 2012, pp. 2104–2113.

[21] F. Miao, M. Pajic, and G. J. Pappas, "Stochastic game approach for replay attack detection," in *Decision and control (CDC), 2013 IEEE 52nd annual conference on*. IEEE, 2013, pp. 1854–1859.

[22] F. Pasqualetti, F. Dörfler, and F. Bullo, "Cyber-physical attacks in power networks: Models, fundamental limitations and monitor design," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. IEEE, 2011, pp. 2195–2201.

[23] Y. Chen, S. Kar, and J. M. F. Moura, "Dynamic attack detection in cyber-physical systems with side initial state information," *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4618–4624, 2017.

[24] Z. Guo, D. Shi, K. H. Johansson, and L. Shi, "Optimal linear cyber-attack on remote state estimation," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 4–13, 2017.

[25] C. Schellenberger and P. Zhang, "Detection of covert attacks on cyber-physical systems by extending the system dynamics with an auxiliary system," in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE, 2017, pp. 1374–1379.

[26] S. Weerakkody and B. Sinopoli, "Detecting integrity attacks on control systems using a moving target approach," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE, 2015, pp. 5820–5826.

[27] M. Krotofil, A. Cardenas, J. Larsen, and D. Gollmann, "Vulnerabilities of cyber-physical systems to stale data—determining the optimal time to launch attacks," *International journal of critical infrastructure protection*, vol. 7, no. 4, pp. 213–232, 2014.

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

[29] E. Bou-Harb, "A brief survey of security approaches for cyber-physical systems," in *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2016, pp. 1–5.

[30] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—a survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, 2017.

[31] A. A. Cárdenas, S. Amin, and S. Sastry, "Research challenges for the security of control systems." in *HotSec*, 2008.

[32] M. Leccadito, T. Bakker, R. Klenke, and C. Elks, "A survey on securing uas cyber physical systems," *IEEE Aerospace and Electronic Systems Magazine*, vol. 33, no. 10, pp. 22–32, 2018.

[33] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, p. 13, 2011.

[34] H. Sandberg, A. Teixeira, and K. H. Johansson, "On security indices for state estimators in power networks," in *First Workshop on Secure Control Systems (SCS), Stockholm, 2010*, 2010.

[35] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[36] T. Irita and T. Namerikawa, "Detection of replay attack on smart grid with code signal and bargaining game," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 2112–2117.

[37] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure estimation and control for cyber-physical systems under adversarial attacks," *IEEE Transactions on Automatic control*, vol. 59, no. 6, pp. 1454–1467, 2014.

[38] Y. Mo, E. Garone, A. Casavola, and B. Sinopoli, "False data injection attacks against state estimation in wireless sensor networks," in *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 5967–5972.

[39] Y. Mo and B. Sinopoli, "False data injection attacks in cyber physical systems," in *First Workshop on Secure Control Systems*, 2010.

[40] M. Pajic, S. Sundaram, G. J. Pappas, and R. Mangharam, "The wireless control network: A new approach for control over networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2305–2318, 2011.

[41] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1495–1508, 2010.

[42] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 90–104, 2011.

[43] M. S. Mahmoud, M. M. Hamdan, and U. A. Baroudi, "Modeling and control of cyber-physical systems subject to cyber attacks: a survey of recent advances and challenges," *Neurocomputing*, vol. 338, pp. 101–115, 2019.

[44] Y. Yuan, Q. Zhu, F. Sun, Q. Wang, and T. Başar, "Resilient control of cyber-physical systems against denial-of-service attacks," in *Resilient Control Systems (ISRCS), 2013 6th International Symposium on*. IEEE, 2013, pp. 54–59.

[45] W. Lucia, B. Sinopoli, and G. Franze, "A set-theoretic approach for secure and resilient control of cyber-physical systems subject to false data injection attacks," in *Cyber-Physical Systems Workshop (SOSCYPS), Science of Security for*. IEEE, 2016, pp. 1–5.

[46] S. B. Rebaï, H. Voos, and S. A. S. Alamdari, "A contribution to cyber-physical systems security: an event-based attack-tolerant control approach," *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 957–962, 2018.

[47] W. Li, Y. Shi, and Y. Li, "Research on secure control and communication for cyber-physical systems under cyber-attacks," *Transactions of the Institute of Measurement and Control*, p. 0142331219826658, 2019.

[48] D. E. Miller and E. J. Davison, "An adaptive controller which provides an arbitrarily good transient and steady-state response," *IEEE Transactions on Automatic Control*, vol. 36, no. 1, pp. 68–81, 1991.

[49] J. C. Piquette, "Method for transducer transient suppression. i: Theory," *The Journal of the Acoustical Society of America*, vol. 92, no. 3, pp. 1203–1213, 1992.

[50] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "Revealing stealthy attacks in control systems," in *50th Annual Allerton Conference on Communication, Control, and Computing, Allerton, IL, USA, October 01-05, 2012*. IEEE conference proceedings, 2012, pp. 1806–1813.

