Sensor Intrusion Detection in Control Systems Using Estimation Theory

Jiayi Su
Marquette University

SENSOR INTRUSION DETECTION IN CONTROL SYSTEMS USING ESTIMATION THEORY

by

JIAYI SU, B.S

A Thesis Submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Master of Science in Electrical and Computer Engineering.

Milwaukee, Wisconsin

May 2019

ABSTRACT

SENSOR INTRUSION DETECTION IN CONTROL SYSTEMS USING ESTIMATION THEORY

Jiayi Su, B.S.

Marquette University, 2018

In this thesis, two different approaches to sensor intrusion detection are presented. In the first approach, an estimation algorithm using a bank of Kalman Filters is designed that is capable of estimating the intrusion signal when sensors are affected in control systems. The mathematical models of the control system will be estabilished and the system measurement will be shown and after that, various false signals, such as constant-type and ramp-type signal, will be selected as the intrusion signal to affect the system output mentioned above. The system measurement will be tested based on a bank of Kalman Filters. The probabilities of each intrusion state (affected and unaffected) of the control system will be calculated as a function of time. The estimation of the states from a bank of Kalman Filters together with the associated probabilities will determine whether the sensor is under attack or not by using the information from the estimation algorithm. The performance of the algorithm will be tested based on the various levels of the system and measurement noise.

In the second approach, a new estimation algorithm is applied to detect the intrusion signal targeting the system mentioned above. By calculating the sample mean value of the system state and measurement in time, the changes of the system measurement can be detected by calculating the residual between the actual value and the theoretical sample mean value of the system measurement and in that case, the intrusion signal can be found. Thesis conclusions, summary and future work is also mentioned in the last chapter of this work.

ACKNOWLEDGEMENTS

Jiayi Su, B.S

I would like to express my sincere gratitude to my parents, Mr. Quanqi Su and Ms. Zhanying Mu, for their lifelong support and encouragement in all my endeavors. I would also like to thank my research advisors, Dr. Edwin Yaz and Dr. Susan Schneider for their discussions on estimation theory and various forms of Kalman Filter used throughout this thesis. This work couldn't be done without the countless hours of discussion on both technical aspects and writing style with them. Having such a well-versed group of faculties in the areas of sensors and estimation theory made it easy to check my ideas and arguments. I also want to thank my entire research committee (Dr. Jennifer Bonniwell, Dr. Edwin Yaz, and Dr. Susan Schneider) for reading, correcting and making suggestions on how to improve this thesis.

Special thanks to my friend, Dr. Karthick Sothivelr, for sharing his MATLAB knowledge.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Sensors are a critical part of feedback control systems, but they are volunerable to attacks in cyber-physical systems. Such attacks may cause significant damage to industrial control systems and this gives attackers a lot of chance to affect this important element. Thus, detection and protection against attack signals become a significant work to guarantee the proper operation of such systems. Estimation theory has been proposed for many years and, as one can expect, many researchers have expanded on it. One specific researcher, R.E. Kalman, came up with an approach to describe the discrete-data linear filtering problem [1]. The technique he developed could be the way to estimate system states and minimize system's disturbance and noise, which could be a great tool of detecting sensor intrusions. The development of the detection algorithm in this thesis utilizes this method. To begin, it is important to have a general background to understand how sensor intrusion happens and how to use this algorithm as a tool to make the detection be possible.

1.1 Sensor Intrusion

Sensors play an important role for measuring system states while also being vulnerable and sometimes exposed on an external environment, which makes it easy to be attacked. Therefore, the number of sensor intrusions has increased significantly with the development of the process control system. Sometimes sensor intrusion happens because the system operates under a harsh environment, like being exposed to extremely cold weather or to the sun for a long time, which makes the sensor unable to detect the correct

system measurement signal. Usually, intruders hack into the sensor, replace the system measurement with a false signal, which leads to a terrible result for the industrial process and may cause a malfunction or permanent damage to its constituents.

There are various types of sensor attacks that could influence system's performance, such as surge attacks, bias attacks and geometric attacks. Surge attacks allow intruders to achieve their maximum damage as soon as possible when they have access to the system. Bias attacks let attackers change the system output by adding a small disturbance over a large period of time. While geometric attacks let attackers try to switch the state of the system at the beginning of the attack and then maximize the damage after the system has been moved to a more vulnerable state [2].

A good example of intrusion targeting a control system is the Maroochy Shire Council's sewage control system in Queensland, Australia [3]. A hacker used a laptop and a radio transmitter to take control of 150 sewage pumping stations. Over a three-month period, he released one million liters of untreated sewage into a storm water drain from where it flowed into local waterways. The attack was motivated by revenge on the part of the hacker after he failed to secure a job with the Maroochy Shire Council. Unfortunately, ways to detect those attacks are still limited because attack signals are always hidden, which increases the difficulties of detection and observation of sensors intrusion, and there are some techniques that show it is impossible to estimate sensor and actuator intrusions under certain conditions [4].

Fortunately, estimation theory is widely applied for detecting and estimating system output and state, which makes it easier to develop a method of observing the attack signal.

1.2 Estimation Theory

Estimation theory is a branch of statistics that deals with estimating the values of parameters based on measured empirical data that has a random component. The estimation process could be done by using an estimator and historical data or measurements to observe unknown parameters in real applications [5]. Usually, there are three topics discussed under estimation theory, including smoothing, filtering and prediction. Smoothing is a method of estimating the unknown historical parameters by using current measurements. Filtering is a method of estimating the current unknown parameters by using known measurements and prediction, which is a way of estimating the future unknown parameters by using current measurements [6]. Problems with these three branches could be approached by using different estimation methods, such as Kalman Filter and its various derivatives, Particle Filter, Markov chain Monte Carlo (MCMC), Cramer-Rao Bound, Bayes estimators, Wiener Filter and Maximum likelihood estimators. Also, a huge number of applications of estimation theory using the methods mentioned above have been used in different technical areas as shown in Table 1.1.

Table 1.1: Applications of estimation theory [7].

Area of applications	Examples
Control Systems	Estimate the position of a powerboat for correcting navigation in the presence of sensor and environmental noise.
Communications	Estimate the carrier frequency of a signal for demodulation to the baseband in the presence of degradation noise.
Seismology	Estimate the underground distance of an oil deposit based on the different densities of oil and rock layers.
Biomedical	Estimate the heart rate of a fetus in the presence of environmental noise.
Image Processing	Estimate the position and orientation of an object from a camera image in the presence of lighting and background noise.
Radar Communications	Estimate the delay of the received pulse echo in the presence of noise.
Speech Signal Processing	Estimate the parameters of the speech model in the presence of speech/speaker variability and environmental noise.
Sensor Fault Detection	Estimate the sensor fault of the industrial control system in the presence of noise.

In this thesis, estimation theory will be used to solve the sensor intrusion problem and, the Kalman filter bank will be introduced and applied as the main estimation algorithm for the topic disused in chapter 2 and 4.

1.3 Previous Work Involving the Use of Estimation Theory

In 1978, R. N. Clark introduced a method of detecting incipient instrument fault [8]. The dedicated observer scheme (DOS) he introduced could be applied for estimating the lateral axis control system of a hydrofoil boat. He used several observers where each observer was designed for each sensor, and each observer could only receive its input

signal from the paired sensors. Also, the plant input could be received from all observers. In this case, the incipient fault could be detected if there is a fault input signal from a certain sensor while the other estimated signal will remain identical. The logic unit he used to make the decision of which sensor is affected is to set up a threshold value for each instrument and the false alarm will not be triggered if the residual of each instrument is less than the threshold value, otherwise the fault could be found and known by using this unit.

In 2003, T. Kobayashi and D. L. Simon introduced the application of a bank of Kalman filters for aircraft engine fault diagnostics [9]. They used multiple Kalman filters where each Kalman filter is designed for a specific sensor fault. When a fault comes through the sensor, all filters expect the one using a hypothesis similar to the faulty signal will show large errors, which could detect the unique sensor fault. Comparing to R. N. Clark's work, T. Kobayashi and D. L. Simon were calculating the weighted sum of squared residual (WSSR) for each filter and use WSSRs to compare with their pre-established thresholds. When a sensor is affected, every WSSRs expect the affected one will go beyond their thresholds, which means the affected one is found successfully based on their WSSR decision unit.

Similarly, W. Xue, Y. Guo and X. Dong applied the Kalman Filter bank as the main estimation algorithm to detect aircraft engine sensor and actuator intrusion in 2007 [10]. The basic logic of their fault detection and isolation is firstly calculate the residual value between low-pressure spool speed from sensors and estimated low-pressure spool speed

from observer measurements and, the second step is to compare the residuals with thresholds and as mentioned, all filters except the one using the correct hypothesis will produce large estimation errors, which could let the fault signal be isolated.

In 2011, D. H. Trinh and H. Chafouk applied the Kalman Filter bank technique to detect the intrusion signals in a wind turbine generator system [11]. The difference between the previous work is that they used a different decision unit to isolate the affected signal. A threshold was set firstly based on the estimated values and residuals, and then the Page-Hinkley's test was applied for the fault signal isolation. They claimed that using Page-Hinkley's test for the fault assessment is because its simplicity and it only needs low computational power.

In 2017, G. Rigatos, D. Serpanos and N. Zervos implemented the same technique on the power grid sensors fault detection [12]. After estimating systems states, calculating residuals and setting up thresholds for each sensor, they applied the χ^2 tests to isolate the fault signal. The results of the detection of the intrusion signal could be found by using χ^2 tests, and the highest scores of the χ^2 tests could show the compromised sensor.

In 2017, M. Rezaee, N. S-Nokhodberiz and J. Poshtan developed a method of using the Kalman Filter to detect and identify the sensor fault in an electro-pump system [13]. Similarly, as mentioned before, they calculated the estimated states and measurements of the electro-pump system and after that, they calculated the root mean

square error (RMSE) comparing to the system state and measurement. By setting up an upper bound of the RMSE, they could find if there's an intrusion signal in the electro-pump system.

In 2018, Y. Chen, S. Kar and J. M. F. Moura use the optimal attack strategy to attack the sensor and the controller they built in order to learn how a hacker could design an intrusion signal so that the attack signal could cause the maximum damage [14]. After knowing the optimal attack signal based on the system model, they also designed an estimation method, which use χ^2 tests to isolate the fault signal.

1.4 Scope of This Work and Main Contributions

This thesis proposes to develop a method to detect sensor intrusions in first-order and second-order discrete-time system that have disturbances both in the systems state and output. The distribution of the disturbances is proposed Gaussian and the intrusion signal is firstly proposed a constant-type and then a step and ramp-type on both first-order and second-order system outputs. A bank of Kalman Filters will be the main algorithm of estimating system state and output, which provides the basis for information available to know if the system is affected or not [19]. Mathematical models of control system will be established and various false signals, such as constant and ramp signal, will be selected and tested based on a bank of Kalman Filter. The probabilities of each state (affected/unaffected) of the control system will be calculated as a function of time.

The estimation of the states from a bank of Kalman Filters together with the associated probabilities will determine whether the sensor is under attack or not by using the data from the estimation algorithm. The performance of the algorithm will be tested based on the various levels of the system and measurement noise.

1.5 Thesis Organization

This thesis is comprised of five chapters. Chapter 2 consists of an introduction and derivation of the Kalman Filter and bank of Kalman Filters that have been proposed. Chapter 3 consists of system modes with attack signals. The mathematical models for the first-order and second-order discrete time systems with system and measurement noise will be established and both the systems will be affected with constant-type and ramp-type attack signal. Chapter 4 discusses the implementation of a bank of Kalman Filters both on the first and second order systems with different attack signals. The performance of the algorithm will then be tested based on the various levels of the system and measurement noise. Chapter 5 is a summary of the previous chapters and suggestions for future work.

2 A REVIEW OF ESTIMATION THEORY AND INTRODUCTION OF KALMAN FILTER

2.1 Introduction: Development of the Estimation Theory and Kalman Filter

As mentioned in Chapter 1, estimation theory is a branch of statistics that deals with estimating values of the states of a system based on measured empirical data that has a random component. By using an estimator with historical data or measurements, the estimation process could be used to estimate values of unknown parameters in real applications as introduced in Chapter 1 [5].

With the growth of computational power, it is easier to use an observer to estimate system states with a lot of measurement data. The Kalman filter (KF), as one of the estimation algorithms, is developed to estimate system states and measurements in a lot of fields. In this thesis, the sensor intrusion problem could be solved by using Kalman Filter and one of its extensions, a bank of Kalman filter (BKF), to detect the changes of the systems measurements and find the intrusion when there is an attack signal enters the system and replaces the system measurement. Both the Kalman filter and a bank of Kalman filters (KF and BKF) will be introduced and derived in this thesis and the algorithms for applying these estimation methods into the sensor intrusion problems will be shown. Some of the typical applications in different areas will also be given in section 2.2

In fact, most of these modern estimation-theory-based techniques can be found at the heart of many electronic signal processing systems designed to extract information [16]. Typical application areas and example applications in areas utilizing estimation theory are listed in Table 2.1 [18].

2.2 Kalman Filter

The Kalman filter, also known as linear quadratic estimation (LQE), is a method of estimating the unknown parameters and states of a system with statistical noise. It can produce the estimated values of unknown variables and can also minimizes the mean of the error. There are a huge amont of applications of using Kalman Filter in many different areas, such as tracking problems, navigation problems, signal processing problems and even in economics. Moreover, the Kalman filter is also a main topic in robotic motion, where its used to optimize the trajectory of the motions. At the same time, as one the estimation algorithms, as mentioned in Chapter 1, the Kalman filter could be used to estimate not only the present state by giving the known measurements, but the past and the future states of a system by some changes to the filter.

Basically, the Kalman filter works with two steps, the first step is called prediction step or time update step, where it could produce the current state estimate together with its associated noise value [15]. The second step is called measurement update step or correct step, where the measurement could be updated using a weighted

average, which could minimize the uncertainty of the measurement. Figure 2.1 shows the process of the two steps below.

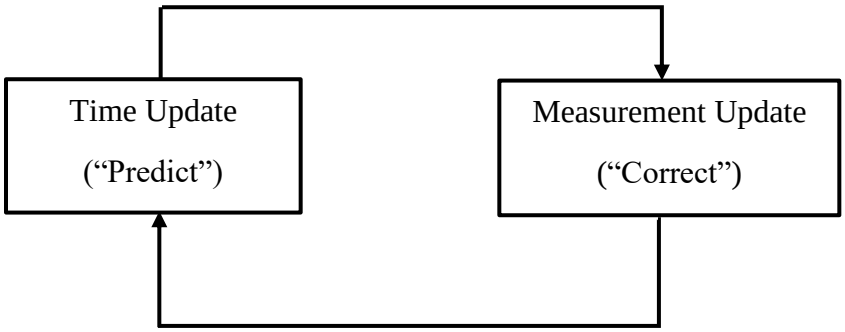

Figure 2.1: Process of the two steps of the Kalman Filter [15]

The Kalman filter is used to reduce the noise in systems states and outputs in numerous applications and the noise is assumed Gaussian on most of the applications. The Kalman filter can also work if the noise disturbance is not Gaussian.

The Kalman filter is named after R.E. Kalman, one of the primary developers of its theory. In 1960, R.E. Kalman first developed a method of a recursive algorithm to deal with the discrete-time linear filtering problem [1,16]. The recursive algorithm means the estimated value $\hat{x}_{k+1}$ can be calculated by using the previous estimated value $\hat{x}_k$. Later on, the development of various extensions on the Kalman filter have been derived targeting different kinds of problems and applications, especially for systems within the security field. Nowadays it has been widely applied in engineering problems, mathematical problems, biomedical problems and even economic problems, and most of

the problems can be solved properly because of this technique. In some of the applications, Kalman filter is a crucial technique and one cannot solve it without using this technique. Some typical examples of applications by using Kalman filter are introduced in Table 2.2.

Table 2.1: Typical applications of various forms of Kalman filter [18].

Area of applications	Examples
Navigation	To control and assist the navigation of automobiles, aircraft or spacecraft using the measured sensor data in the environment with noise and disturbance [21].
Image processing	Using various forms of Kalman filter to estimate the position and orientation of an object from a camera image in the presence of lighting and background noise.
Radar communications	Estimating the distance/velocity of the target object by various forms of the Kalman filter.
Control system	Active noise control in control systems [15].
Economics	Parameter estimation of linear or non-linear econometric models [22].
Speech signal processing	To estimate the parameters of the speech model and to get rid of the noise out of the speech signal.
Forecasting	Estimating the parameters of the forecasting model using the historical data.
Sensor Fault Detection	To estimate the sensor fault of the industrial control system in the presence of noise.

2.2.1 Derivation of Kalman Filter

Consider a linear discrete-time stochastic system with system states $x_k \in \mathbb{R}^n$, system measurements $y_k \in \mathbb{R}^p$, system inputs $u_k \in \mathbb{R}^m$ and system matrices A_k, B_k, C_k and D_k, where A_k, B_k, C_k and D_k are all time-varying matrices,

$$x_{k+1} = A_k x_k + B_k u_k + F_k v_k \quad (2.1a)$$

$$y_k = C_k x_k + D_k u_k + G_k w_k \quad (2.1b)$$

in (2.1a), v_k is the system state noise, where the covariance of the noise is V_k, and w_k is the system measurement noise, where the covariance of the noise is W_k. S_k is the cross covariance, where it is between the covariance of the state noise V_k and the covariance of the measurement W_k. The system state noise vector v_k, system measurement noise vector w_k and the initial state value of the system x_0 can be expressed with arbitrary densities below,

$$\begin{bmatrix} x_0 \\ v_k \\ w_k \end{bmatrix} \sim \left(\begin{bmatrix} \bar{x}_0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} X_0 & 0 & 0 \\ 0 & V_k & S_k \\ 0 & S_k^T & W_k \end{bmatrix} \right)$$

In this thesis, the sensor intrusion detection problem, x_k represent the system states, where it needs to be observed by Kalman filter, and y_k is the system outputs. Suppose u_k represents the unit step input of the system, then the system estimated states x_k could be known by using a Kalman filter with the system outputs as long as the system is observable.

Before deriving the Kalman filter, it is necessary to assume an observer that could estimate the system state at time $k+1$, where the estimated state should be $\hat{x}_{k+1}$. After assuming an observer, some other information at time k, will also be needed to derive the

Kalman filter. The first information will be the present estimate states $\hat{x}_k$, also the current input u_k and the current system outputs y_k should be available. After knowing these three pieces of information, an observer could be given in (2.2).

$$\hat{x}_{k+1} = A_k\hat{x}_k + B_k u_k + K_k(y_k - \hat{y}_k) \tag{2.2}$$

where $\hat{y}_k$ is the estimate of the system output given by (2.3),

$$\hat{y}_k = C_k\hat{x}_k + D_k u_k \tag{2.3}$$

K_k is the Kalman gain, which minimizes the variances of the error, and the error is defined as the residual of the true state and the estimated state, which is given by (2.4)

$$e_{k+1} = x_{k+1} - \hat{x}_{k+1} \tag{2.4}$$

The estimated value of the unknown states is unbiased (i.e. $E\{e_{k+1}\} = 0$). The error covariance, which is defined as $P_{k+1} = E\{(e_{k+1})(e_{k+1})^T\}$, needs to be found before getting the Kalman gain, K_k. While from the definition of the error covariance, some relationship between system states, system matrices and error covariance could be found.

First, substitute (2.1b) and (2.3) into (2.2), resulting in the estimated state $\hat{x}_{k+1}$ expressed below,

$$\hat{x}_{k+1} = A_k\hat{x}_k + B_k u_k + K_k[(C_k x_k + D_k u_k + G_k w_k) - (C_k\hat{x}_k + D_k u_k)] \quad (2.5a)$$

Next, the error between the true state x_{k+1} and estimated state $\hat{x}_{k+1}$ can be expressed by submitting (2.1a) and (2.5a),

$$e_{k+1} = \{A_k x_k + B_k u_k + F_k v_k\} - \{A_k\hat{x}_k + B_k u_k + K_k[(C_k x_k + D_k u_k + G_k w_k) - (C_k\hat{x}_k + D_k u_k)]\} \quad (2.5b)$$

After some transformation, (2.5b) can be shown as (2.5),

$$e_{k+1} = (A_k - K_k C_k)e_k + F_k v_k - K_k G_k w_k \quad (2.6)$$

Substituting (2.5) into the definition of the error covariance yields

$$P_{k+1} = E\{[(A_k - K_k C_k)e_k + F_k v_k - K_k G_k w_k] [(A_k - K_k C_k)e_k + F_k v_k - K_k G_k w_k]^T\} \quad (2.7)$$

After some transformation on (2.7), the error covariance equation can be found as

$$P_{k+1} = A_k P_k A_k^T - A_k P_k C_k^T K_k^T - K_k C_k P_k A_k^T + K_k C_k P_k C_k^T K_k^T + F_k V_k F_k^T - K_k G_k S_k^T F_k^T - F_k S_k G_k^T K_k^T + K_k G_k W_k G_k^T K_k^T \quad (2.8)$$

After finding the error covariance equation, it is possible to derive the Kalman gain, K_k. During the process of deriving the error covariance P_{k+1}, there is an important property that needs to be noticed and that is P_{k+1} is a symmetric posite definite matrix. Thus, by using this property, one can minimize the error covariance to find the Kalman gain K_k, and this could be transformed to minimize the trace ($Tr\{P_{k+1}\}$) of the error covariance matrix P_{k+1}. Therefore, there is a way of getting Kalman gain by taking the partial derivative of the trace of the error covariance $Tr\{P_{k+1}\}$ with respect to K_k. After taking the partial derivative of $Tr\{P_{k+1}\}$, one can let the partial derivative equation equal zero to get the expression of the Kalman gain K_k. [17] The equation of the partial derivative of the trace of the error covariance $Tr\{P_{k+1}\}$ can be expressed as (2.9),

$$\frac{\delta\, Tr\{P_{k+1}\}}{\delta\, K_k} = -2A_k P_k C_k^T - 2F_k S_k G_k^T + 2K_k(C_k P_k C_k^T + G_k W_k G_k^T) \tag{2.9}$$

Setting the partial derivative equation (2.9) equal to zero, the Kalman gain could be found as below,

$$K_k = (A_k P_k C_k^T + F_k S_k G_k^T)(C_k P_k C_k^T + G_k W_k G_k^T)^{-1} \tag{2.10}$$

As mentioned previously, the Kalman gain minimizes the error covariance in time, so the error covariance equation (2.8) is simplified after substituting (2.10) into it. Then the error covariance could be expressed as

$$P_{k+1} = A_k P_k A_k^T + F_k V_k F_k^T - (A_k P_k C_k^T + F_k S_k G_k^T)(C_k P_k C_k^T + G_k W_k G_k^T)^{-1} (C_k P_k A_k^T + G_k S_k^T F_k^T) \quad (2.11)$$

If the system state noise and system measurement noise are white noise, which is the most commonlt the case for most of the system, including the system considered in this thesis, then their values will be uncorrelated with each other [18] and because the system state noise and system measurement noise are uncorrelated with each other, the cross-covariance, S_k, will be zero, then the system state noise vector v_k, system measurement noise vector w_k and the initial state value of the system x_0 are independent white random variables with arbitrary densities, which could be expressed as below [19]:

$$\begin{bmatrix} x_0 \\ v_k \\ w_k \end{bmatrix} \sim \left(\begin{bmatrix} \bar{x}_0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} X_0 & 0 & 0 \\ 0 & V_k & 0 \\ 0 & 0 & W_k \end{bmatrix} \right) \quad (2.12)$$

The expression of the Kalman gain K_k and the error covariance P_{k+1} can be then simplified as (2.13) and (2.14) if the cross-covariance $S_k = 0$,

$$K_k = A_k P_k C_k^T (C_k P_k C_k^T + G_k W_k G_k^T)^{-1} \quad (2.13)$$

$$P_{k+1} = A_k P_k A_k^T + F_k V_k F_k^T - A_k P_k C_k^T (C_k P_k C_k^T + G_k W_k G_k^T)^{-1} (C_k P_k A_k^T) \quad (2.14)$$

After finding the expression of the Kalman gain and the error covariance, the state update equation can be shown as (2.15),

$$\hat{x}_{k+1} = A_k\hat{x}_k + B_k u_k + K_k\tilde{y}_k \tag{2.15}$$

Where $\tilde{y}_k$ is the innovation term, which is the difference between the system output y_k and the estimated output $\hat{y}_k$ at each time k and it could be shown below

$$\begin{aligned}\tilde{y}_k &= y_k - \hat{y}_k \\ &= y_k - (C\hat{x}_k + Du_k)\end{aligned} \tag{2.16}$$

From (2.13), (2.14) and (2.15), the recursive algorithm to calculate the system state estimate is designed. This algorithm works recursively according to the measurement state at every time step k and, because of its recursive nature, the only information that the Kalman filter needs to know are the current estimate states $\hat{x}_k$, the input u_k and the measurement states y_k for calculating the updated estimated value $\hat{x}_{k+1}$. The advantage of this recursive algorithm is that there is no need to store the past measurements because it only requires the last "best guess" to do the estimation rather than the entire historical data.

2.2.2 Kalman Filter Algorithm

As mentioned previously, the Kalman filter is a recursive estimation algorithm, where it only needs the latest estimate of the states and the measurement states to calculate the updated state estimate. After deriving the Kalman filter, an introduction will be shown on how this recursive algorithm works.

When implementing the Kalman filter algorithm, it is necessary to make sure the systems matrices A_k, B_k, C_k and D_k are known and, the value of the measurement noise covariance W_k and the value of state noise covariance V_k are available. After making sure the systems matrices, the measurement noise covariance and the state noise covariance are all available, the next step is to assume the value of initial systems state estimate $\hat{x}_0$ and the initial error covariance P_0. Basically, $\hat{x}_0$ and P_0 needs to be set up based on the situation. For example, if the system's uncertainty is extremely high, then the initial error covariance P_0 needs to be set up at a relatively high value, so that the Kalman filter could work "harder" to decrease the uncertainty of the system [1,18]. Also, the initial state estimate $\hat{x}_0$ needs to be set up within a reasonable range so that the Kalman filter could work properly. Once $\hat{x}_0$ and P_0 are set up, the next step is to find the Kalman gain K_0 where it could be found by using (2.13). After finding the Kalman gain K_0, the state estimate $\hat{x}_1$ and the error covariance P_1 could be updated with the associated system measurement y_0 by using (2.14) and (2.15). It could be noticed that the process above happens at time $k = 0$. The updated $\hat{x}_1$ and P_1 could be used as the initial value at time $k = 1$ to calculate the new Kalman gain K_1, after finding the new gain, repeat the process again until the error covariance P_k becomes small or the measurement is taken at time k [18]. This process could be shown as Fig 2.2 below.

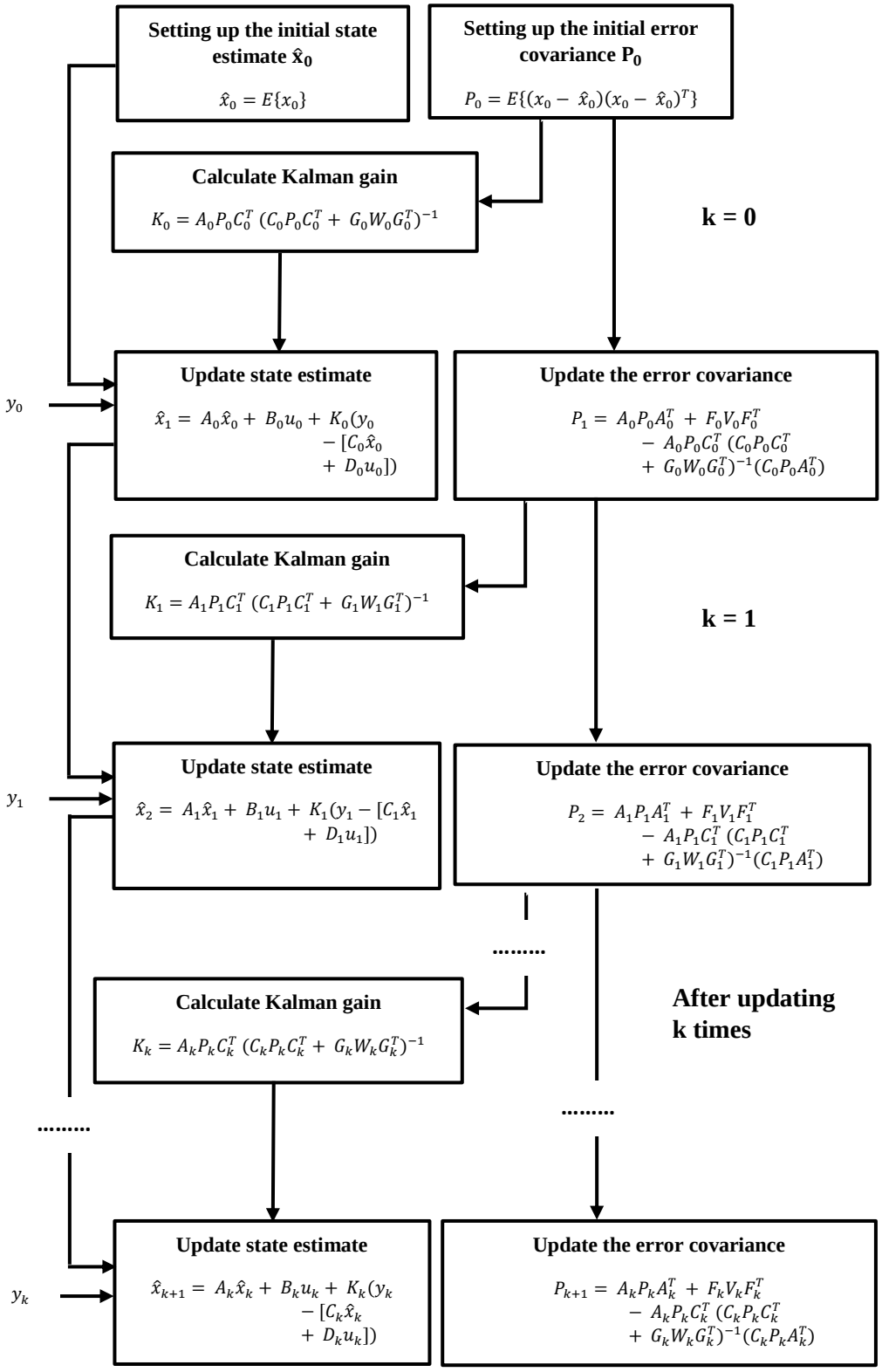

Figure 2.2: Flowchart of Kalman filter algorithm

2.3 A Bank of Kalman Filters

When systems matrices A_k, B_k, C_k and D_k are known and, the value of measurement noise covariance W_k and the value of state noise covariance V_k are available, it is easy to obtain the state estimate by implementing the Kalman. On the other hand, the process above would not be so easy when there are some uncertainties in the system model. For example, consider the sensor intrusion problem, a hacker come into the system and then modifies the system measurement state by replacing the state signal $C_k e_k$ with another one, then it will be hard for letting the Kalman filter to obtain the precise value of the system state estimate even with a large initial error covariance P_k and an educated guess of the initial state estimate $\hat{x}_0$.

In this technique, the parameter of a system can be adaptively estimated if the assumptions of the parameter can be made properly. Suppose the unknown parameter belongs to a discrete set which has known upper and lower bounds, and this set includes N values where each value is a possible value or a hypothesis for the unknown parameter, then the set of each possible values or hypothesis could be represented as $\theta = \{\theta_1, \theta_2, \dots, \theta_i, \dots, \theta_N\}$. So, N number of Kalman filters can be designed specifically corresponding to each possible values of the unknown parameter. After knowing each possible hypothesis for the unknown parameter, the next step is to calculate the conditional probabilities for each hypothesis based on the Bayes' rule and, after that the specific Kalman filter with a conditional probability that is closest to one represents the most probable value of the unknown parameter [19, 20].

2.3.1 Derivation of a bank of Kalman filters

Knowing the possible values of the unknown parameter can be represented as $\theta = \{\theta_1, \theta_2, \dots, \theta_i, \dots, \theta_N\}$, Bayes' rule can be used as follows

$$p(\theta_i|Y_k) = \frac{p(Y_k, \theta_i)}{p(Y_k)} = \frac{p(Y_k|\theta_i)p(\theta_i)}{\sum_{i=1}^{N} p(Y_k|\theta_i)p(\theta_i)} \tag{2.17}$$

Here, $p(\theta_i|Y_k)$ represent the conditional probabilities of each hypothesis θ_i and Y_k represents all the system measurements up throuth time instant *k*. The $p(Y_k|\theta_i)$ are defined as the likelihood functions for each hypothesis and they are used for the recursive calculation of a bank of conditional Kalman filters [19]. (2.17) is further expanded and simplified as

$$\begin{aligned} p(\theta_i|Y_k) &= \frac{p(y_k, Y_{k-1}, \theta_i)}{p(y_k, Y_{k-1})} \\ &= \frac{p(y_k, \theta_i|Y_{k-1})p(Y_{k-1})}{p(y_k|Y_{k-1})p(Y_{k-1})} \\ &= \frac{p(y_k, \theta_i|Y_{k-1})}{p(y_k|Y_{k-1})} \\ &= \frac{p(y_k|Y_{k-1}, \theta_i)p(\theta_i|Y_{k-1})}{\sum_{i=1}^{N} p(y_k|Y_{k-1}, \theta_i)p(\theta_i|Y_{k-1})} \end{aligned} \tag{2.18}$$

where y_k represents the system measurement at time k, Y_{k-1} represents all system measurements from $k = 1$ through $k - 1$ and, as mentioned above, θ_i represents the possible value for the unknown parameter where each Kalman filter is designed specifically corresponding to each θ_i. This equation can be solved recursively, and the calculation could begin with an assumed probability $p(\theta_i|Y_0)$ between 0 and 1 when $k = 0$, where the sum of the probabilities is one. Note that $p(\theta_i|Y_{k-1})$ is the previous value of $p(\theta_i|Y_k)$.

After that, the most important step is to calculate $p(y_k|Y_{k-1}, \theta_i)$ where it is part of the probability density function in (2.18). In this work, system state noise and measurement noise are all assumed to have Gaussian distribution, which produces Gaussian conditional probabilities. Therefore, $p(y_k|Y_{k-1}, \theta_i)$ could be represented as (2.19) because the density function of Gaussian is known

$$p(y_k|Y_{k-1}, \theta_i) = (2\pi)^{-m/2}\left|\Omega_{k|\theta_i}^{-1}\right|^{1/2} \exp\left\{-\frac{1}{2}\tilde{y}_{k|\theta_i}^T \Omega_{k|\theta_i}^{-1}\tilde{y}_{k|\theta_i}\right\} \tag{2.19}$$

where m is the order of the system, $\tilde{y}_{k|\theta_i}$ is the innovation sequence where each Kalman filter is responsible for estimation based on its corresponding hypothesis θ_i

$$\tilde{y}_{k|\theta_i} = y_k - \hat{y}_{k|k-1,\theta_i} \tag{2.20}$$

and $\Omega_{k|\theta_i}$ is the innovation covariance for each Kalman filter with its corresponding hypothesis where it could be calculated from below

$$\Omega_{k|\theta_i} = C_k P_{k|\theta_i} C_k^T + G_k W_k G_k^T \quad (2.21)$$

Therefore, the conditional probability of each Kalman filter can be found using equation (2.18), (2.19), (2.20) and (2.21). The convergence will occur when there is a hypothesis closest to the correct value and that probability will be equal to one for this assumption while all the probabilities of other possible values of θ_i will go to zero [20].

2.3.2 Algorithm of a bank of Kalman filters

Suppose $\theta = \{\theta_1, \theta_2, \dots, \theta_i, \dots, \theta_N\}$ where N represents the quantities of possible values for the unknown parameter, where the upper and lower bounds can be defined as θ_1 and θ_N, which means the possible values of the unknown parameter is included in this range. After knowing the set of the hypotheses, a bank of Kalman filters is set up where each Kalman filter is designed specifically with its associated hypothesis and, then $\tilde{y}_{k|\theta_i}$ and $\Omega_{k|\theta_i}$ could be calculated by substituting the estimated measurement $\hat{y}_{k|k-1,\theta_i}$ and the error covariance $P_{k|\theta_i}$ to equation (2.20) and (2.21). After knowing $\tilde{y}_{k|\theta_i}$ and $\Omega_{k|\theta_i}$ for each possible value of the unknown parameter, the conditional probabilities $p(\theta_i|Y_k)$ can be calculated recursively using equation (2.18) and the one which is closest to one represents the true value of the unknown parameter. Fig 2.3 shows the flowchart of the bank of Kalman filters algorithm.

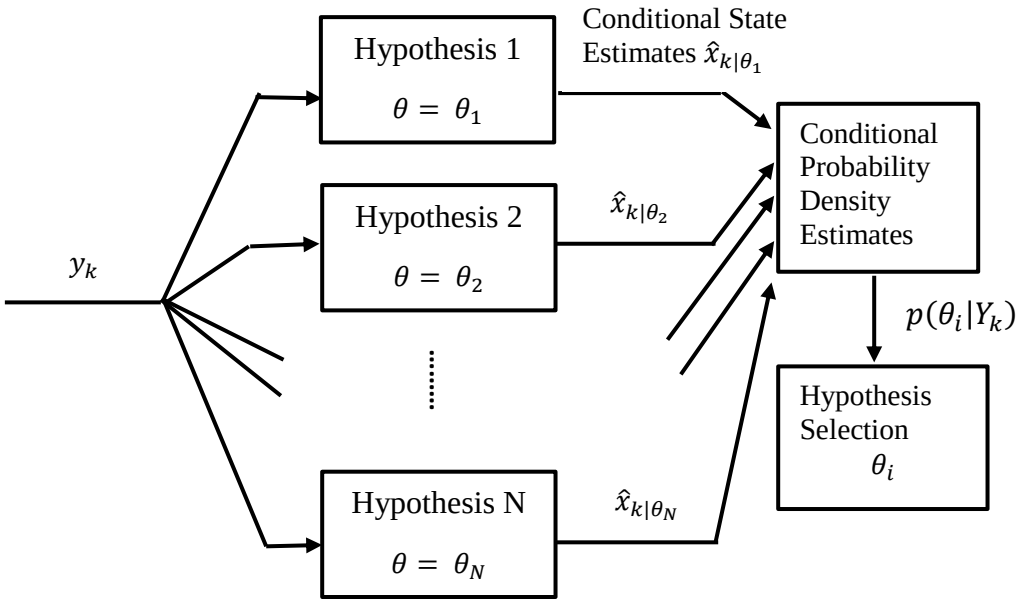

Figure 2.3: Flowchart of a bank of Kalman filters algorithm [19]

3 SYSTEM MODELING

In this chapter, a first-order and a second-order discrete time-invariant system will be used separately, and the performance of both systems will be shown. After knowing both systems' performance, the constant-type attack signal and the ramp-type attack signal will enter the systems, replacing the systems' output to affect the intrusion, so that the sensor cannot relay the true measurement signal. The performance of both the affected first-order and second-order system will be shown. The flow chart of the sensor intrusion process is shown in Fig 3.1

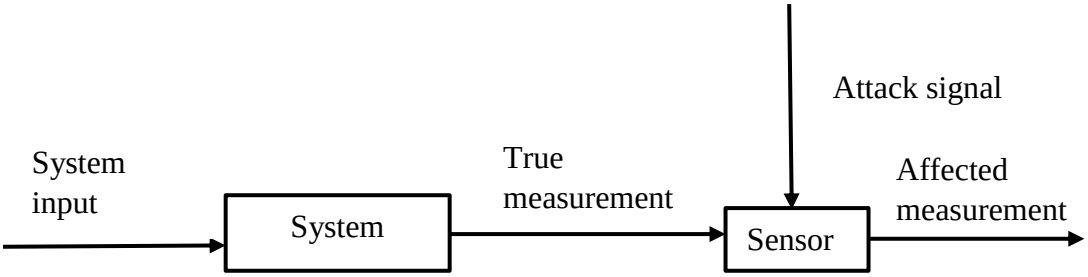

Figure 3.1: Flowchart of the sensor intrusion process

3.1 Model of the First-Order System

Consider a first-order discrete-time stochastic system with state and measurement noise

$$x_{k+1} = A_k x_k + B_k u_k + F_k v_k \qquad (3.1a)$$

$$y_k = C_k x_k + D_k u_k + G_k w_k \quad (3.1b)$$

Where $A_k = 0.9$, $B_k = 0$, $C_k = 1$, $D_k = 0$, $F_k = 1$, $G_k = 1$, and the covariance of the system state noise $V_k = 0.1$, the covariance of the system measurement noise $W_k = 0.05$ and both of the system state and measurement noises are zero-mean white and Gaussian. Therefore, the system can be represented as (3.2a) and (3.2b)

$$x_{k+1} = 0.9x_k + v_k \quad (3.2a)$$

$$y_k = x_k + w_k \quad (3.2b)$$

It can be noticed that this system is asymptotically stable from its system matrix A_k. Fig 3.2 and Fig 3.3 show the system state and system measurement responses with its initial state $x_0 = 2$ from $k = 0$ to 200.