[51] G. P. John, G. M. Dimitris, and G. Manolakis, "Digital signal processing: Principles, algorithms and applications," *Pentice Hall*, 1996.

[52] F. Auger and P. Flandrin, "Improving the readability of time-frequency and time-scale representations by the reassignment method," *Transactions on Signal Processing*, vol. 43, pp. 1068–1089, May 1995.

[53] S. A. Fulop and K. Fitz, "Algorithms for computing the time-corrected instantaneous frequency (reassigned) spectrogram, with applications," *the Acoustical Society of America*, vol. 119, pp. 360–371, 2006.

[54] X. Xie, D. Zhou, and Y. Jin, "Strong tracking filter based adaptive generic model control," *Journal of Process Control*, vol. 9, no. 4, pp. 337–350, 1999.

Appendix A

Appendix

A.1 Appendix I

```
% Simulates the application of input sine signals
% version: 15 Oct. 2019
% Azam_Ghamari


clear all;
clc;
close all;
gr=9.81;                % m/s^1
alpha=0.450289;
AA=0.0154;              % m^2
S=5*10^-5;          % m^2

% Operating point
%
```

```
16 h0=0.3;              % m
17 q10=alpha*S*sqrt(2*gr*h0);
18 q20=q10;
19
20 % Linearized model
21 %
22 beta=AA/(alpha^2 * S^2 * gr);
23 num=1;
24 den=[beta*q20  1];
25 g=tf(num,den);
26
27 % Controller
28 %
29 k=tf([5.5  0.1],[1  0]);
30 %%%%%%%%%%%%%%%%%%Noise
31 sigmaw=sqrt(0.02*10^-12);
32 sigmah=sqrt(0.01*10^-4);
33 sigmav=alpha*S*sqrt(gr/(2*h0))*sigmah;
34 Q=0.02*10^-12;  %processing noise
35 R=82.81*10^-16; %measurement noise
36 %  Q=0.02*10^-8;
37 %  R=82.81*10^-12;
38 sigmaw=sqrt(Q);
39 sigmav=sqrt(R);
40
41 % Transfer functions
42 gol=series(k,g);
```

```
43 Gry=feedback(gol,1);
44 Gdy=feedback(g,k);
45 Gvy=feedback(1,g*k);
46 Gru=feedback(k,g);
47 Gvyy=feedback(g*k,1);
48 %%%%% steady state of Gru
49 [num11,den11] = tfdata(Gru,'v');
50 [A11,B11,C11,D11]=tf2ss(num11,den11);
51 sysru=ss(A11,B11,C11,D11);
52 %%%%%%%%%%%%%%%%%Bode Diagram
53 stepinfo(Gry)
54 figure(1)
55 subplot(2,1,1)
56 bode(Gry)
57 legend('Bode of Gry')
58 subplot(2,1,2)
59 bode(Gdy)
60 legend('Bode of Gmy')
61
62
63
64 %%%%%%%%%%%%%%%%%%%%Considering Periodogram
65
66 w1=0.01;          % Frequencies of sines
67 w2=w1*3;
68 f1=w1/(2*pi);
69 f2=w2/(2*pi);
```

```
70 p1=(2*pi)/w1;      % Periods of sines
71 p2=(2*pi)/w2;
72 p=3*p2;              % Period of combined sine
73 tf=floor(1*p);            % duration of simulation
74
75 w3=0.07;             % Frequencies of sines
76 w4=w3*1.67;
77 f3=w3/(2*pi);
78 f4=w4/(2*pi);
79 p3=(2*pi)/w3;      % Periods of sines
80 p4=(2*pi)/w4;
81 pp=3*p3;              % Period of combined sine
82 tff=floor(1*pp);            % duration of simulation
83
84 w5=0.2;            % Frequencies of sines
85 w6=w5*3;
86 f5=w5/(2*pi);
87 f6=w6/(2*pi);
88 p5=(2*pi)/w5;      % Periods of sines
89 p6=(2*pi)/w6;
90 ppp=3*p6;              % Period of combined sine
91 tfff=floor(1*ppp);            % duration of simulation
92
93
94 [G1mag,G1ph]=bode(Gdy,w1);
95 [G2mag,G2ph]=bode(Gdy,w2);
96 A1=(0.07*sqrt(sigmav^2 + sigmaw^2))/G1mag;
```

```
%A1=sqrt(0.004*(sigmav^2 + sigmaw^2)/G1mag^2);
A2=(A1*G1mag)/G2mag;
phi1=(pi/2)- ((G1ph*pi)/180);
phi2=-(pi/2)- ((G2ph*pi)/180);


[G3mag,G3ph]=bode(Gdy,w3);
[G4mag,G4ph]=bode(Gdy,w4);
A3=(0.07*sqrt(sigmav^2 + sigmaw^2))/G3mag;
A4=(A3*G3mag)/G4mag;
phi3=(pi/2)- ((G3ph*pi)/180);
phi4=-(pi/2)- ((G4ph*pi)/180);


[G5mag,G5ph]=bode(Gdy,w5);
[G6mag,G6ph]=bode(Gdy,w6);