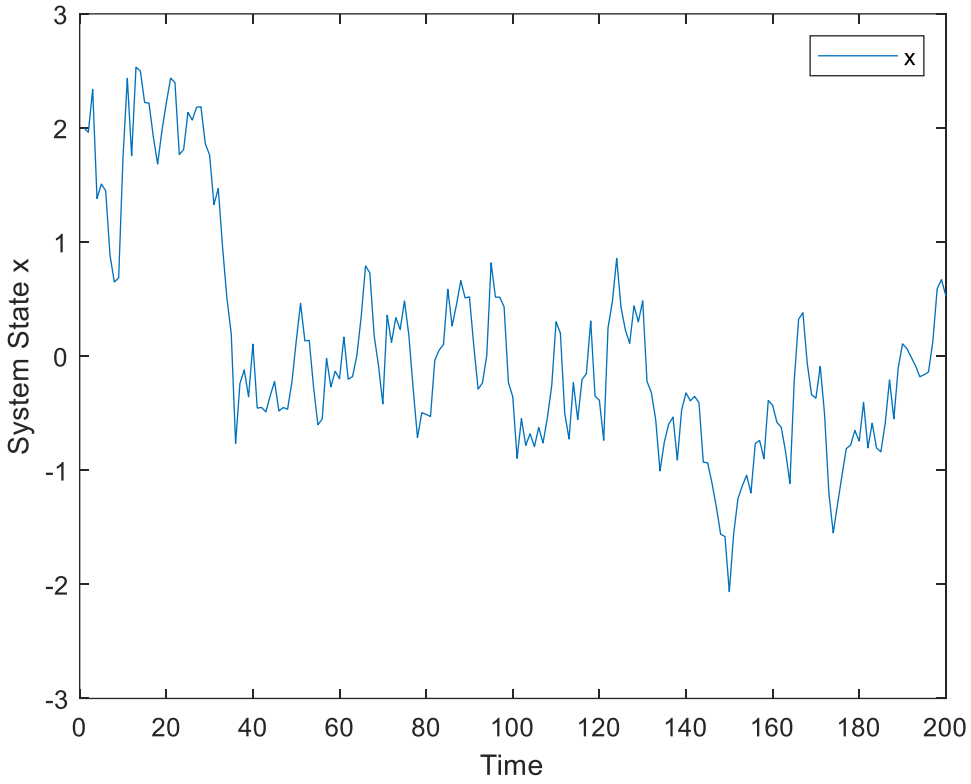

Figure 3.2: The First-Order Discrete-Time Stochastic system state response with its initial state $x_0 = 2$.

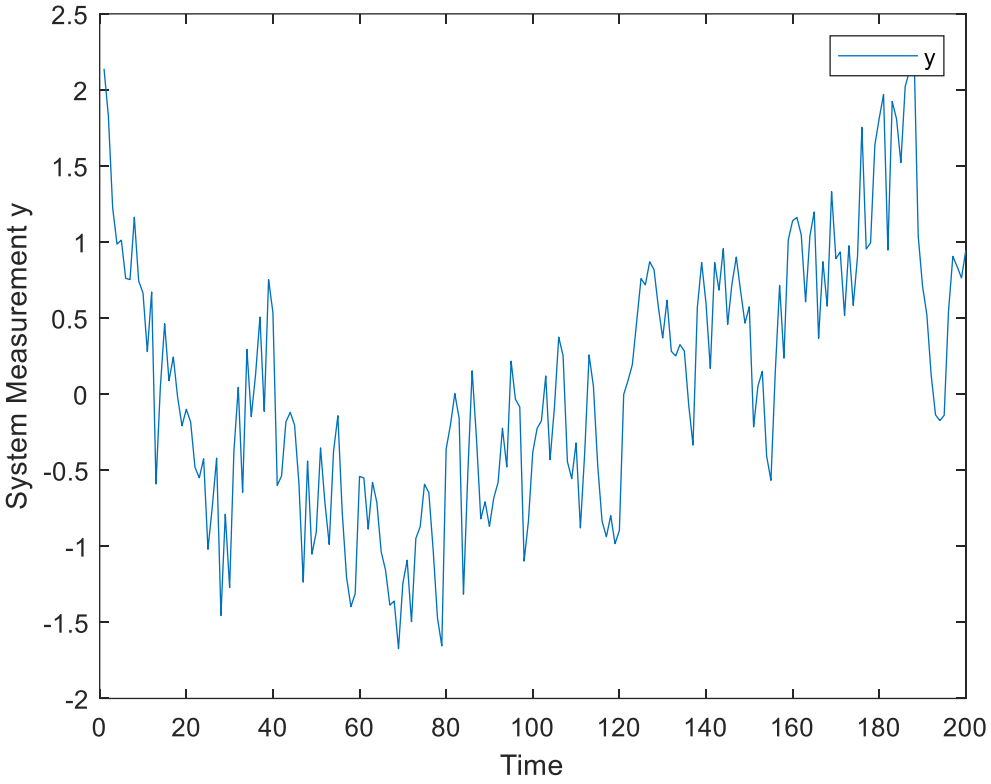

Figure 3.3: The First-Order Discrete-Time Stochastic system measurement response with its initial state $x_0 = 2$.

3.2 Attack Model for the First-Order System

3.2.1 Constant-Type Attack Signal

Consider the first-order discrete time-invariant system (3.2a) and (3.2b), where a hacker affects an intrusion, replacing most of the signal component of the system measurement by a constant signal h_k at a certain time. Then the model would be modified as below after hacking happens

$$\begin{bmatrix} x_{k+1} \\ h_{k+1} \end{bmatrix} = \begin{bmatrix} 0.9 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ h_k \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} v_k \qquad (3.3a)$$

$$y_k = \begin{bmatrix} 0.05 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ h_k \end{bmatrix} + w_k \qquad (3.3b)$$

Here, h_k is a time-invariant constant-type intrusion signal where $h_k = h_{k+1}$, whose model is added to the state equation. It can be found that the model is changed with its associated intrusion signal, where $A_k = \begin{bmatrix} 0.9 & 0 \\ 0 & 1 \end{bmatrix}$, $B_k = 0$, $C_k = [0.05 \quad 1]$, $D_k = 0$, $F_k = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $G_k = 1$. While, one can find that the system measurement is not completely replaced by the intrusion signal from $C_k = [0.05 \quad 1]$, there is still some "unhacked" measurements left, and this is because if the hacker replaces the whole measurement with a constant-type attack signal h_k, the model would be unobservable and the intrusion could be detected very easily by the failure of the Kalman filter used in estimating the state.

Suppose the intrusion happens at a certain time $k > 0$ when the system is running, and in this thesis, the time point k when intrusion happens is called "shiftpoint", which represents that the system is hacked at time k, and as for hackers, they could select any shiftpoint to attack the system. Here, three different shiftpoints $k = 50$, $k = 100$ and $k = 150$ are selected arbitrarily to show the changes of model state and measurement when there is a constant-type attack signal $h_k = 10$ enters the system

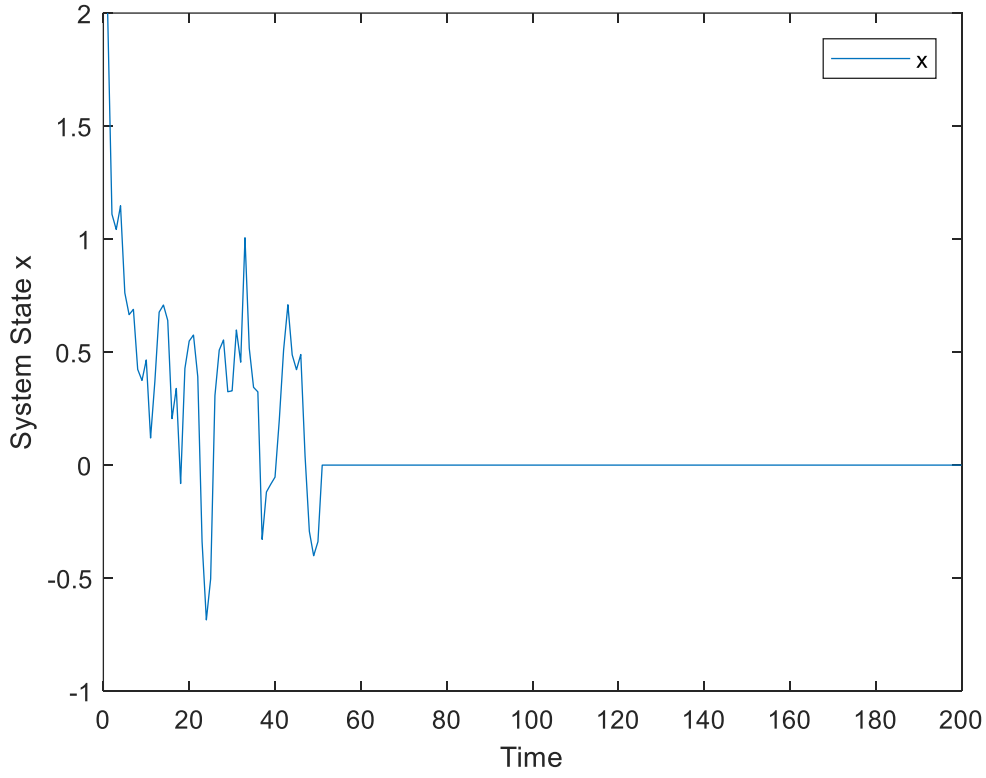

Figure 3.4: The First-Order Discrete-Time stochastic system state response with its initial state $x_0 = 2$ when the Constant-Type sensor intrusion happens at shiftpoint $k = 50$.

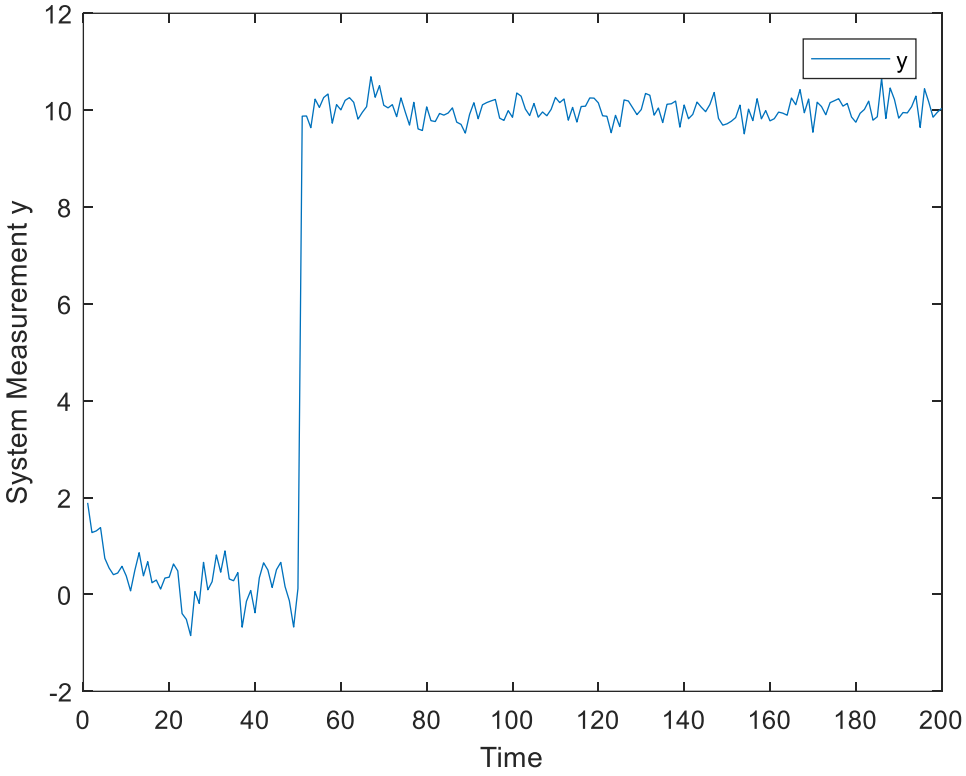

Figure 3.5: The First-Order Discrete-Time stochastic system measurement state response with its initial state $x_0 = 2$ when the Constant-Type sensor intrusion happens at shiftpoint $k = 50$.

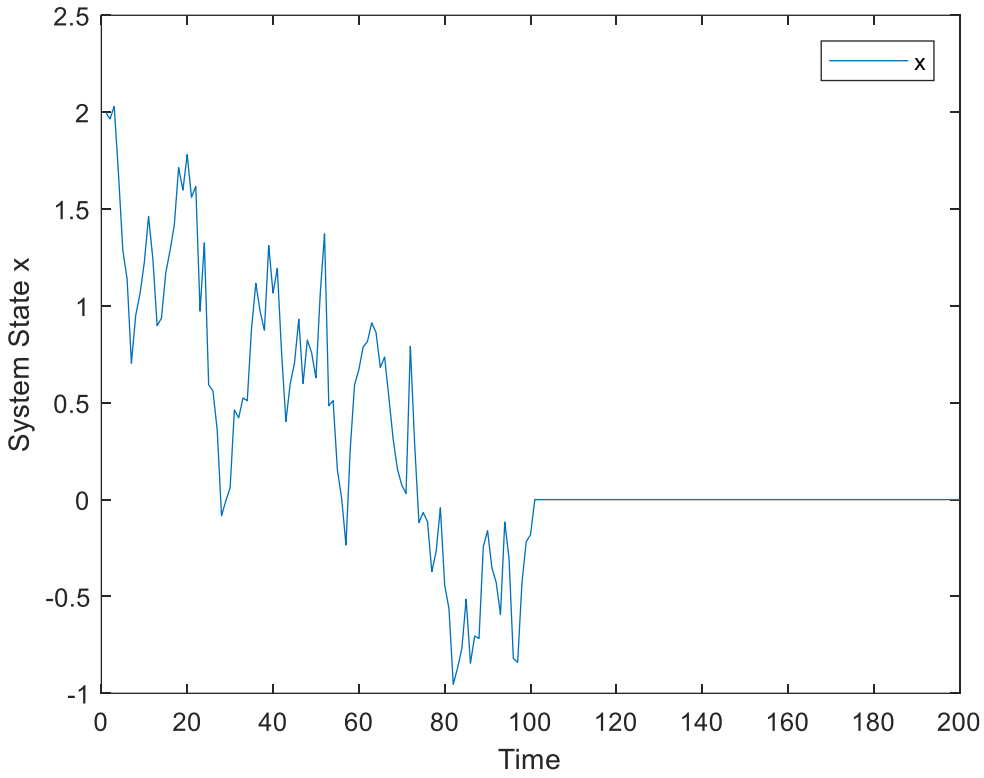

Figure 3.6: The First-Order Discrete-Time stochastic system state response with its initial state $x_0 = 2$ when the Constant-Type sensor intrusion happens at shiftpoint $k = 100$.

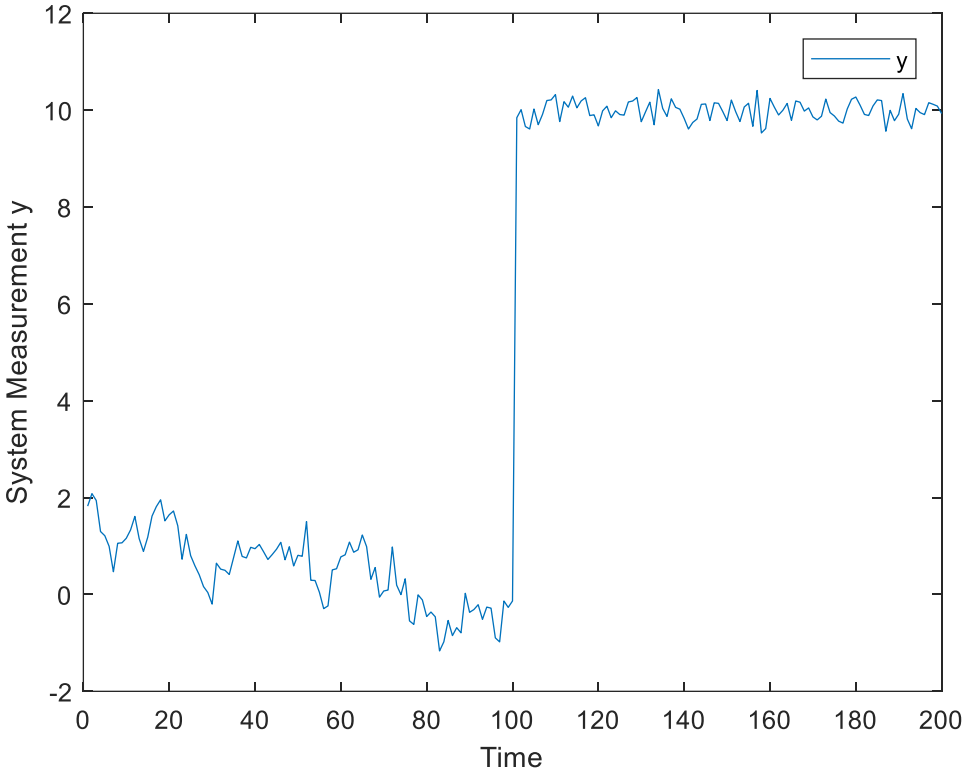

Figure 3.7: The First-Order Discrete-Time stochastic system measurement state response with its initial state $x_0 = 2$ when the Constant-Type sensor intrusion happens at shiftpoint $k = 100$.

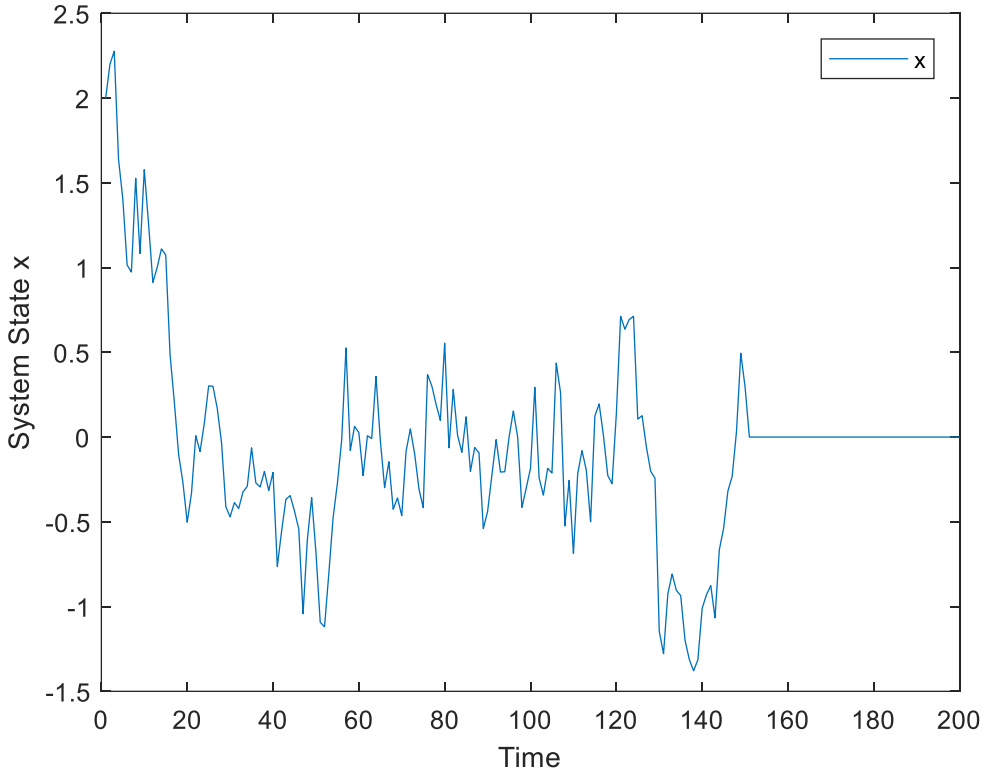

Figure 3.8: The First-Order Discrete-Time stochastic system state response with its initial state $x_0 = 2$ when the Constant-Type sensor intrusion happens at shiftpoint $k = 150$.