A5=(0.07*sqrt(sigmav^2 + sigmaw^2))/G5mag;
A6=(A5*G5mag)/G6mag;

phi5=(pi/2)- ((G5ph*pi)/180);
phi6=-(pi/2)- ((G6ph*pi)/180);

%%%%%%%%%%%%%%%%%%%%%t
tsample=0.01;
fs=1/tsample;
t1=0:tsample:2*tf;
t2=(t1(:,end)+tsample):tsample:2*(tf+tff);
```

```
124 t3=(t2(:,end)+tsample):tsample:2*(tf+tff+tfff);
125 nt1=size(t1,2);
126 nt2=size(t2,2);
127 nt3=size(t3,2);
128 t=[t1 t2 t3];
129 ts=0:tsample:(t(1,end));
130 nt4=size(t,2);
131 nt5=size(ts,2);
132 nt6=size(ts,2)-size(t,2);
133 nt7=size(t1,2)+size(t2,2);
134
135 %%%%%%%%%%%%%%%%%%%%%%%%Noise
136 v=sigmav*randn(size(ts));
137 w=sigmaw*randn(size(ts));
138 N=cov(v,w);
139 %%%%%%%%% sys openloop
140 [numo,deno] = tfdata(g,'v');
141 [Ao,Bo,Co,Do]=tf2ss(numo,deno);
142 syso=ss(Ao,Bo,Co,Do);
143 %%%%%%%%%%%%%Making Discrete
144 sysod = c2d(syso,1);
145 %[num111,den111] = tfdata(g,'v');
146 %sysod=ss(Ao,Bo,Co,Do,0.01);
147 Aod=sysod.a;
148 Bod=sysod.b;
149 Cod=sysod.c;
150 Dod=sysod.d;
```

```
151 Gd1 = c2d(g,0.01,'impulse');
152 kd1 = c2d(k,0.01,'impulse');
153 %%%%%%%% sys Controller
154 [numc,denc] = tfdata(k,'v');
155 [Ac,Bc,Cc,Dc]=tf2ss(numc,denc);
156 sysc=ss(Ac,Bc,Cc,Dc);
157 syscd=c2d(sysc,1);
158 %syscd=ss(Ac,Bc,Cc,Dc,1);
159 Acd=syscd.a;
160 Bcd=syscd.b;
161 Ccd=syscd.C;
162 Dcd=syscd.d;
163 %%%%%%%%%%%%%%%%%%%%%%%%%%close loop ss
164 [num,den] = tfdata(Gry,'v');
165 [A,B,C,D]=tf2ss(num,den);
166 sys=ss(A,B,C,D);
167 [b,a]=ss2tf(A,B,C,D);
168 % sys1=series(sysc,syso);
169 % sys=feedback(sys1,+1);
170 % [A,B,C,D]=ssdata(sys);
171 % [b,a]=ss2tf(A,B,C,D);
172 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
173 r=0.1*10^-5*ones(nt5,1);  %%%% delta input reference for
      linearized model
174 dsa=[A1*sin(w1*t1) + A2*sin(w2*t1)    A3*sin(w3*t2)+A4*sin(w4*t2)
    A5*sin(w5*t3)+A6*sin(w6*t3)];      % no phasing
```

```
dsb=[A1*sin(w1*t1+phi1) + A2*sin(w2*t1+phi2) A3*sin(w3*(t2-t2(1,1))+phi3)+A4*sin(w4*(t2-t2(1,1))+phi4) A5*sin(w5*(t3-t3(1,1))+phi5)+A6*sin(w6*(t3-t3(1,1))+phi6)];      % phasing for preventing transients
% dsa=[A1*sin(w1*t1) + A2*sin(w2*t1)    zeros(1,nt6)];     % no phasing
% dsb=[A1*sin(w1*t1+phi1) + A2*sin(w2*t1+phi2) zeros(1,nt6)];
X=10^-5*randn(nt5,1);
Y=X;
Y([2000:30000 4000:10000 50000:70000 110000:114000])=0;
% Y = sin((0 : ts(1:end))*pi/180);
% alpha= 0.96;
% Z = alpha* X + (1 -alpha)*Y;
%mean(autocorr(X))
%mean(autocorr(Z))
var(X)
var(Y)
ya=lsim(Gdy,dsa',ts)+lsim(Gdy,w',ts)+lsim(Gvy,v',ts);  %% y with noise without phasing
ya1=lsim(Gdy,Y,ts)+lsim(Gdy,w',ts)+10^-6*ones(size(t,2),1);  %%y without noise without phasing
%yb=lsim(Gdy,dsb',ts)+lsim(Gdy,w',ts)+lsim(Gvy,v',ts)+lsim(Gry,r,ts);  %%y  with phasing
%yb=lsim(Gdy,dsb',ts)+lsim(Gdy,w',ts)+lsim(Gvy,v',ts);
yb=lsim(Gdy,w',ts)+lsim(Gvy,v',ts)+lsim(Gry,r,ts)+lsim(Gdy,dsb',ts);
```

```
193 ybr=lsim(Gdy,w',ts)+lsim(Gvy,v',ts)+lsim(Gry,r,ts)+lsim(Gdy,X,ts)
    ;%%% y with rand n input
194 yb1=lsim(Gdy,w',ts)+lsim(Gdy,dsb',ts)+lsim(Gvyy,v',ts);  %%y
    without noise with phasing
195 z1=trapz(ts,yb.^2);  %%%%%%%%%% the area under curve output^2 =
    energy of signal
196
197 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
198 [u,~,xc]=lsim(sysc,-yb,ts,0);  %%%output of controller
199 [yo,~,xo]=lsim(syso,(u+dsb'+w'),ts,0);  %%% output of System
200 %%%%%%%%%%%%%% output without attack
201 figure(2)
202 plot(ts,yb1,ts,ya1,'--')
203 xlabel('time (s)')
204 ylabel('Output (m^3/s)')
205 title('Output (m^3/s) for 2 cases in 2 combined period in each
    interval without mesurement noise')
206 legend('with phasing', 'no phasing')
207 grid
208 figure(222)
209 plot(ts,yb,ts,ya,'--')
210 xlabel('time (s)')
211 ylabel('Output (m^3/s)')
212 title('Output (m^3/s) for 2 cases, with/without phasing in frame
    size = 2T_{combined} for each frame')
213 legend('with phasing','no phasing')
214 grid
```

```
%%%%%%%%%%%%%%%%%%Applying Replay Attack
%%%%%%%% U output to controller
%RR=r(800/tsample:size(ts,2)-1,1);
t4=800:tsample:ts(1,end);
KK=[yb(600*1/tsample:800*1/tsample,1);yb(600*1/tsample:800*1/
    tsample,1);yb(600*1/tsample:800*1/tsample,1);yb(600*1/tsample
    :800*1/tsample,1);yb(600*1/tsample:800*1/tsample,1);yb(600*1/
    tsample:((600*1/tsample)-6+mod((size(t,2)-(800*1/tsample))
    ,20000)),1)];
KK1=[yb1(600*1/tsample:800*1/tsample,1);yb1(600*1/tsample:800*1/
    tsample,1);yb1(600*1/tsample:800*1/tsample,1);yb1(600*1/
    tsample:800*1/tsample,1);yb1(600*1/tsample:800*1/tsample,1);
    yb1(600*1/tsample:((600*1/tsample)-6+mod((size(t,2)-(800*1/
    tsample)),20000)),1)];
uu=lsim(sysc,-KK,t4,xc(800/tsample));%%%%% u output of controller
     after attack
% %%%%%%%%%%%%%%%%%% U output of system
Dsb=dsb(1,(800/tsample:size(t,2)-1));
Dsb=Dsb';
W=w(1,1:(size(ts,2)-800/tsample))';
V=v(1,1:(size(ts,2)-800/tsample))';
%%%%%%%%%%plot output of the system
%ybb=lsim(syso,(uu+Dsb)+W,t4,xo(800/tsample));  %%%%%output of
    plant after attack
yr=lsim(syso,1.2*(uu+Dsb)+0.2*5.4*10^-5+W,t4,xo(800/tsample));%
   %%%%real output after attack
```

```
230 yrr=[yb((1:800/tsample),1);yr+v(1,(800/tsample):nt4-1).'];  %%%%%
      real output in whole of the time
231 %yws=[t4' ybs];%%%% for simulink
232 ybbb=[yb((1:800/tsample),1);KK]; %%%%output of system after
     sensor under attack
233 ybbs=[yb1((1:800/tsample),1);KK1];
234 figure(3)
235 plot(ts,ybbs,'b');
236 xlabel('time');
237 ylabel('output (m^3/s)');
238 title('y output before sensor affected by attack');
239 % h1 = line([800 800],[-2*10^-7 12*10^-7]);
240 % h2 = line([2000 2000],[-2*10^-7 12*10^-7]);
241  h1 = line([800 800],[-2*10^-8 2*10^-8]);
242  h2 = line([2000 2000],[-2*10^-8 2*10^-8]);
243
244 % Set properties of lines
245 set([h1 h2],'Color','k','LineWidth',1)
246 % Add a patch
247 gray = [0.7 0.7 0.7];
248 patch([800 2000 2000 800],[-2*10^-8 -2*10^-8 2*10^-8 2*10^-8],
     gray,'FaceAlpha', 0.5);
249 txt = 'Attack';
250 text(1000, 1.8*10^-8,txt,'FontSize',14)
251 %text(1000, 11*10^-7,txt,'FontSize',14)
252
253 figure(4)
```

```
254 plot(ts,ybbb,ts,yrr);
255 xlabel('time');
256 ylabel('output (m^3/s)');
257 title('y output after sensor affected by attack');
258 legend('y under attack after sensor',' the real output of system
       after sensor');
259
260 figure(5)
261 subplot(211)
262 plot(ts,yb1,ts,[yb1((1:800/tsample),1);yr],'--')
263 xlabel('time (s)')
264 ylabel('Output (m^3/s)')
265 title(' Real \Delta Output (m^3/s) of system with/without attack'
       )
266 legend('y output of system','y under attack')
267 grid
268 subplot(212)
269 plot(ts,yb1+5.4*10^-5*ones(size(t,2),1),ts,[yb1((1:800/tsample)
       ,1);yr]+5.4*10^-5*ones(size(t,2),1),'--')
270 xlabel('time (s)')
271 ylabel('Output (m^3/s)')
272 title('Real Output (m^3/s) of system with/without attack around
       operating point')