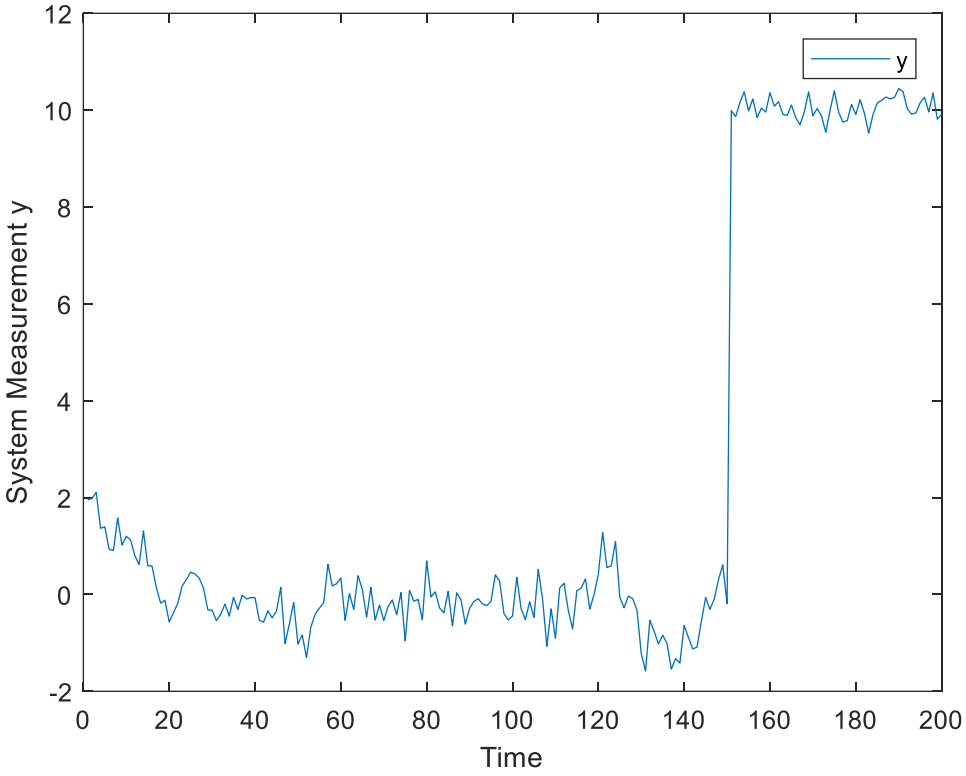

Figure 3.9: The First-Order Discrete-Time stochastic system measurement state response with its initial state $x_0 = 2$ when the Constant-Type sensor intrusion happens at shiftpoint $k = 150$.

3.2.2 Step and ramp-type Attack Signal

Consider the first-order discrete time-invariant system (3.2a) and (3.2b), where a hacker enters the system, replaces most of the signal component of the system measurement by a step and step and ramp-type signal h_k at a certain time. Then the system model would be modified as below after hacking happens

$$\begin{bmatrix} x_{k+1} \\ h_{k+1}^1 \\ h_{k+1}^2 \end{bmatrix} = \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ h_k^1 \\ h_k^2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} v_k \tag{3.4a}$$

$$y_k = [0.05 \quad 1 \quad 1] \begin{bmatrix} x_k \\ h_k^1 \\ h_k^2 \end{bmatrix} + w_k \tag{3.4b}$$

Here, h_k^1 is a step and ramp-type and h_k^2 is a step-type hacking signal with the state space models

$$h_{k+1} = \begin{bmatrix} h_{k+1}^1 \\ h_{k+1}^2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} h_k^1 \\ h_k^2 \end{bmatrix} \tag{3.5}$$

And this step and ramp-type signal could be shown as Fig 3.10 with its initial value $h_0 = \begin{bmatrix} 1 \\ 0.1 \end{bmatrix}$

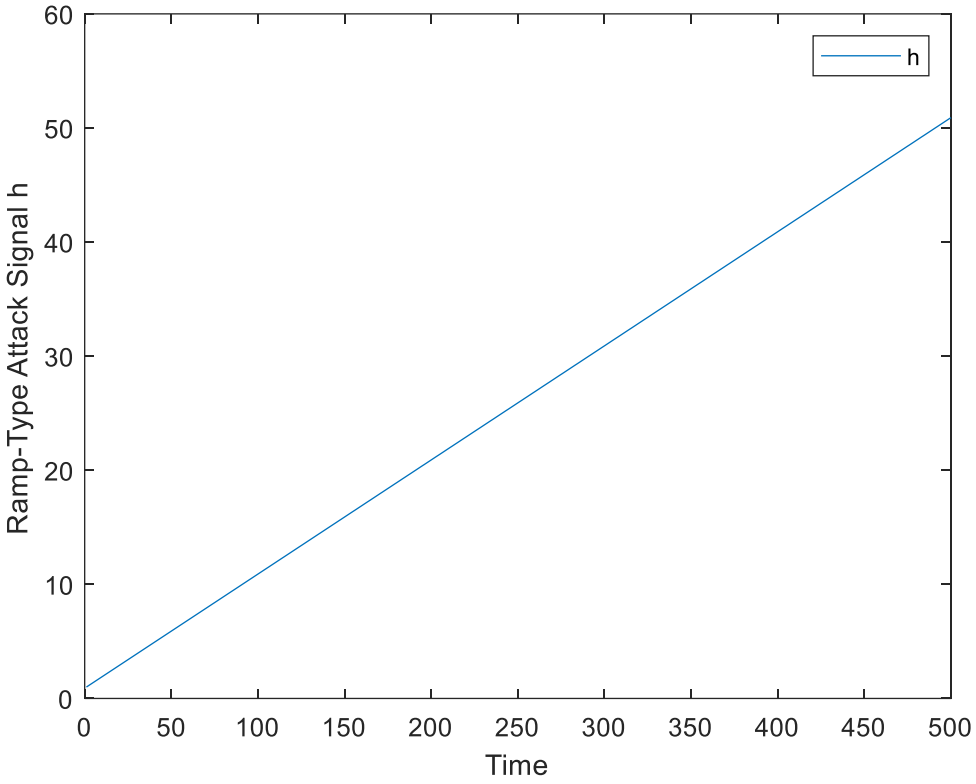

Figure 3.10: Step and Step and ramp-type intrusion signal with its initial response

$$\boldsymbol{h_0 = \begin{bmatrix} 1 \\ 0.1 \end{bmatrix}.}$$

also, it can be found that the whole system is changed with its associated intrusion signal,

where $A_k = \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$, $B_k = 0$, $C_k = [0.05 \quad 1 \quad 1]$, $D_k = 0$, $F_k = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ and $G_k = 1$.

As mentioned above, the system measurement could not be replaced completely by the intrusion signal from $C_k = [0.05 \quad 1 \quad 1]$ because of the unobservability of the system. There is still some "unhacked" measurement needs to be left to make sure the modification of the system measurement cannot be detected easily.

Here, three different shiftpoints $k = 50$, $k = 100$ and $k = 150$ are selected arbitrarily to show the changes of system state and system measurement when there is a step and ramp-type attack signal (3.5) enters the system

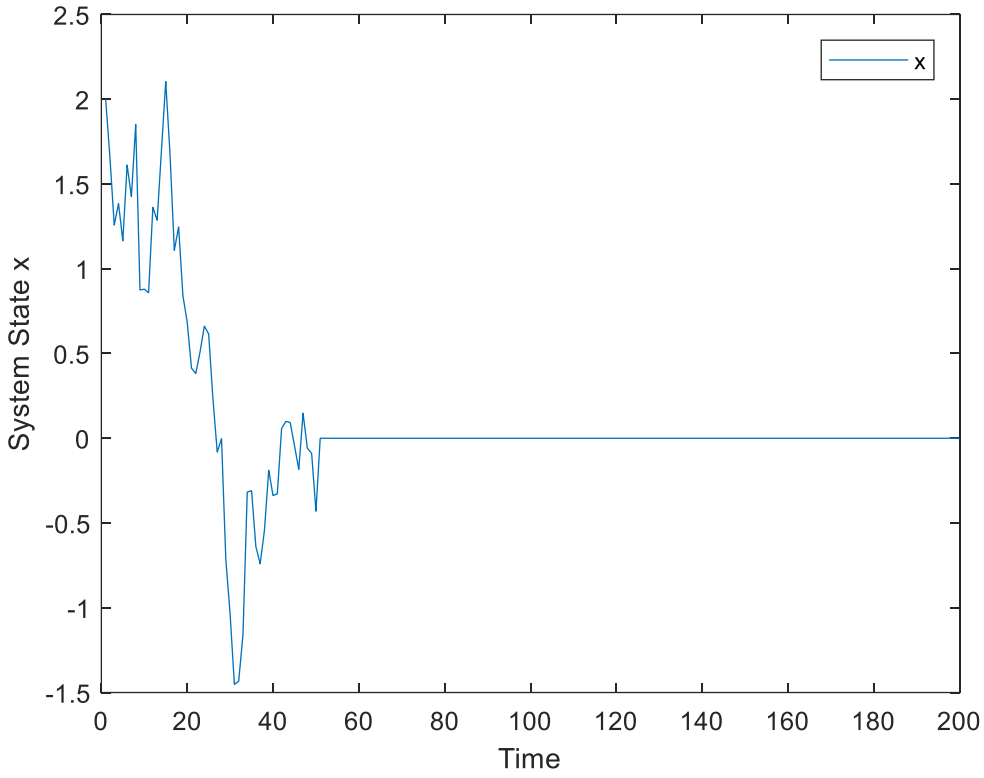

Figure 3.11: The First-Order Discrete-Time stochastic system state response with its initial state $x_0 = 2$ when the Step and ramp-type sensor intrusion happens at shiftpoint $k = 150$.

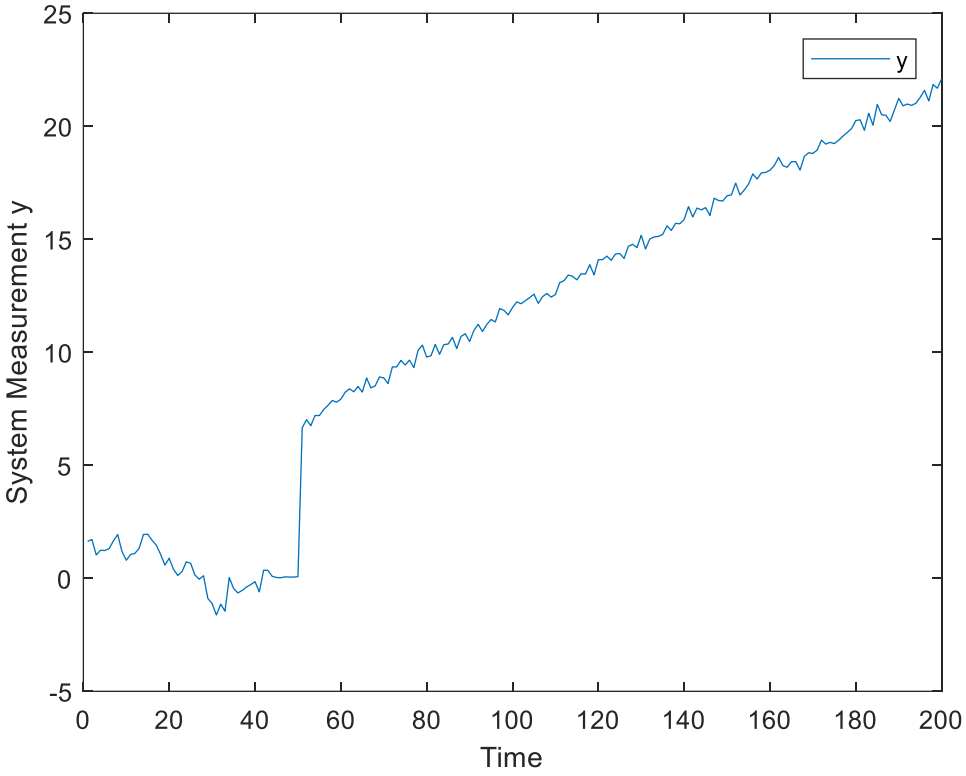

Figure 3.12: The First-Order Discrete-Time stochastic system measurement state response with its initial state $x_0 = 2$ when the Step and Ramp-Type sensor intrusion happens at shiftpoint $k = 150$.

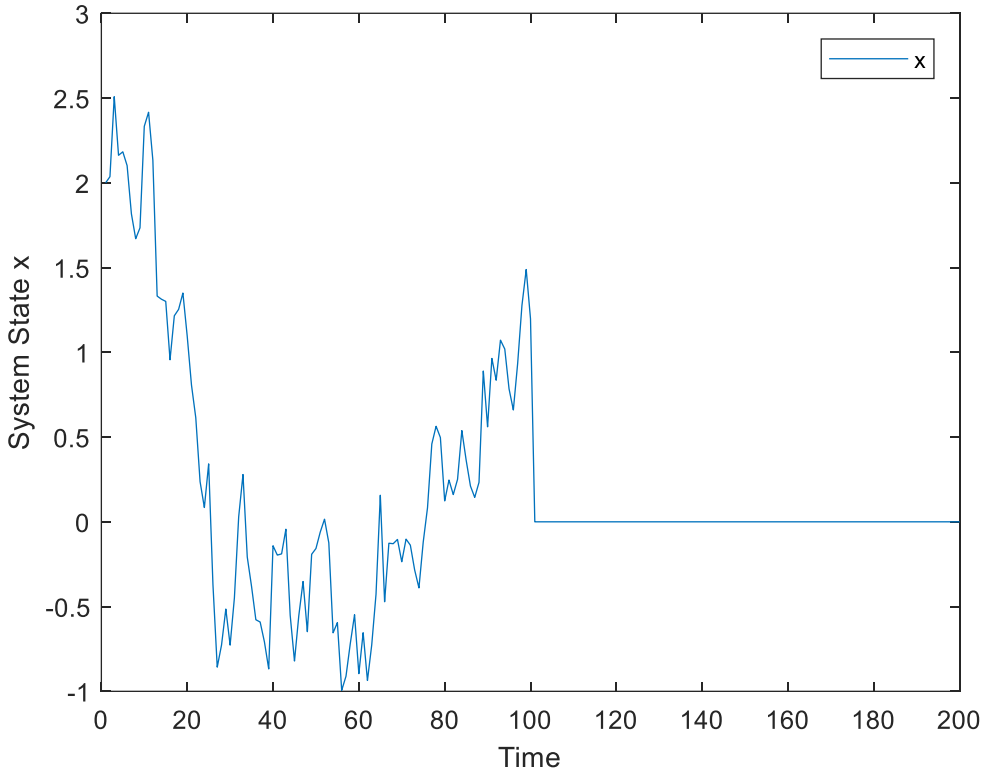

Figure 3.13: The First-Order Discrete-Time stochastic system state response with its initial state $x_0 = 2$ when the Step and Ramp-Type sensor intrusion happens at shiftpoint $k = 150$.

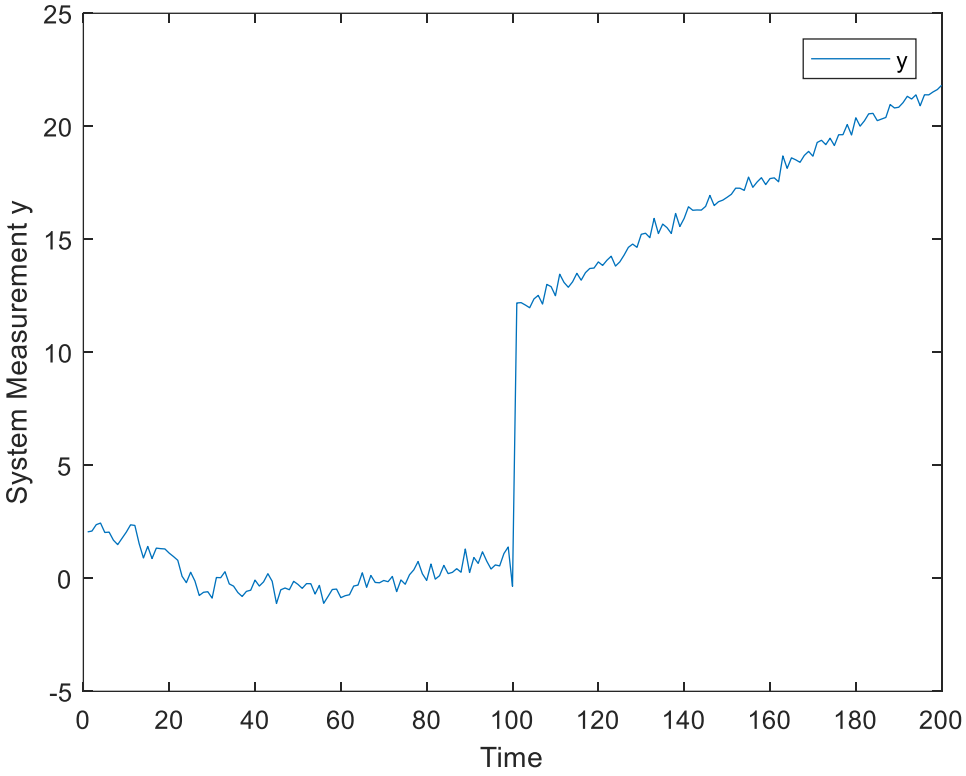

Figure 3.14: The First-Order Discrete-Time stochastic system measurement state response with its initial state $x_0 = 2$ when the Step and Ramp-Type sensor intrusion happens at shiftpoint $k = 150$.

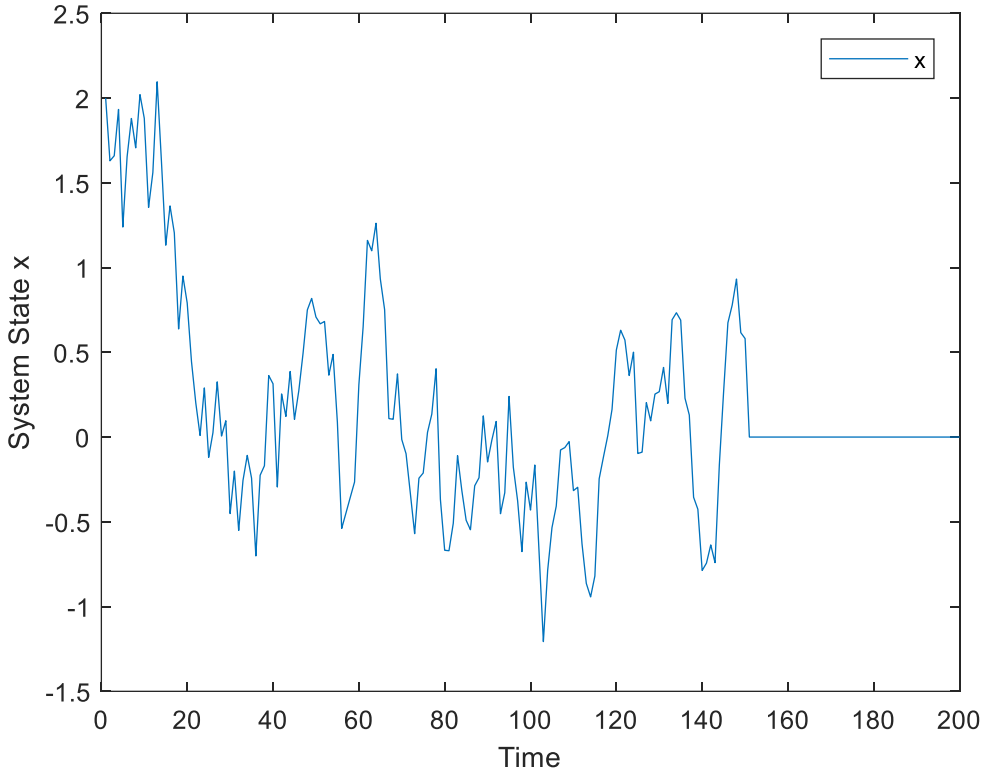

Figure 3.15: The First-Order Discrete-Time stochastic system state response with its initial state $x_0 = 2$ when the Step and Ramp-Type sensor intrusion happens at shiftpoint $k = 150$.

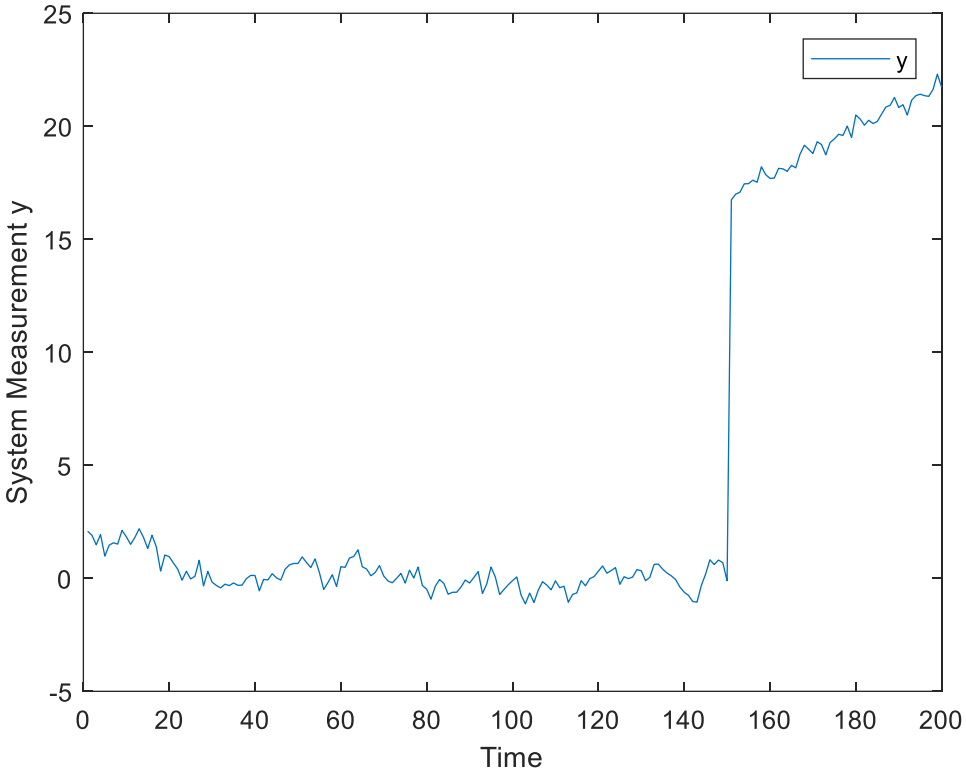

Figure 3.16: The First-Order Discrete-Time stochastic system measurement state response with its initial state $x_0 = 2$ when the Step and Ramp-Type sensor intrusion happens at shiftpoint $k = 150$.

3.3 The Second-Order System Model

Consider a second-order discrete-time stochastic system with state and measurement noise

$$x_{k+1} = A_k x_k + B_k u_k + F_k v_k \tag{3.6a}$$

$$y_k = C_k x_k + D_k u_k + G_k w_k \tag{3.6b}$$

where $A_k = \begin{bmatrix} 0 & 0.9 \\ -1 & -1 \end{bmatrix}$, $B_k = 0$, $C_k = [1 \quad 1]$, $D_k = 0$, $F_k = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $G_k = 1$, and the covariance of the system state noise $v_k = 1$, the covariance of the system measurement noise $w_k = 1$ and both of the system state and measurement noises are zero mean, white and Gaussian. Therefore, the system can be represented as (3.7a) and (3.7b)

$$x_{k+1} = \begin{bmatrix} 0 & 0.9 \\ -1 & -1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 0 \end{bmatrix} v_k \tag{3.7a}$$

$$y_k = [1 \quad 1] x_k + w_k \tag{3.7b}$$

It can be noticed that this system is also an asymptotically stable system where the system's eigenvalues are -0.5 ± 0.8062i, which are inside the unit circle. Fig 3.10a, Fig 3.10b and Fig 3.11 show the system state value and system measurement response with the initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ from $k = 0$ to 200.

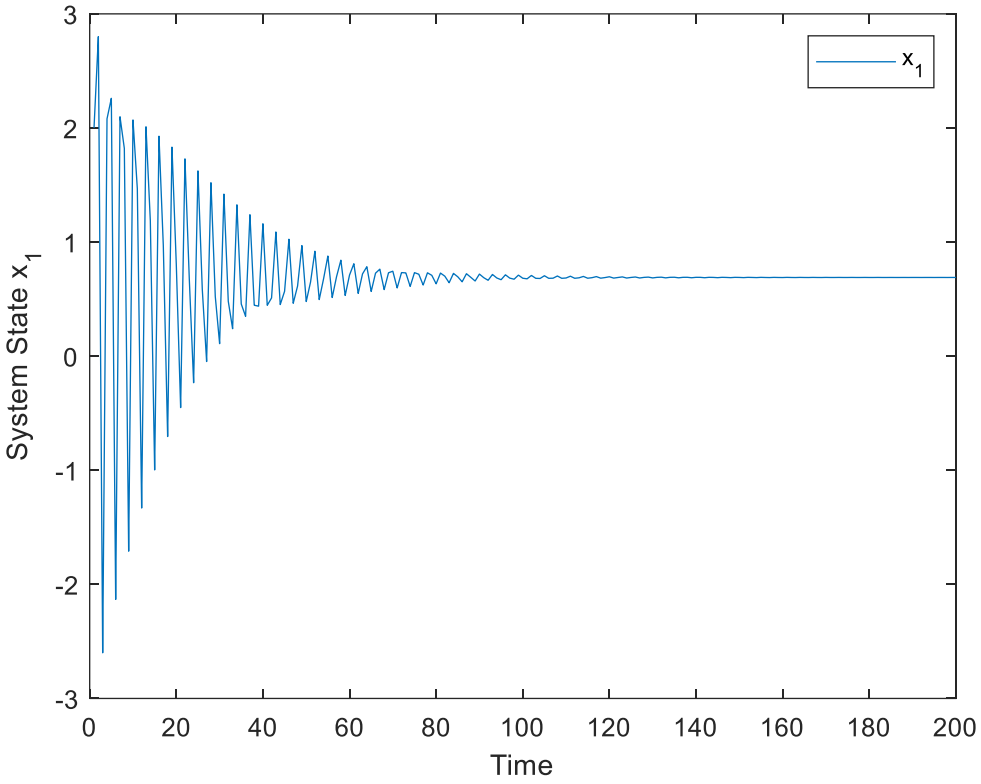

Figure 3.17: The Second-Order Discrete-Time Stochastic system state $x_1$response with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$.

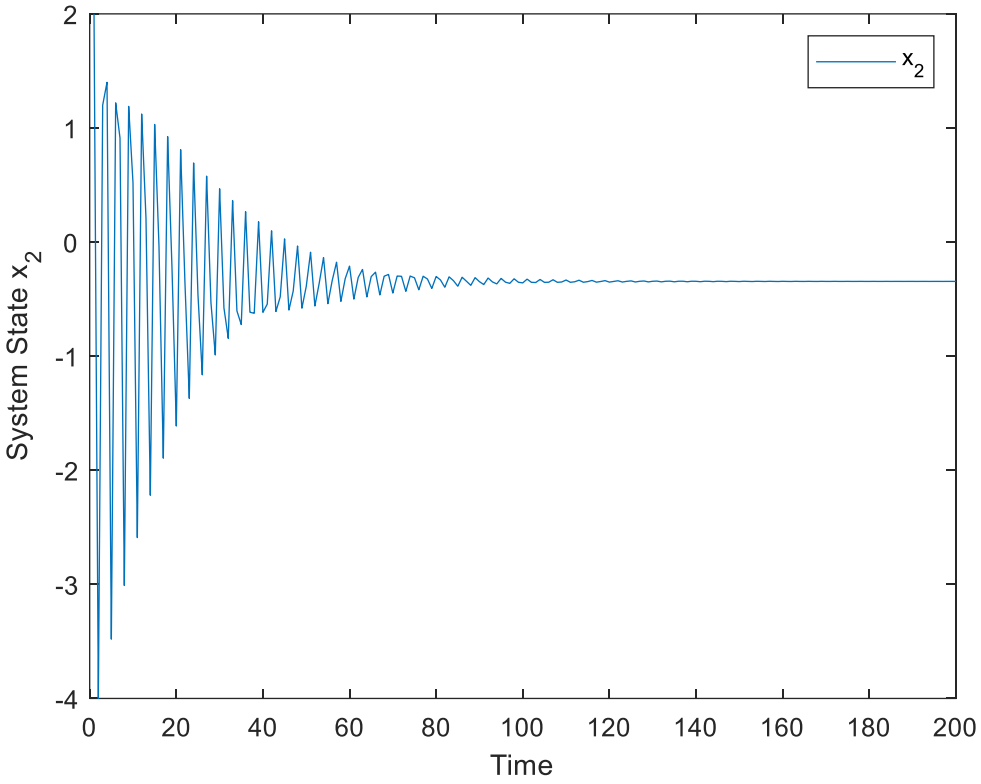

Figure 3.18: The Second-Order Discrete-Time Stochastic system state x_2 response with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$.

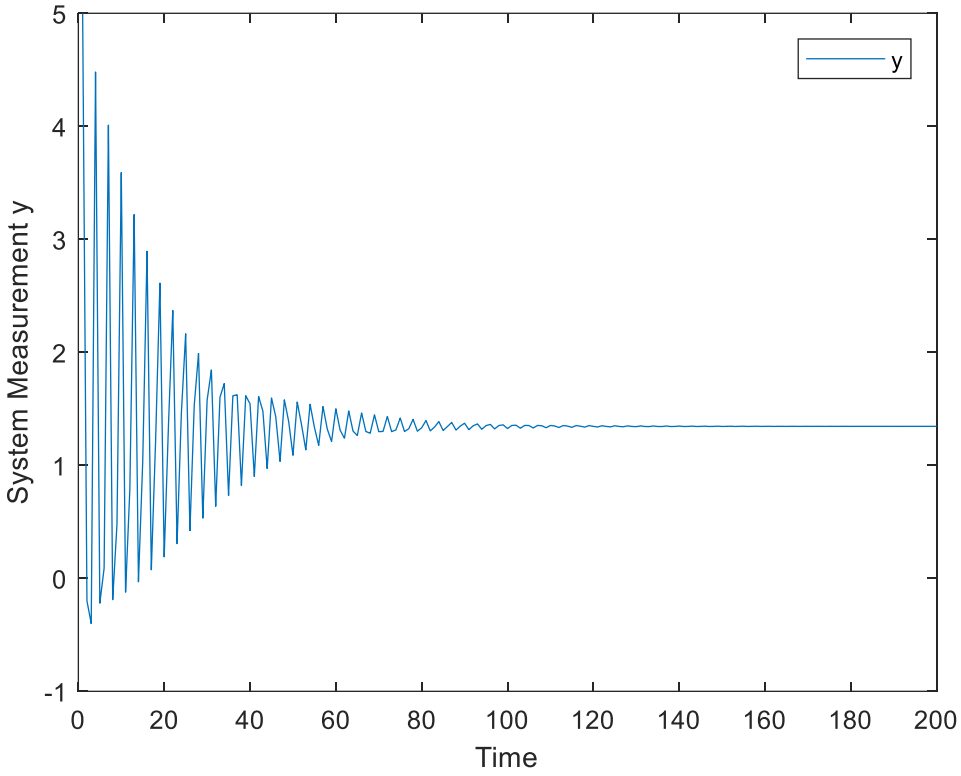

Figure 3.19: The Second-Order Discrete-Time Stochastic system measurement response with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$.

3.4 Attack Model for the Second-Order System

3.4.1 Constant-Type Attack Signal

Consider the second-order discrete time-invariant system (3.7a) and (3.7b), where a hacker enters the system, replace most of the signal component of the system measurement to a constant signal h_k at a certain shiftpoint. Then the system model would be modified as below after the intrusion happens

$$\begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \\ h_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 0.9 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \\ h_k \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} v_k \tag{3.8a}$$

$$y_k = [0 \quad 0.1 \quad 1] x_k + w_k \tag{3.8b}$$

Here, h_k is a time-invariant constant-type intrusion signal where $h_k = h_{k+1}$, and it can be found that the whole system is changed with its associated intrusion signal, where $A_k = \begin{bmatrix} 0 & 0.9 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $B_k = 0$, $C_k = [0 \quad 0.1 \quad 1]$, $D_k = 0$, $F_k = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ and $G_k = 1$. While, it can also be noticed that the system measurement is not replaced completely by the intrusion signal from $C_k = [0 \quad 0.1 \quad 1]$, and that is because if the hacker wants to replace the whole states into the intrusion signal, then $C_k = [0 \quad 0 \quad 1]$ and in this case, the system would be unobservable. Thus, as mentioned, hackers cannot replace the whole states completely and they need to leave some "unhacked" measurement to keep the system observable so that the intrusion could not be found easily.

Here, three different shiftpoints $k = 100$, $k = 250$ and $k = 400$ are selected arbitrarily to show the changes of system state and system measurement when there is a constant-type attack signal $h_k = 10$ enters the system

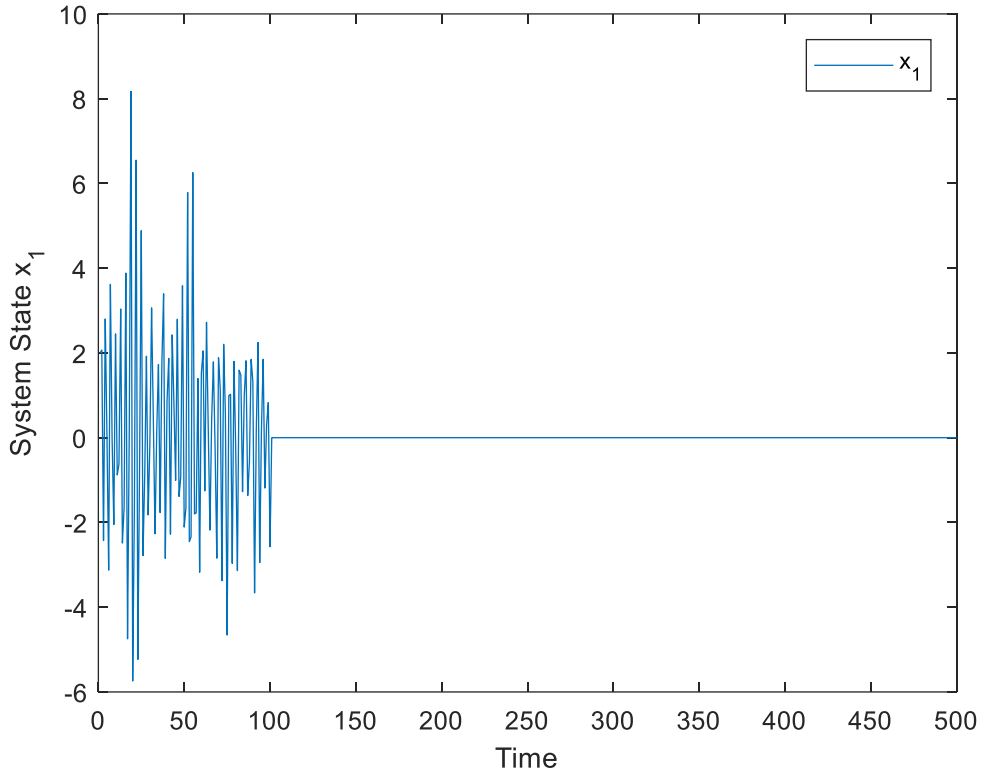

Figure 3.20: The Second-Order Discrete-Time Stochastic system state response x_1 with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Constant-Type sensor intrusion happens at shiftpoint $k = 100$.

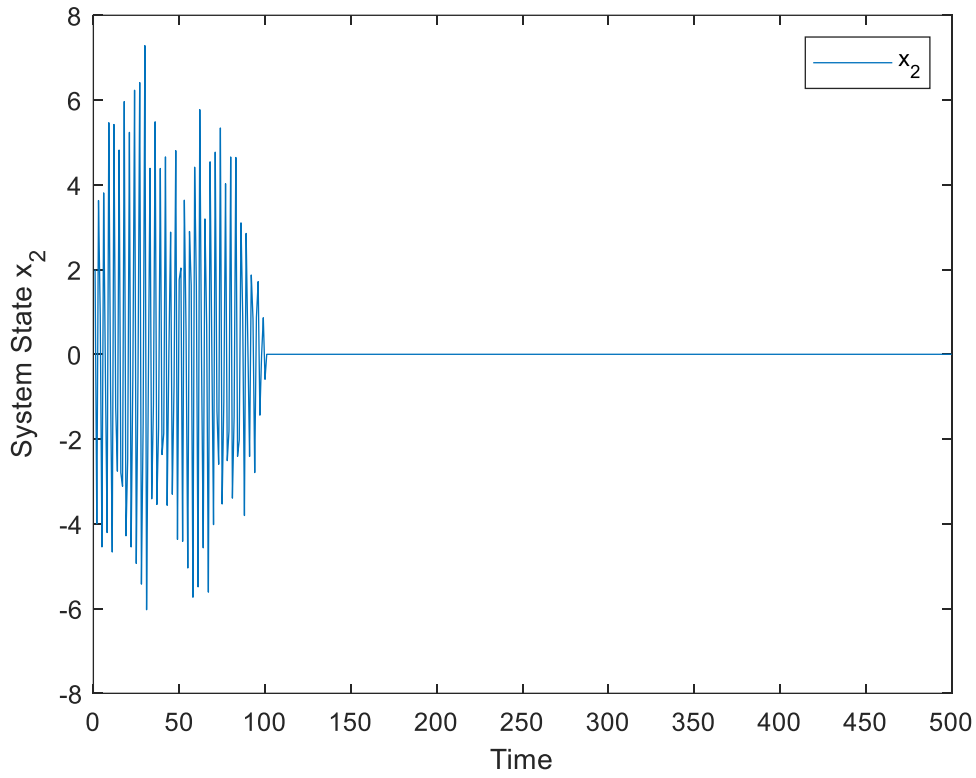

Figure 3.21: The Second-Order Discrete-Time Stochastic system state response x_2 with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Constant-Type sensor intrusion happens at shiftpoint $k = 100$.

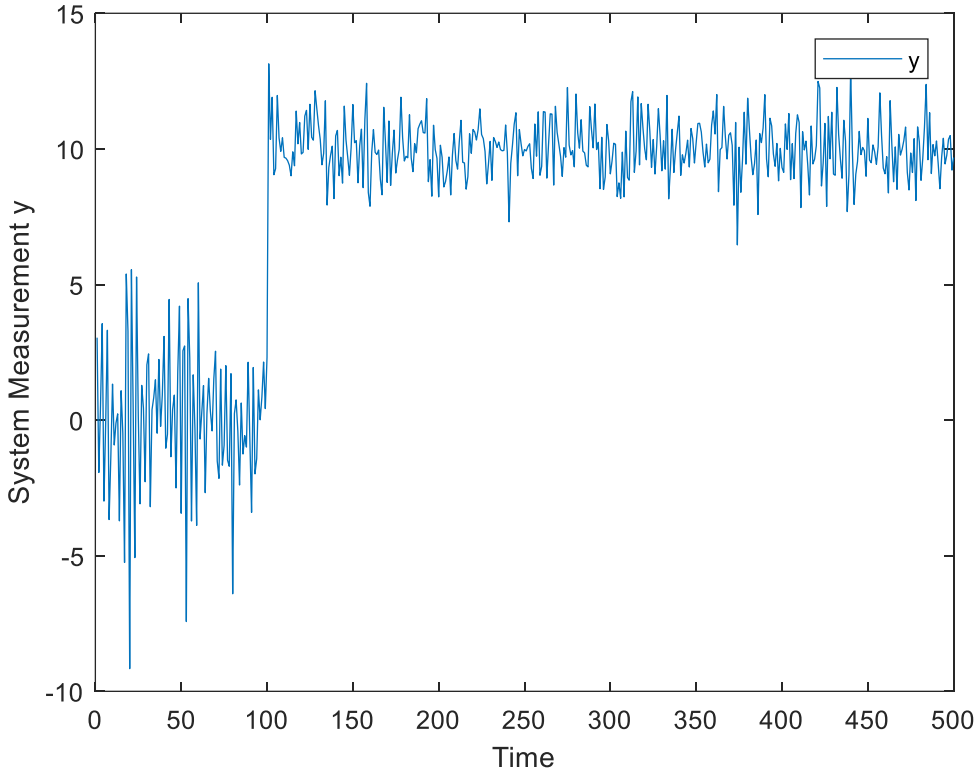

Figure 3.22: The Second-Order Discrete-Time Stochastic system measurement response with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Constant-Type sensor intrusion happens at shiftpoint $k = 100$.