273 legend('y output of system','y under attack')
274 grid
275
276 figure(66)
```

```
277 plot(ts,yb1,ts,ybbs,'--');
278 xlabel('time (s)');
279 ylabel('Output (m^3/s)');
280 title('Output (m^3/s) of system with/without attack without
      mesurement noise');
281 legend('Outout (m^3/s) of system befor attack','Output of system
      (m^3/s) under attack');
282
283
284 figure(6)
285 plot(ts,yb,ts,ybr,'--');
286 z1=trapz(ts,yb);
287 z2=trapz(ts,ybr);
288 legend('y under sine','y under random signal')
289 %%%%%%%%%%Kalman filter
290 Plant = ss(Ao,[Bo Bo],Co,0,'inputname',{'u''w'},'outputname','y')
      ;
291 [kalmf,L,P] = kalman(Plant,Q,R);
292 %%%%%%%%%%u controller befor & after attack happens
293 t5=0:tsample:800-tsample;
294 U=[u((1:800/tsample),1);uu]+dsb';%%%%%%%%%%u output of controller
      in whole of the time +sin wave
295 yxhat=lsim(kalmf,[U ybbb],ts);
296 yhat=yxhat(:,1:size(Co,1));
297 xhat=yxhat(:,size(Co,1)+1:end);
298 [yoo,~,xoo]=lsim(syso,(U+w'),ts,0); %%%%%state of the plant in
      the whole of the time
```

```
299 %ey=ybbb-yhat;
300 ey=ybbb-yhat;
301 ex=xoo-xhat;
302 %%%%%%%%%%%%%%%%%%%%%%figures
303 figure(21)
304 subplot(2,2,[1 2]);
305 plot(t,ybbs(:,1),'b');
306 hold on;
307 plot(t,yhat(:,1),'r');
308 xlabel('t');
309 ylabel('y');
310 legend('Fake output received by controller','Estimated');
311
312 subplot(2,2,3);
313 plot(t,ey(:,1));
314 xlabel('t');
315 ylabel('e_y');
316
317 subplot(2,2,4);
318 histfit(ey(:,1),100);
319 ylabel('Histogram');
320
321 figure(22)
322 subplot(2,2,[1 2]);
323 plot(t,xoo(:,1),'b');
324 hold on;
325 plot(t,xhat(:,1),'r');
```

```
xlabel('t');
ylabel('x');
legend('Actual','Estimated');

subplot(2,2,3);
plot(t,ex(:,1));
xlabel('t');
ylabel('e_x');

subplot(2,2,4);
histfit(ex(:,1),100);
ylabel('Histogram');
%%%%%%%%%%%%%%%%%%%%%%%%%Periodogram of the First Interval
ya=ya-mean(ya);
yb=yb-mean(yb);
ybbb=ybbb-mean(ybbb);
flow=0.1*f1;
fup=1.5*f2;
nf=1000;
fstep=(fup-flow)/nf;
f=flow:fstep:fup;
%
figure(7)
[pxxa,fa]=periodogram(ya(1:(nt1)),[],f,fs);
plot(fa*2*pi,pxxa); hold on;
xlabel('rad/s')
ylabel('PSD')
```

```
353 title({'Periodogram of output for the first interval','w1=0.01,w2
      =0.03 rad/s, without phase'});
354 [pxxb,fb]=periodogram(yb(1:(nt1)),[],f,fs);
355 plot(fb*2*pi,pxxb)
356 grid
357 legend('Periodogram without phasing','Periodogram with phasing')
358 z=trapz(fa,pxxa);
359
360 figure(8)
361 [pxxb,fb]=periodogram(yb(1:(nt1)),[],f,fs);
362 plot(fb*2*pi,pxxb)
363 xlabel('rad/s')
364 ylabel('PSD')
365 title({'Periodogram of output for the first interval','w1=0.01,w2
      =0.03 rad/s, with phase'})
366 grid
367 set(gcf,'PaperPositionMode', 'auto');
368 saveas(gcf,'test.pdf');
369 %%%%%%%%%%%%%%%%%%%%%%%%%%Periodogram of the Second Interval
370 flow1=0.1*f3;
371 fup1=1.5*f4;
372 nf=1000;
373 fstep1=(fup1-flow1)/nf;
374 f11=flow1:fstep1:fup1;
375 %
376 figure(9)
377 [pxxaa,faa]=periodogram(ya(nt1:(nt1+nt2)),[],f11,fs);
```

```
plot(faa*2*pi,pxxaa)
xlabel('rad/s')
ylabel('PSD')
title({'Periodogram of output without phase for the second
    interval','w3=0.07,w4=0.11 rad/s'})
grid

figure(10)
[pxxbb,fbb]=periodogram(yb(nt1:(nt1+nt2)),[],f11,fs);
plot(fbb*2*pi,pxxbb);hold on;
plot(faa*2*pi,pxxaa)
xlabel('rad/s')
ylabel('PSD')
title({'Periodogram of output with phase for the second interval'
    ,'w3=0.07,w4=0.11 rad/s'})
grid
zz=trapz(fbb,pxxbb);