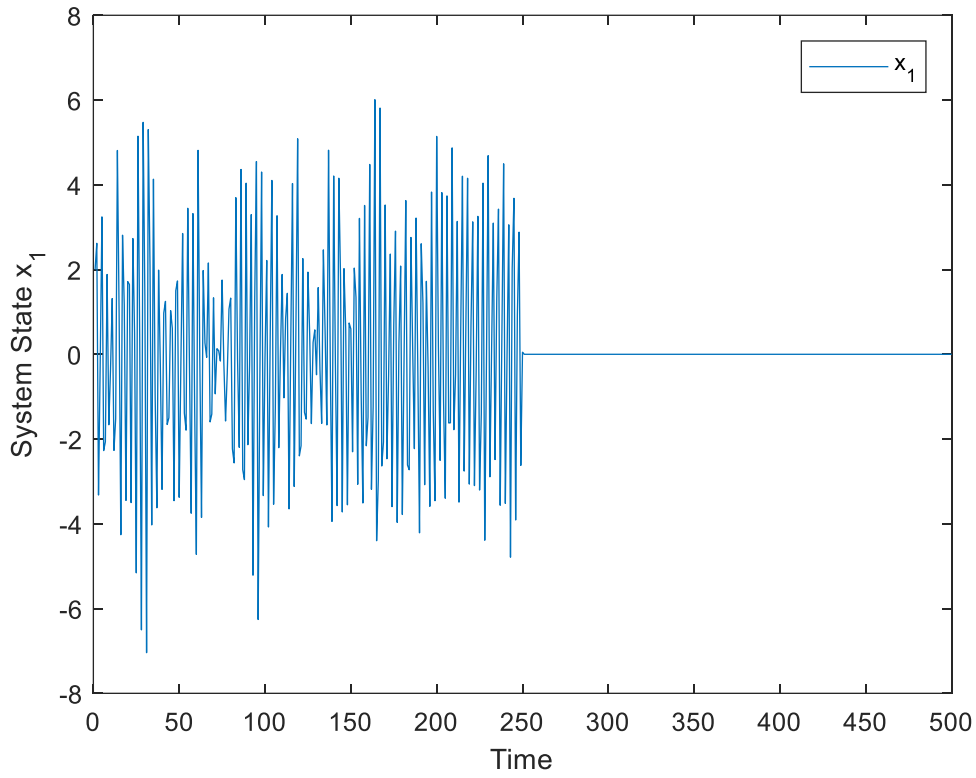

Figure 3.23: The Second-Order Discrete-Time Stochastic system state response x_1 with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Constant-Type sensor intrusion happens at shiftpoint $k = 250$.

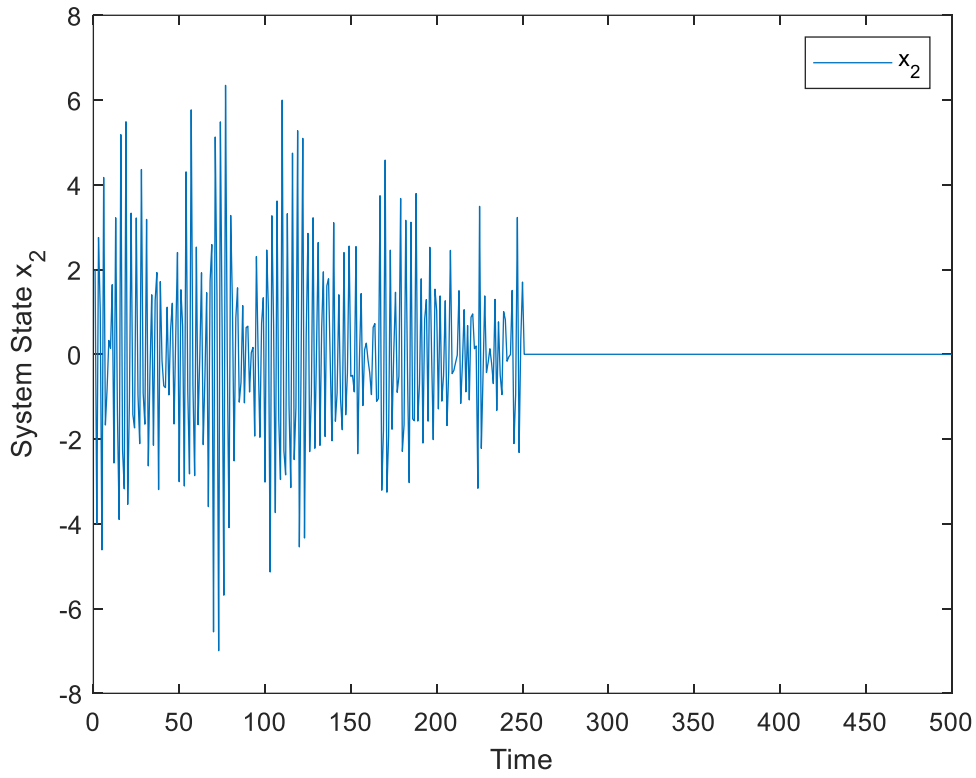

Figure 3.24: The Second-Order Discrete-Time Stochastic system state response x_2 with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Constant-Type sensor intrusion happens at shiftpoint $k = 250$.

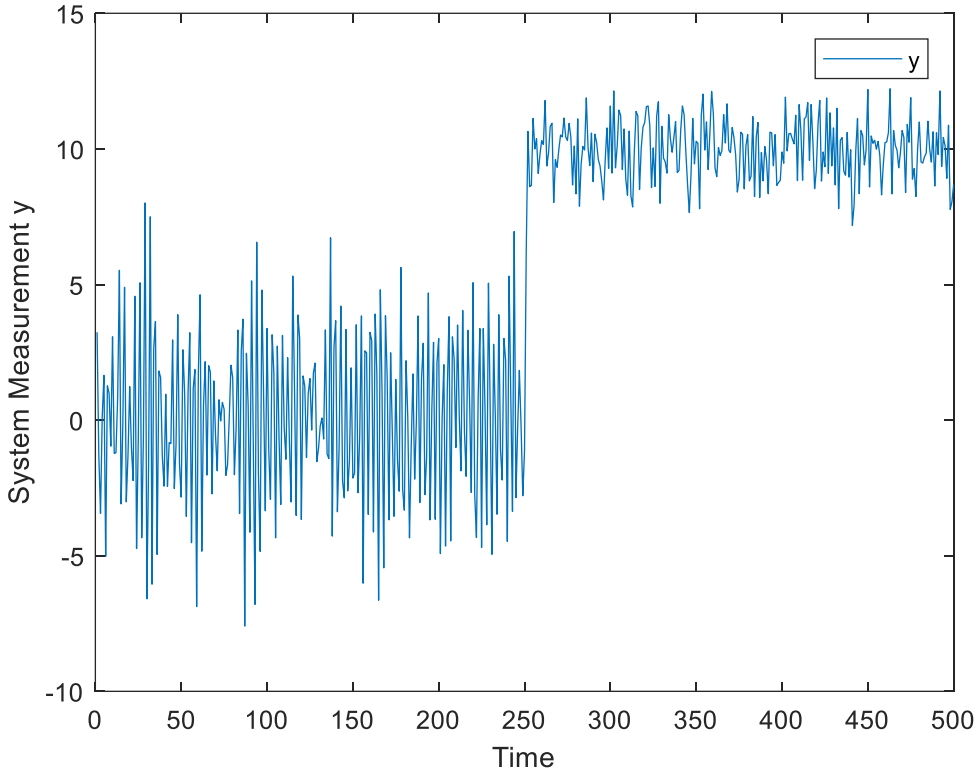

Figure 3.25: The Second-Order Discrete-Time Stochastic system measurement response with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Constant-Type sensor intrusion happens at shiftpoint $k = 250$.

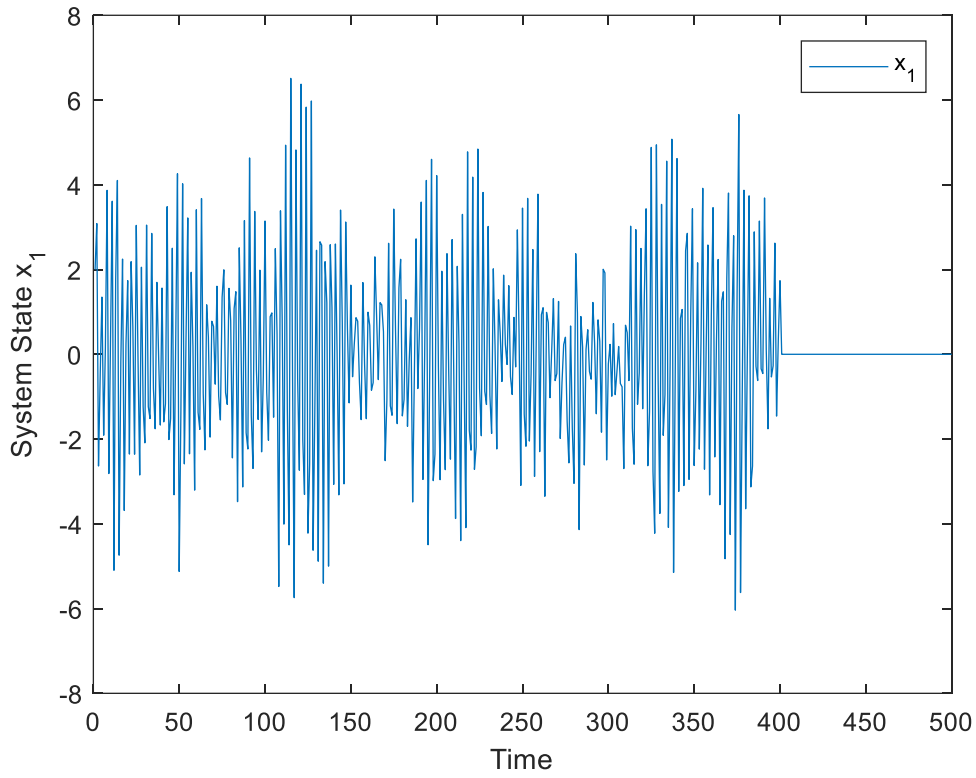

Figure 3.26: The Second-Order Discrete-Time Stochastic system state response x_1 with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Constant-Type sensor intrusion happens at shiftpoint $k = 400$.

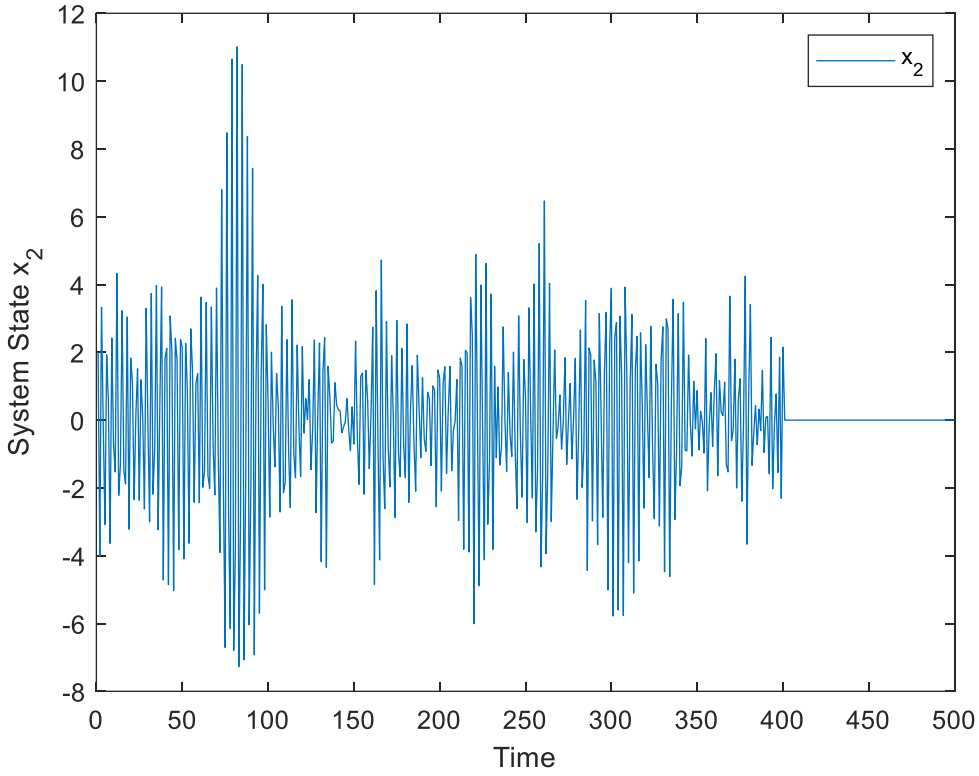

Figure 3.27: The Second-Order Discrete-Time Stochastic system state response x_2 with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Constant-Type sensor intrusion happens at shiftpoint $k = 400$.

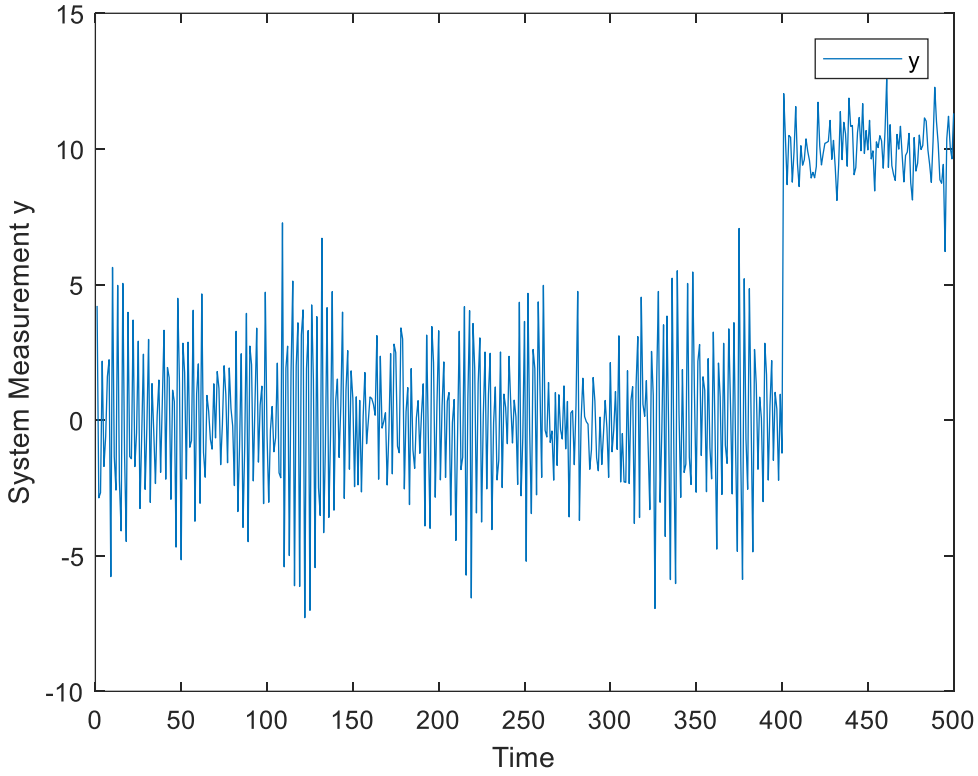

Figure 3.28: The Second-Order Discrete-Time Stochastic system measurement response with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Constant-Type sensor intrusion happens at shiftpoint $k = 400$.

3.4.2 Ramp-Type Attack Signal

Consider the second-order discrete time-invariant system (3.7a) and (3.7b), where a hacker enters the system, replace most of the signal component of the system measurement to a setp and ramp-type signal h_k at a certain shiftpoint. Then the system model would be modified as below after the intrusion happens

$$\begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \\ h_{k+1}^1 \\ h_{k+1}^2 \end{bmatrix} = \begin{bmatrix} 0 & 0.9 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \\ h_k^1 \\ h_k^2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} v_k \tag{3.9a}$$

$$y_k = [0 \quad 0.1 \quad 1 \quad 0]x_k + w_k \tag{3.9b}$$

Here, h_k is a step and ramp-type intrusion signal where it is the same signal as mentioned in (3.5) and because of the intrusion signal is added into the system, the system matrices are changed, where $A_k = \begin{bmatrix} 0 & 0.9 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $B_k = 0$, $C_k = [0 \quad 0.1 \quad 1 \quad 0]$, $D_k = 0$, $F_k = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ and $G_k = 1$. While, as mentioned before, it can be noticed that the system measurement is not replaced completely by the intrusion signal from $C_k = [0 \quad 0.1 \quad 1 \quad 0]$, if the hacker wants to replace both states into the intrusion signal, then $C_k = [0 \quad 0 \quad 1 \quad 0]$ and in this case, the system would be unobservable. Thus, in order

to make sure the intrusion could not be found easily, hackers still need to leave some "unhacked" measurement to keep the system be observable.

Here, three different shiftpoints $k = 100$, $k = 250$ and $k = 400$ are selected arbitrarily to show the changes of system state and system measurement when there is a step and ramp-type intrusion signal h_k enters the system

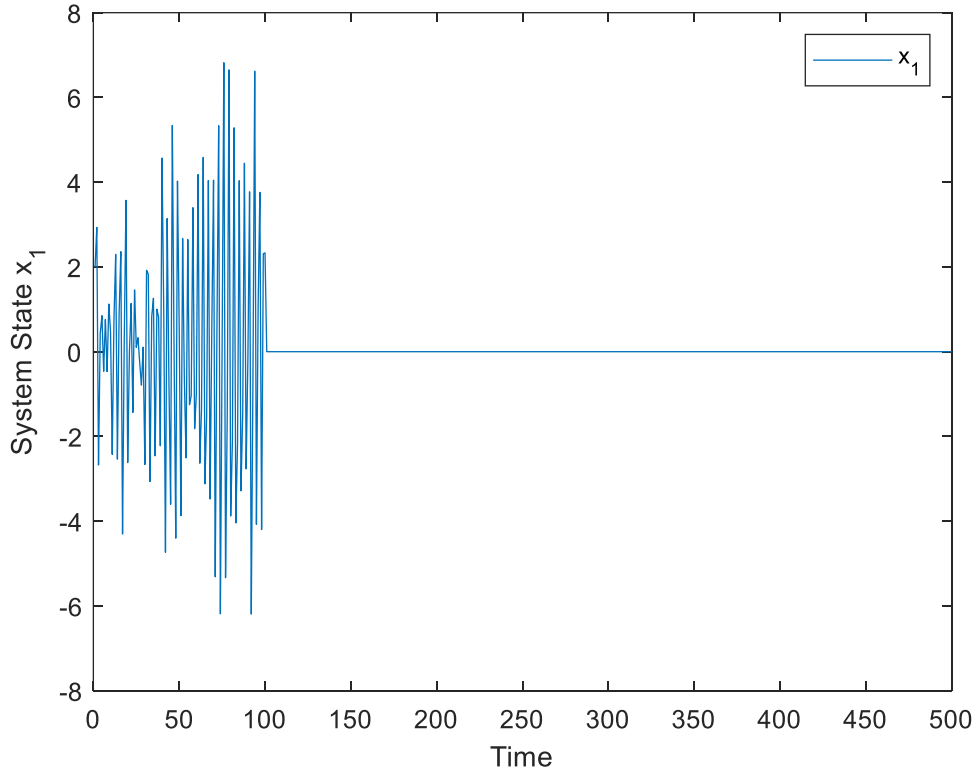

Figure 3.29: The Second-Order Discrete-Time Stochastic system state response x_1 with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Step and Ramp-Type sensor intrusion happens at shiftpoint $k = 100$.

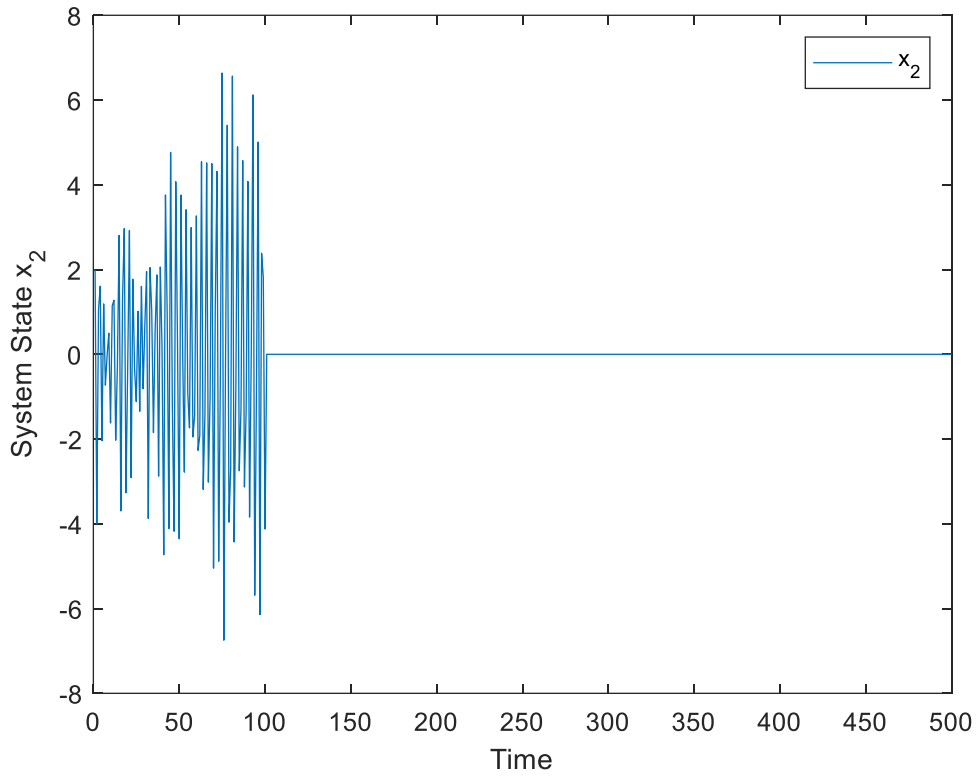

Figure 3.30: The Second-Order Discrete-Time Stochastic system state response x_2 with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Step and Ramp-Type sensor intrusion happens at shiftpoint $k = 100$.

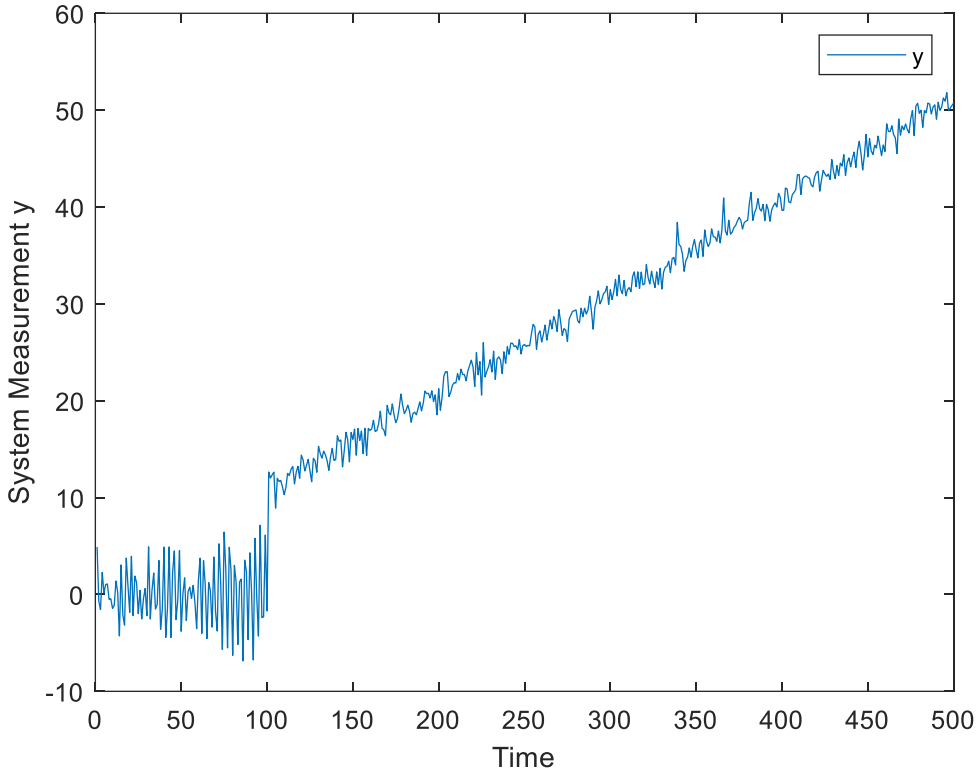

Figure 3.31: The Second-Order Discrete-Time Stochastic system measurement response with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Step and Ramp-Type sensor intrusion happens at shiftpoint $k = 100$.

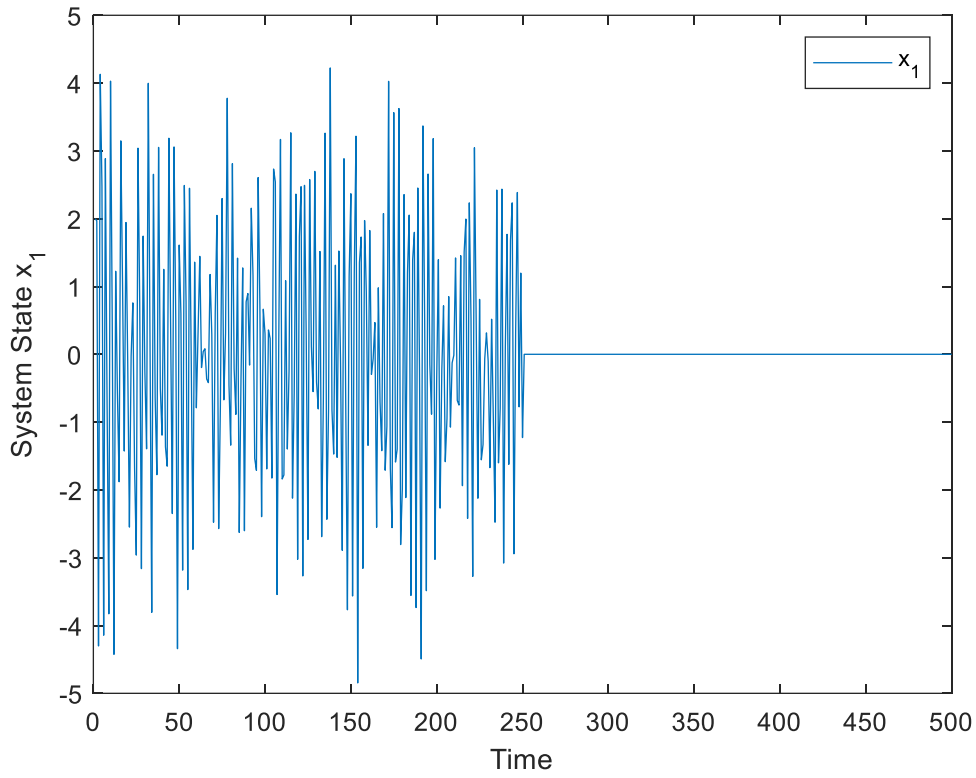

Figure 3.32: The Second-Order Discrete-Time Stochastic system state response x_1 with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Step and Ramp-Type sensor intrusion happens at shiftpoint $k = 250$.

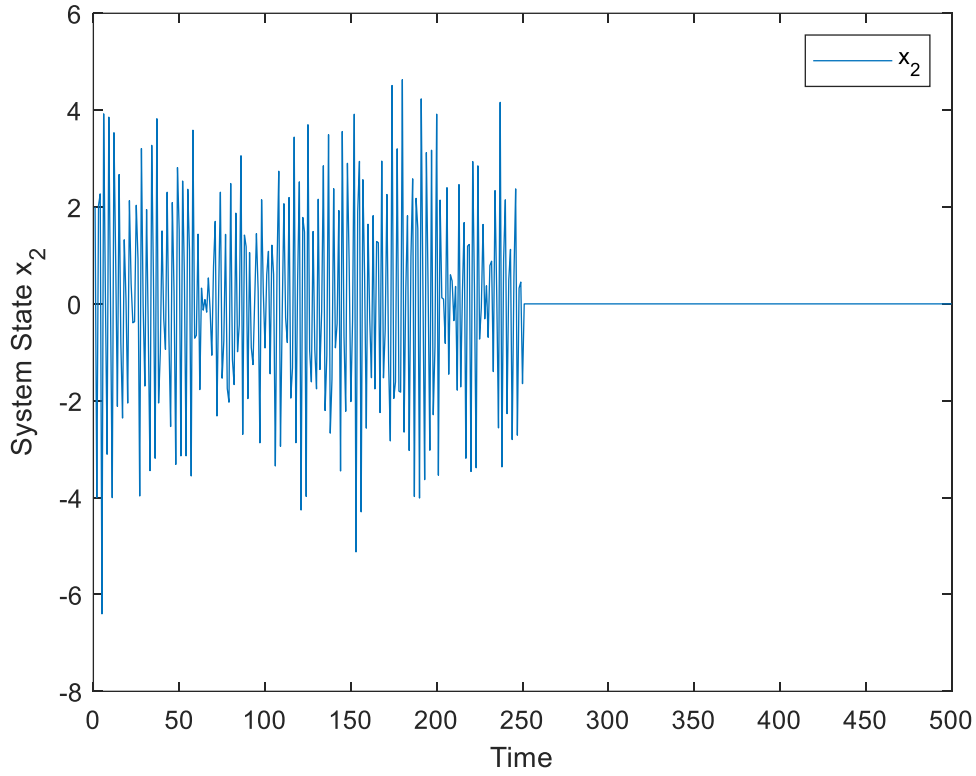

Figure 3.33: The Second-Order Discrete-Time Stochastic system state response x_2 with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Step and Ramp-Type sensor intrusion happens at shiftpoint $k = 250$.

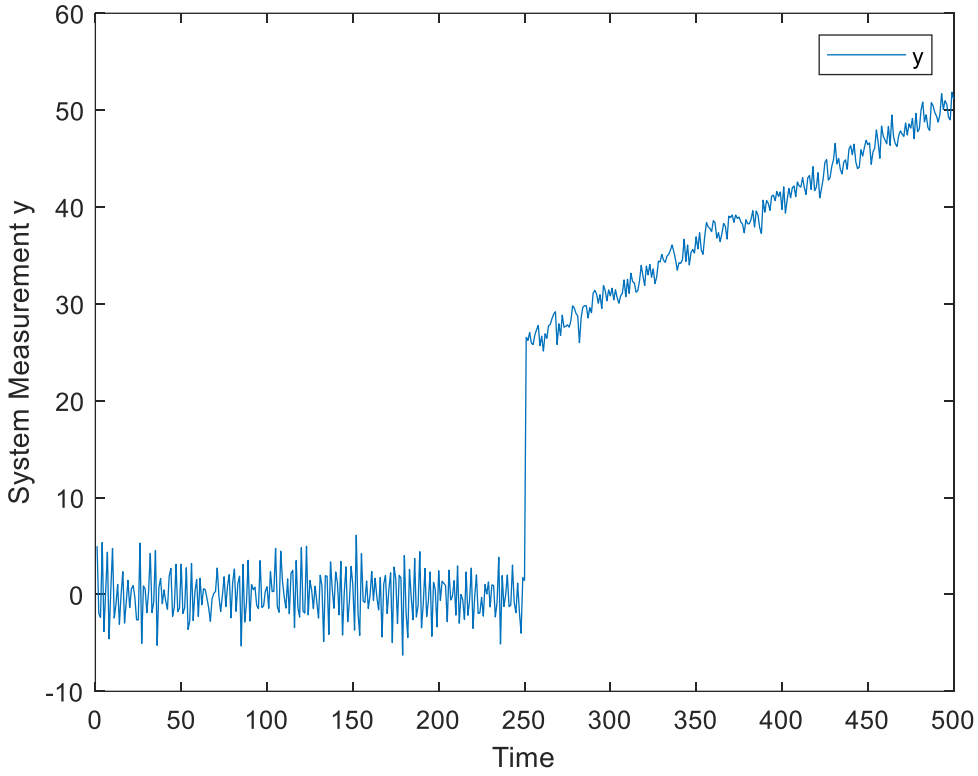

Figure 3.34: The Second-Order Discrete-Time Stochastic system measurement response with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Step and Ramp-Type sensor intrusion happens at shiftpoint $k = 250$.

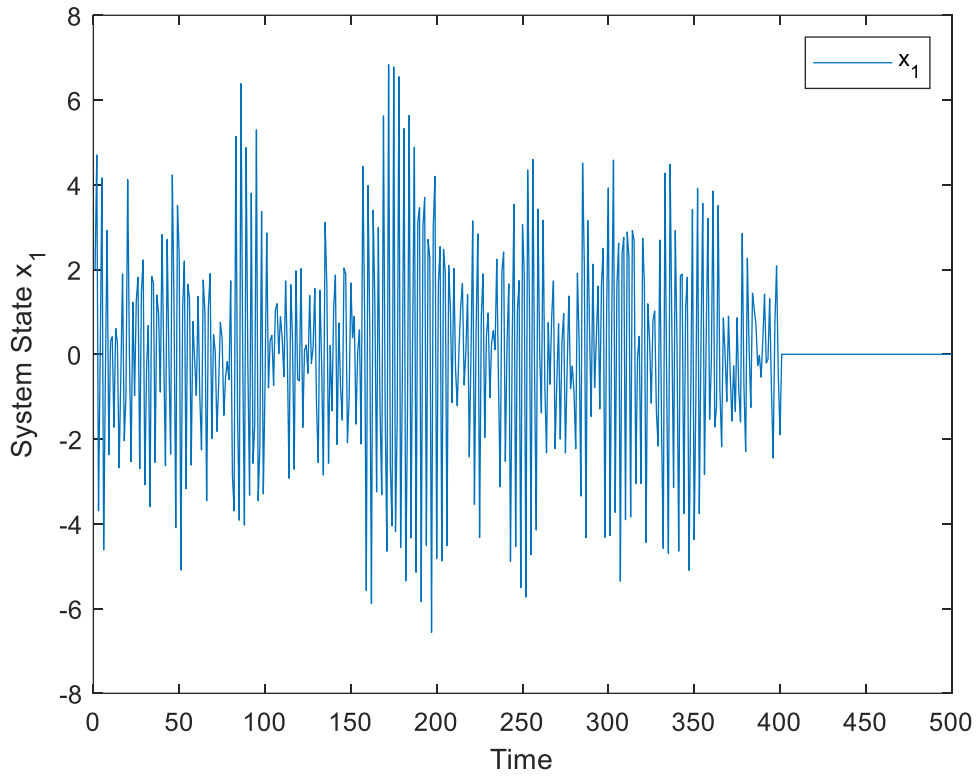

Figure 3.35: The Second-Order Discrete-Time Stochastic system state response x_1 with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Step and Ramp-Type sensor intrusion happens at shiftpoint $k = 400$.

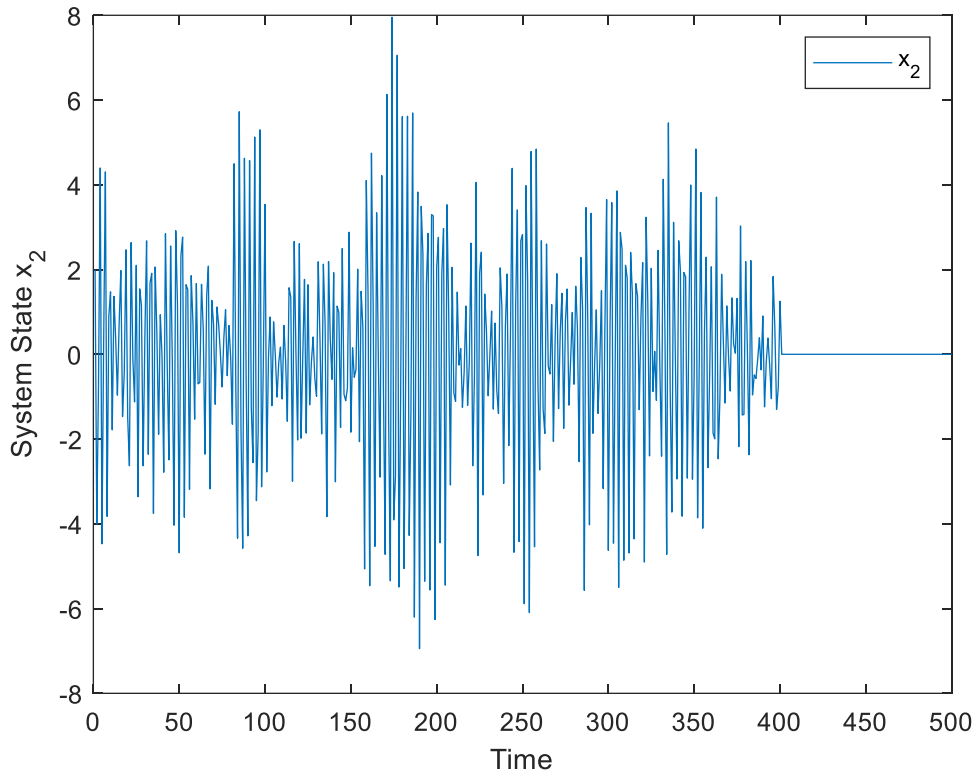

Figure 3.36: The Second-Order Discrete-Time Stochastic system state response x_1 with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Step and Ramp-Type sensor intrusion happens at shiftpoint $k = 400$.

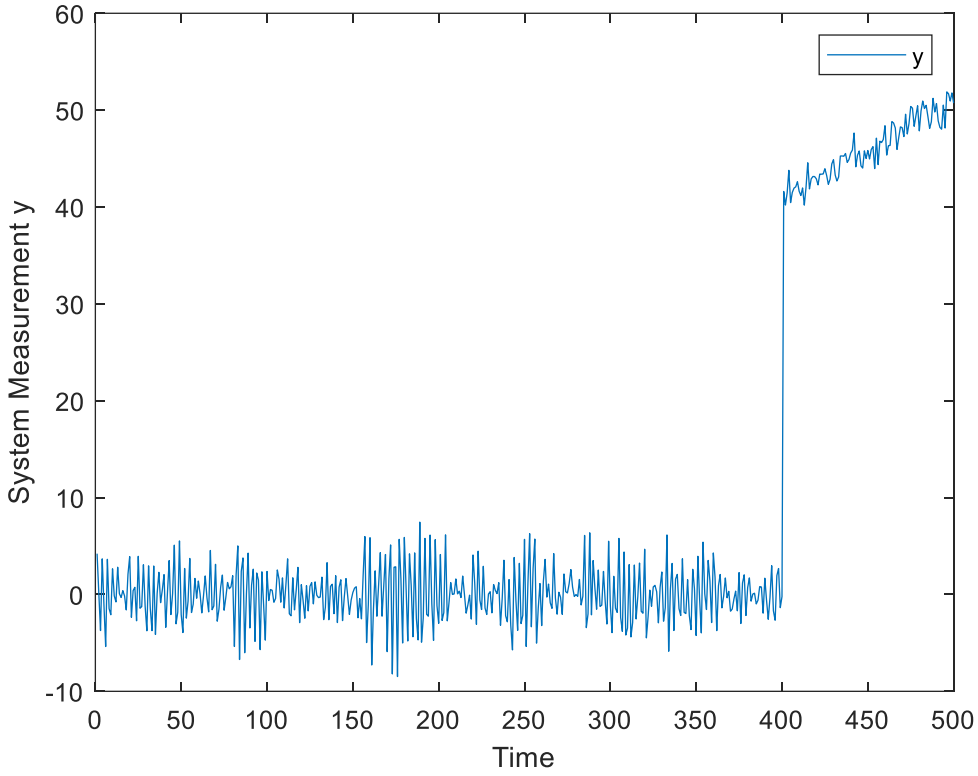

Figure 3.37: The Second-Order Discrete-Time Stochastic system measurement response with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ when the Step and Ramp-Type sensor intrusion happens at shiftpoint $k = 400$.