legend('with phase', 'without phase');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Periodogram of the Third Interval
flow2=0.1*f5;
fup2=1.5*f6;
nf=1000;
fstep2=(fup2-flow2)/nf;
f111=flow2:fstep2:fup2;
[pxxaaa,faaa]=periodogram(ya((nt1+nt2):(nt1+nt2+nt3)),[],f111,fs)
    ;
figure(11)
```

```
402 plot(faaa*2*pi,pxxaaa);
403 xlabel('rad/s')
404 ylabel('PSD')
405 title({'Periodogram of output without phase for the third
      interval','w5=0.2,w6=0.6 rad/s'})
406 grid
407 zzz=trapz(faaa,pxxaaa);
408 figure(12)
409 [pxxbbb,fbbb]=periodogram(yb((nt1+nt2):(nt1+nt2+nt3)),[],f111,fs)
      ;
410 plot(fbbb*2*pi,pxxbbb);hold on;
411 plot(faaa*2*pi,pxxaaa);
412 xlabel('rad/s')
413 ylabel('PSD')
414 title({'Periodogram of output with and without phase for the
      third interval','w5=0.2,w6=0.6 rad/s'})
415 grid
416 legend('with phase', 'without phase');
417
418 %%%%%%%%%%%%%%%%%%%%%%%%%%Attack, Periodogram of the First Interval
419 flow3=0.1*f1;
420 fup3=1.5*f6;
421 nf=1000;
422 fstep3=(fup3-flow3)/nf;
423 f3=flow3:fstep3:fup3;
424 figure(13)
425 [pxxat,fat]=periodogram(ya(40000:(nt1/2)),[],f3,fs);
```

```
426 plot(fat*2*pi,pxxat)
427 xlabel('rad/s')
428 ylabel('PSD')
429 title({'Periodogram of output from t=400','till the combined
      period for the first interval'})
430 grid
431
432 figure(14)
433 [pxxbt,fbt]=periodogram(ya(60000:(nt1+nt2/2)),[],f3,fs);
434 plot(fbt*2*pi,pxxbt)
435 xlabel('rad/s')
436 ylabel('PSD')
437 title({'Periodogram of output when the time is from t=600','till
      2*combined period of the first interval+the combined period of
       the second interval'})
438 grid
439
440 figure(15)
441 [pxxaat,faat]=periodogram(yb(1:(nt5)),[],f3,fs);
442 plot(faat*2*pi,pxxaat)
443 xlabel('rad/s')
444 ylabel('PSD')
445 title({'Periodogram of output in','duration of double period in
      each interval'})
446 grid
447 zzz=trapz(faat,pxxaat);
448
```

```
449 figure(16)
450 [pxxbbt,fbbt]=periodogram(yb(20000:0.6*(nt5)),[],f3,fs);
451 plot(fbbt*2*pi,pxxbbt)
452 xlabel('rad/s')
453 ylabel('PSD')
454 title({'Periodogram of output from t=200s','till 0.6*(duration of
        double period in each interval'})
455 grid
456
457
458 figure(17)
459 [pxxbbt,fbbt]=periodogram(yb(1:(nt1)/4),[],f,fs);
460 plot(fbbt*2*pi,pxxbbt)
461 xlabel('rad/s')
462 ylabel('PSD')
463 title({'Periodogram of output from t=0','till half the first
      combined period'})
464 grid
465 %%%%%%%%%Confidence Periodogram in the first interval of yb
466 figure(19)
467 [pxx1,f8,pxxc1]=periodogram(yb(1:(nt1)),rectwin(nt1),length(yb(1:
      nt1)),fs,'ConfidenceLevel',0.95);
468 plot(f8*2*pi,10*log10(pxx1)); hold on;
469 plot(f8*2*pi,10*log10(pxxc1),'r--','linewidth',2);hold on;
470 k1=floor( max(10*log10(pxx1)))*ones(1,length(f8));
471 plot(f8*2*pi,k1)
472 axis([0 1 min(min(10*log10(pxxc1))) max(max(10*log10(pxxc1)))]);
```

```
473 xlabel('rad/s')
474 ylabel('dB PSD')
475 title({'Periodogram with 0.95 confidence bound for the first
      interval in double of combined period for y'});
476 grid
477 legend('Peridogram', 'upper limit','lower limit','the first
      thereshold');
478 grid
479 %%%%%%%%%%%%%%%%%%%% Confidence Periodogram of the second interval
      of yb
480 figure(25)
481 [pxx5,f12,pxxc5]=periodogram(yb(nt1:nt1+nt2),rectwin(nt2+1),
      length(yb(nt1:(nt1+nt2))),fs,'ConfidenceLevel',0.95);
482 plot(f12*2*pi,10*log10(pxx5)); hold on;
483 plot(f12*2*pi,10*log10(pxxc5),'r--','linewidth',2);hold on;
484 k7=floor( max(10*log10(pxx5)))*ones(1,length(f12));
485 plot(f12*2*pi,k7)
486 axis([0 1 min(min(10*log10(pxxc5))) max(max(10*log10(pxxc5)))]);
487 xlabel('rad/s')
488 ylabel('dB PSD')
489 title({'Periodogram with 0.95 confidence bound for the second
      interval in double of combined period for y'});
490 grid
491 legend('Peridogram', 'upper limit','lower limit','the socond
      thereshold')
492 grid
493
```

```
494 %%%%%%%%%%%%%%%%%%%%%%%%%%Confidence Periodogram of the third interval
      of yb
495 figure(26)
496 [pxx6,f13,pxxc6]=periodogram(yb(nt1+nt2:nt1+nt2+nt3),rectwin(nt3
      +1),length(yb((nt1+nt2):(nt1+nt2+nt3))),fs,'ConfidenceLevel'
      ,0.95);
497 plot(f13*2*pi,10*log10(pxx6)); hold on;
498 plot(f13*2*pi,10*log10(pxxc6),'r--','linewidth',2);hold on;
499 k8=floor( max(10*log10(pxx6)))*ones(1,length(f13));
500 plot(f13*2*pi,k8)
501 axis([0 1 min(min(10*log10(pxxc6))) max(max(10*log10(pxxc6)))]);
502 xlabel('rad/s')
503 ylabel('dB PSD')
504 title({'Periodogram with 0.95 confidence bound for the third
      interval in double of combined period for y'});
505 grid
506 legend('Peridogram', 'upper limit','lower limit','the third
      thereshold')
507 grid
508 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Confidence Periodogram of total yb
509 figure(18)
510 [pxx,f7,pxxc]=periodogram(yb(1:(nt4)),rectwin(length(ybbs)),
      length(ybbs),fs,'ConfidenceLevel',0.95);
511 plot(f7*2*pi,10*log10(pxx)); hold on;
512 plot(f7*2*pi,10*log10(pxxc),'r--','linewidth',2);hold on;
513 k2=floor( max(10*log10(pxx1)))*ones(1,length(f7));
514 k3=floor( max(10*log10(pxx5)))*ones(1,length(f7));
```

```
k4=floor( max(10*log10(pxx6)))*ones(1,length(f7));
plot(f7*2*pi,k2);hold on;
plot(f7*2*pi,k3);hold on;
plot(f7*2*pi,k4);hold on;
axis([0 1 -190 -110]);
xlabel('rad/s')
ylabel('dB PSD')
title({'Periodogram with 0.95 confidence bound of total y'});
grid
legend('Peridogram', 'upper limit','lower limit','the first
    thereshold','the socond thereshold','the third thereshold')
grid
zzzz=trapz(f7,pxx);
%%%%%%%%%%%Confidential level for attack
figure(20)
[pxx2,f9,pxxc2]=periodogram(ybbb(40000:(nt1/2)),rectwin(nt1
    /2-40000),length(ybbb(40000:(nt1/2))),fs,'ConfidenceLevel'
    ,0.95);
plot(f9*2*pi,10*log10(pxx2)); hold on;
plot(f9*2*pi,10*log10(pxxc2),'r--','linewidth',2);hold on;
axis([0 1 min(min(10*log10(pxxc2))) max(max(10*log10(pxxc2)))]);
xlabel('rad/s')
ylabel('dB PSD')
title({'Periodogram with 0.95 confidence bound for output under
    attack since t=400s till 628s'});
legend('Peridogram', 'upper limit','lower limit','the first
    thereshold','the socond thereshold','the third thereshold')
```

```
grid
%%%%%%%%%%%%Confidential level for attack
figure(27)
[pxx7,f14,pxxc7]=periodogram(ybbb(20000:0.6*(nt5)),rectwin(0.6*(
    nt5)-20000),length(yb(20000:0.6*(nt5))),fs,'ConfidenceLevel'
    ,0.95);
plot(f14*2*pi,10*log10(pxx7)); hold on;
plot(f14*2*pi,10*log10(pxxc7),'r--','linewidth',2);hold on;
axis([0 1 min(min(10*log10(pxxc7))) max(max(10*log10(pxxc7)))]);
xlabel('rad/s')
ylabel('dB PSD')
title({'Periodogram with 0.95 confidence bound for output under
    attack since t=200s till 1113s'});

%%%%%%%%%%%%%%Confidence level of prtiodogram of the first
    interval of y
%%%%%%%%%%%%%%attack

figure(28)
[pxx8,f15,pxxc8]=periodogram(ybbb(1:nt1),rectwin(nt1),length(yb
    (1:nt1)),fs,'ConfidenceLevel',0.95);
plot(f15*2*pi,10*log10(pxx8)); hold on;
plot(f15*2*pi,10*log10(pxxc8),'r--','linewidth',2);hold on;
k9=floor( max(10*log10(pxx1)))*ones(1,length(f15));
plot(f15*2*pi,k9)
axis([0 1 min(min(10*log10(pxxc8))) max(max(10*log10(pxxc8)))]);
xlabel('rad/s')
```

```
559 ylabel('dB PSD')
560 title({'Periodogram with 0.95 confidence bound for the first
      interval in double of combined period for output under attack'
      });
561 grid
562 legend('Peridogram', 'upper limit','lower limit','the first
      thereshold');
563 grid
564 %%%%%%%%%Confidence Periodogram in the second interval of y
      attack
565 figure(23)
566 [pxx3,f10,pxxc3]=periodogram(ybbb(nt1:(nt1+nt2)),rectwin(nt2+1),
      length(ybbb(nt1:(nt1+nt2))),fs,'ConfidenceLevel',0.95);
567 plot(f10*2*pi,10*log10(pxx3)); hold on;
568 plot(f10*2*pi,10*log10(pxxc3),'r--','linewidth',2);hold on;
569 k5=floor( max(10*log10(pxx5)))*ones(1,length(f10));
570 plot(f10*2*pi,k5)
571 axis([0 1 min(min(10*log10(pxxc3))) max(max(10*log10(pxxc3)))]);
572 xlabel('rad/s')
573 ylabel('dB PSD')
574 title({'Periodogram with 0.95 confidence bound for the second
      interval in double of combined period for output under attack'
      });
575 grid
576 legend('Peridogram', 'upper limit','lower limit','the socond
      thereshold')
577 grid
```

```
%%%%%%%%%Confidence Periodogram in the third interval
figure(24)
[pxx4,f11,pxxc4]=periodogram(ybbb(nt1+nt2:(nt1+nt2+nt3)),rectwin(
    nt3+1),length(ybbb(nt1+nt2:nt1+nt2+nt3)),fs,'ConfidenceLevel'
    ,0.95);
plot(f11*2*pi,10*log10(pxx4)); hold on;
plot(f11*2*pi,10*log10(pxxc4),'r--','linewidth',2);hold on;
k6=floor( max(10*log10(pxx6)))*ones(1,length(f11));
plot(f11*2*pi,k6)
axis([0 1 min(min(10*log10(pxxc4))) max(max(10*log10(pxxc4)))]);
xlabel('rad/s')
ylabel('dB PSD')
title({'Periodogram with 0.95 confidence bound for the third
    interval in double of combined period for output under attack'
    });
grid
legend('Peridogram', 'upper limit','lower limit','the third
    thereshold')
grid
%%%%%%%%%%%%%%%Confidence periodogram of y under attack in whole
    period
figure(29)
[pxx9,f16,pxxc9]=periodogram(ybbb(1:nt5),rectwin(nt5),length(yb
    (1:nt5)),fs,'ConfidenceLevel',0.95);
plot(f16*2*pi,10*log10(pxx9)); hold on;
plot(f16*2*pi,10*log10(pxxc9),'r--','linewidth',2);hold on;
k2=floor( max(10*log10(pxx1)))*ones(1,length(f16));
```

```
598 k3=floor( max(10*log10(pxx5)))*ones(1,length(f16));
599 k4=floor( max(10*log10(pxx6)))*ones(1,length(f16));
600 plot(f16*2*pi,k2);hold on;
601 plot(f16*2*pi,k3);hold on;
602 plot(f16*2*pi,k4);hold on;
603 plot(f12*2*pi,k7)
604 axis([0 1 -200 -100]);
605 xlabel('rad/s')
606 ylabel('dB PSD')
607 title({'Periodogram with 0.95 confidence bound for the whole of
      double of combined period for output under attack'});
608 grid
609 legend('Peridogram', 'upper limit','lower limit','the first
      thereshold','the socond thereshold','the third thereshold')
610 grid
611 %%%%%%%%%%%%%%%%%%%%%%pburg periodogram
612 figure(30)
613 morder = 12;
614 ws=2*pi*fs;
615 pburg(yb,morder,[],fs)
616 %pburg(yb,morder,1024,ws)
617 %pburg(yb,morder,length(yb))
618 [px,ff]=pburg(yb,morder,[],fs)
619 % [___,pxxc] = pburg(yb,morder,,'ConfidenceLevel',1)
620  plot(2*pi*ff,10*log10(px))
```