4 DETECTION OF ATTACKS AND CASE STUDIES

In this chapter, the Constant-Type Intrusion Signal and the Ramp-Type Intrusion Signal targeting the systems mentioned in chapter 3 will be detected by using a bank of Kalman filters algorithm which is introduced in chapter 2. The estimated value of each system states and measurements will be shown and the probabilities of the state (affected/unaffected) of each control system will be calculated as a function of time. The estimation of the states from a bank of Kalman filters together with the associated probabilities will also be calculated as a function of time and, by showing the probabilities of each state (affected/unaffected) based on the data from the bank of Kalman filters, it's determined whether the sensor is under attack or not. The performance of the algorithm will be tested based on various levels of system and measurement noises. An alternative estimation method (sample mean method) will also be introduced and the performance of that algorithm will also be shown.

4.1 First-Order System with Constant-Type Intrusion Signal

Consider the first-order system mentioned in chapter 3, where the system is shown as (4.1a) and (4.1b),

$$x_{k+1} = 0.9x_k + v_k \tag{4.1a}$$

$$y_k^1 = x_k + w_k \tag{4.1b}$$

where $A^1 = 0.9,\ B^1 = 0, C^1 = 1, D^1 = 0, F^1 = 1, G^1 = 1$, and the covariance of the system state noise $V = 0.1$, the covariance of the system measurement noise $W = 0.05$ and both system state noise and system measurement noise are Gaussian. Note that the superscript 1 represents the system is currently not under attack.

Before estimating system state and measurement, the observability of this first-order system needs to be checked and it could be checked easily by using the system matrices A^1 and C^1 with the observability criteria, which shows the system is observable.

Then, the system estimated state and measurement can be observed using the Kalman Filter by setting up the system's initial state estimate $\hat{x}_0$ and initial error covariance P_0^1 based on system's uncertainty. According to (2.10), (2.11), (2.12) and (2.13), the Kalman Filter can be expressed as below

$$P_{k+1}^1 = A^1 P_k^1 A^{1^T} + F^1 V F^{1^T} - A^1 P_k^1 C^{1^T} \left(C^1 P_k^1 C^{1^T} + G^1 W G^{1^T}\right)^{-1} \left(C^1 P_k^1 A^{1^T}\right) \quad (4.2)$$

$$K_k^1 = A^1 P_k^1 C^{1^T} \left(C^1 P_k^1 C^{1^T} + G^1 W G^{1^T}\right)^{-1} \quad (4.3)$$

$$\hat{x}_{k+1} = A^1 \hat{x}_k + K_k^1 \tilde{y}_k^1 \quad (4.4)$$

$$\tilde{y}_k^1 = y_k^1 - C^1 \hat{x}_k \quad (4.5)$$

Here, the system measurement estimate is defined as

$$\hat{y}_k^1 = C^1 \hat{x}_k \tag{4.6}$$

The attack model of this first-order system can be known from (3.3a) and (3.3b)

$$\begin{bmatrix} x_{k+1} \\ h_{k+1} \end{bmatrix} = \begin{bmatrix} 0.9 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ h_k \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} v_k \tag{4.7a}$$

$$y_k^2 = \begin{bmatrix} 0.05 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ h_k \end{bmatrix} + w_k \tag{4.7b}$$

From the previous chapter, it could be known that h_k is a constant-type intrusion signal where $h_k = h_{k+1} = 10$, and it can be found that the whole system is changed to a second-order system with its associated intrusion signal, where $A^2 = \begin{bmatrix} 0.9 & 0 \\ 0 & 1 \end{bmatrix}$, $B^2 = 0$, $C^2 = [0.05 \quad 1]$, $D^2 = 0$, $F^2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $G^2 = 1$. Note that superscript 2 represents the system is currently under attack.

Similarly, the observability of this second-order attack model needs to be checked, by submitting A^2 and C^2 into O^2, where

$$O^2 = \begin{bmatrix} C_2 \\ C_2 A_2 \end{bmatrix} = \begin{bmatrix} 0.05 & 1 \\ 0.045 & 1 \end{bmatrix}$$

Here, O^2 represents the observability matrix for this second-order attack model and it can be known that O^2 is full rank, which shows the system is observable.

Knowing the observability of this attack model, the estimated state and measurement of this second-order system can be estimated using the Kalman Filter by setting up the system's initial state estimate $\hat{x}_0$, initial value of the intrusion signal estimate h_0 and initial error covariance P_0^2 based on this system state's uncertainty. According to (2.10), (2.11), (2.12) and (2.13), the Kalman Filter can be expressed as below

$$P_{k+1}^2 = A^2 P_k^2 A^{2^T} + F^2 V F^{2^T} - A^2 P_k^2 C^{2^T} \left(C^2 P_k^2 C^{2^T} + G^2 W G^{2^T}\right)^{-1} \left(C^2 P_k^2 A^{2^T}\right) \quad (4.8)$$

$$K_k^2 = A^2 P_k^2 C^{2^T} \left(C^2 P_k^2 C^{2^T} + G^2 W G^{2^T}\right)^{-1} \quad (4.9)$$

$$\hat{x}_{k+1} = A^2 \hat{x}_k + K_k^2 \tilde{y}_k^2 \quad (4.10)$$

$$\tilde{y}_k^2 = y_k^2 - C^2 \hat{x}_k \quad (4.11)$$

Here, the system measurement estimate could also be known form above, where

$$\hat{y}_k^2 = C^2 \hat{x}_k \quad (4.12)$$

After calculating both system measurement estimates $\hat{y}_k^1$ and $\hat{y}_k^2$system innovation terms $\tilde{y}_k^1$ and $\tilde{y}_k^2$, the conditional probabilities of each hypothesis θ_i could also be found. Note that $\theta = \{\theta_1, \theta_2\}$, where θ_1 represents the hypothesis when the system is not under attack and θ_2 represents the system is under attack. According (2.15), (2.16), (2.17) and (2.18), the conditional probabilities of each hypothesis $p(\theta_i|Y_k)$ could be found.

From the chapter 2, the first step of getting $p(\theta_i|Y_k)$ is to calculate each covariance for each Kalman Filter with its corresponding hypothesis $\Omega_{k|\theta_i}$, where

$$\Omega_{k|\theta_1} = C^1 P_{k|\theta_1} C^{1^T} + G^1 W G^{1^T} \tag{4.13}$$

Here, $\Omega_{k|\theta_1}$ represents the covariance for the hypothesis θ_1, where the system is currently not under attack and $P_{k|\theta_1} = P_k^1$. Thus, $\Omega_{k|\theta_1}$ could be calculated by submitting system matrices C_1, G_1, system measurement noise W and system error covariance P_k^1 from (4.2).

Similarly, $\Omega_{k|\theta_2}$ represents the covariance for the hypothesis θ_2, where the system is currently under attack and $P_{k|\theta_2} = P_k^2$. Thus, $\Omega_{k|\theta_2}$ could be calculated by submitting system matrices C_2, G_2, system measurement noise W and system error covariance P_k^2 from (4.8). and $\Omega_{k|\theta_2}$ could be shown as below

$$\Omega_{k|\theta_2} = C^2 P_{k|\theta_2} C^{2^T} + G^2 W G^{2^T} \tag{4.14}$$

After calculating $\Omega_{k|\theta_1}$ and $\Omega_{k|\theta_2}$, the second step of getting $p(\theta_i|Y_k)$ is to calculate the likelihood function $p(y_k|Y_{k-1}, \theta_i)$, where it is part of the probability density function in (2.15) and it could be known from (2.16) by submitting $\Omega_{k|\theta_1}$, $\Omega_{k|\theta_2}$, $\tilde{y}_{k|\theta_1}$, and $\tilde{y}_{k|\theta_2}$. Note that $\tilde{y}_{k|\theta_1} = \tilde{y}_k^1$ and $\tilde{y}_{k|\theta_2} = \tilde{y}_k^2$,.

$$p(y_k|Y_{k-1}, \theta_1) = (2\pi)^{-m/2}\left|\Omega_{k|\theta_1}^{-1}\right|^{1/2} \exp\left\{-\frac{1}{2}\tilde{y}_{k|\theta_1}^T \Omega_{k|\theta_1}^{-1}\tilde{y}_{k|\theta_1}\right\} \tag{4.15}$$

$$p(y_k|Y_{k-1}, \theta_2) = (2\pi)^{-m/2}\left|\Omega_{k|\theta_2}^{-1}\right|^{1/2} \exp\left\{-\frac{1}{2}\tilde{y}_{k|\theta_2}^T \Omega_{k|\theta_2}^{-1}\tilde{y}_{k|\theta_2}\right\} \tag{4.16}$$

In (4.15), $p(y_k|Y_{k-1}, \theta_1)$ represents the likelihood function when the system is not under attack, and $m = 1$ because the unhacked system is a first order system as mentioned in (4.1a) and (4.1b). Similarly, $p(y_k|Y_{k-1}, \theta_2)$ represents the likelihood function in (4.16) when the system is under attack, and $m = 2$ because when the system is under attack, it will become a second order system mentioned in (4.7a) and (4.7b) with its associate constant-type intrusion signal.

After calculating the likelihood function for both cases (hacked/ not hacked), the conditional probabilities of each hypothesis $p(\theta_i|Y_k)$ could be found by submitting (4.15) and (4.16) to (2.15) separately, where

$$p(\theta_1|Y_k) = \frac{p(y_k|Y_{k-1},\theta_1)p(\theta_1|Y_{k-1})}{\sum_{i=1}^{N} p(y_k|Y_{k-1},\theta_1)p(\theta_1|Y_{k-1})} \tag{4.17}$$

$$p(\theta_2|Y_k) = \frac{p(y_k|Y_{k-1},\theta_2)p(\theta_2|Y_{k-1})}{\sum_{i=1}^{N} p(y_k|Y_{k-1},\theta_2)p(\theta_2|Y_{k-1})} \tag{4.18}$$

Here, (4.17) and (4.18) can be solved recursively, and the calculation would begin with an initial probability $p(\theta_i|Y_0)$ between 0 and 1 when $k = 0$, note that $p(\theta_i|Y_{k-1})$ is the previous value of $p(\theta_i|Y_k)$.

The simulation results can be shown by submitting the two systems (hacked/unhacked) matrices into a bank of Kalman Filters and Fig. 4.1 shows the changes of the system measurement before and after it's being hacked.

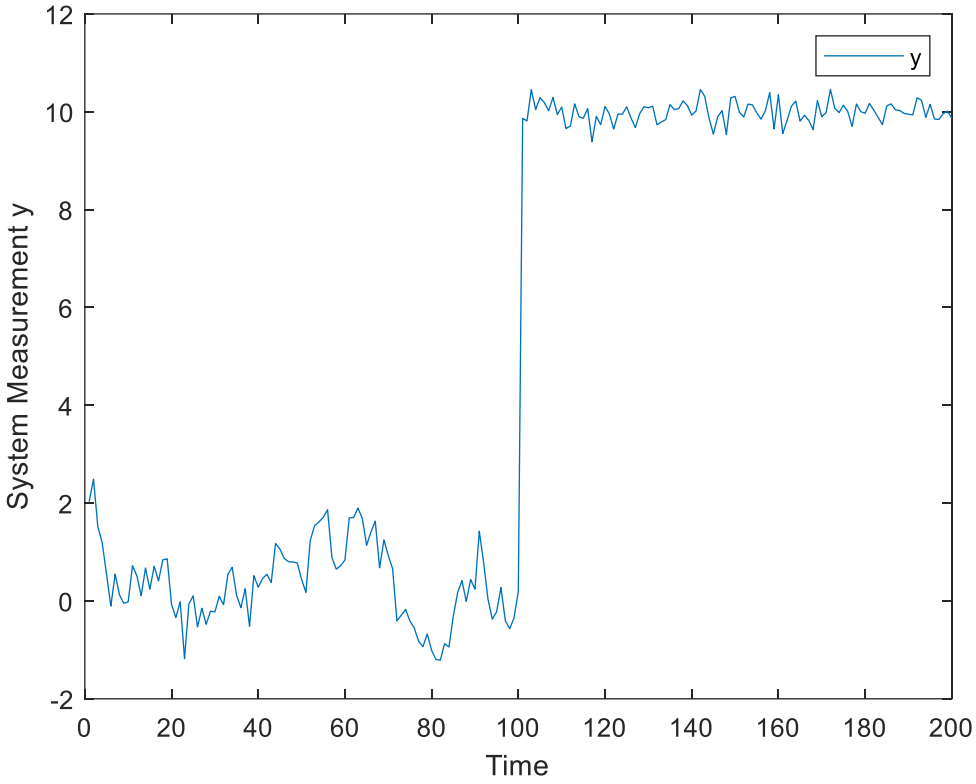

Figure 4.1: The First-Order Discrete-Time stochastic system measurement y with its initial state $x_0 = 2$ and the Constant-Type sensor intrusion happens at shiftpoint $k = 100$.

As mentioned, before the constant-type intrusion signal enters this first-order system, the system measurement is $y_k^1 = x_k + w_k$ like what Fig 4.1 shows before $k = 100$. After that, the intrusion signal replaces the system measurement so that $y_k^2 = [0.05 \quad 1]\begin{bmatrix} x_k \\ h_k \end{bmatrix} + w_k$ like what Fig 4.1 shows after $k = 100$.

After showing the system measurement, the next step is to calculate the system state and measurement estimate by using a bank of Kalman Filters. By submitting (4.2), (4.3), (4.4) and (4.6) to (4.5) with its initial state estimate $\hat{x}_0 = 1$ and initial error

covariance $P_0^1 = 100$, the innovation term $\tilde{y}_k^1$ could be calculated. Similarly, by submitting (4.8), (4.9), (4.10) and (4.12) to (4.11) with its initial state estimate $\begin{bmatrix}\hat{x}_0 \\ \hat{h}_0\end{bmatrix} = \begin{bmatrix}0 \\ 0\end{bmatrix}$ and initial error covariance $P_0^2 = \begin{bmatrix}100 & 0 \\ 0 & 100\end{bmatrix}$, the innovation term $\tilde{y}_k^2$ could also be calculated. Fig 4.2 shows the innovation terms $\tilde{y}_k^1$ and $\tilde{y}_k^2$ in time

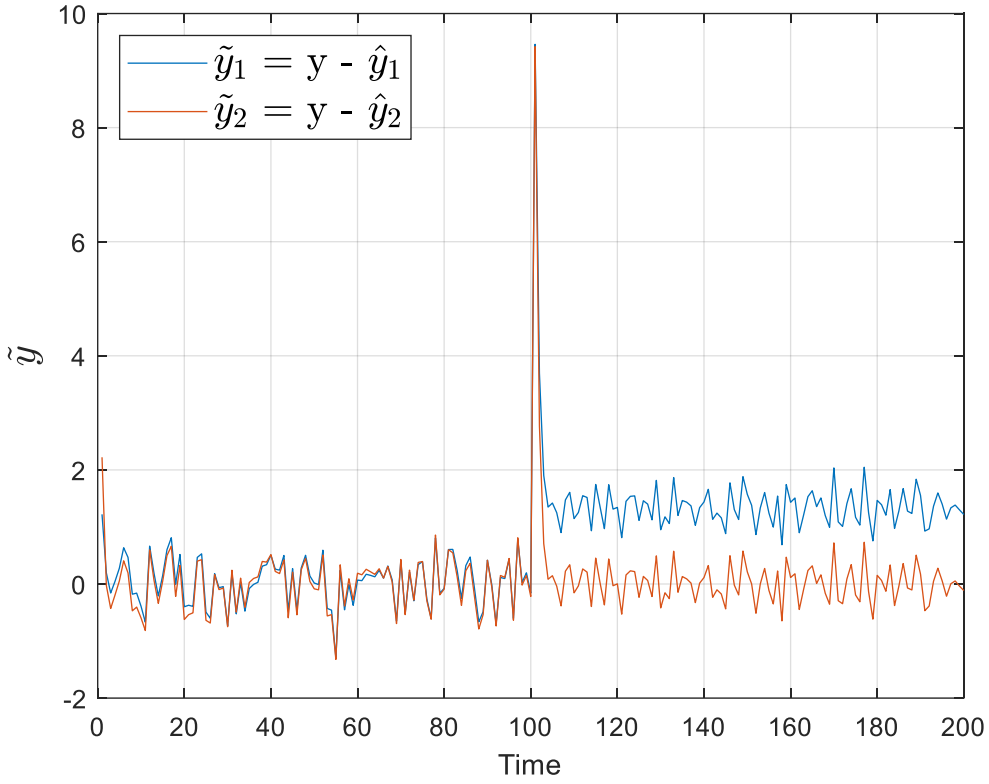

Figure 4.2: The innovation terms $\tilde{y}_k^1$ and $\tilde{y}_k^2$ when there is a constant-type intrusion signal at shiftpoint $k = 100$

After finding the innovation term $\tilde{y}_k^1$ and $\tilde{y}_k^2$, the likelihood function (4.15) and (4.16) could be calculated and the conditional probabilities of each hypothesis $p(\theta_1|Y_k)$ and $p(\theta_2|Y_k)$ could be found by submitting the likelihood function (4.15) and (4.16) to (2.15) separately and by setting up both the initial probability $p(\theta_1|Y_0) = 0.5$ and

$p(\theta_2|Y_0) = 0.5$ when $k = 0$, the conditional probabilities of each hypothesis can be shown as Fig 4.3, note that the intrusion happens at shiftpoint $k = 100$

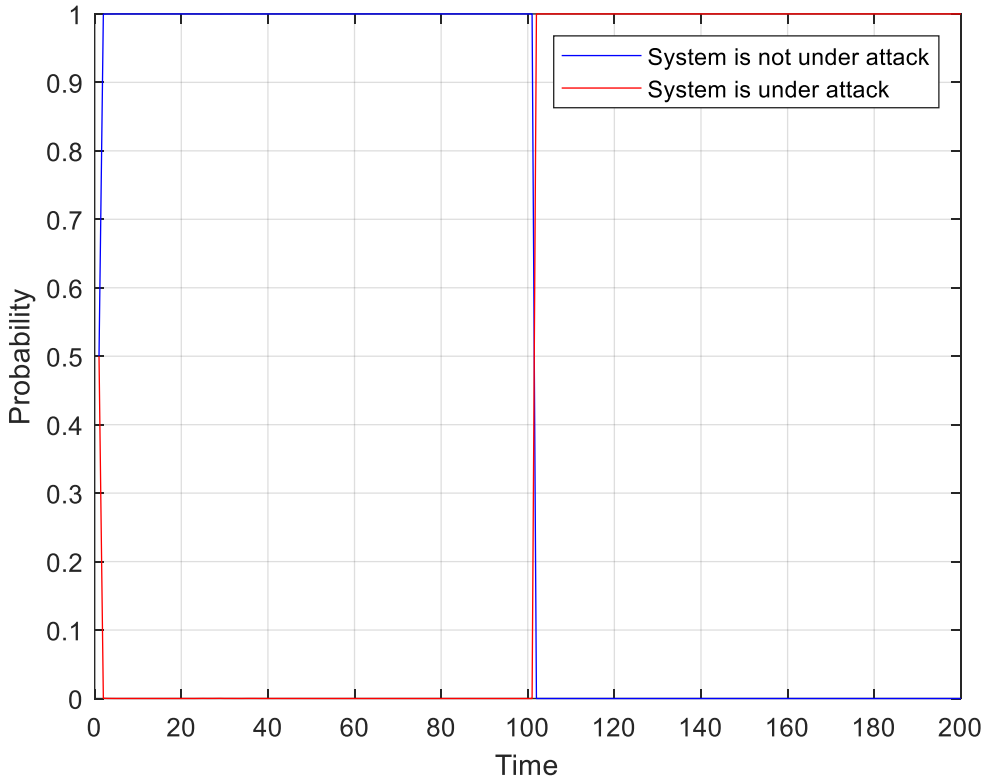

Figure 4.3: Conditional probabilities of each hypothesis $p(\theta_1|Y_k)$ (unhacked case) and $p(\theta_2|Y_k)$ (hacked case) when there is a constant-type intrusion signal enters the system at shiftpoint $k = 100$, starting with each initial probability $p(\theta_1|Y_0) = 0.5$ and $p(\theta_2|Y_0) = 0.5$

From Fig 4.3, it could be noticed that both the hacked and unhacked cases are all start with the initial probability $p(\theta_i|Y_0) = 0.5$. The conditional probability for the unhacked case $p(\theta_1|Y_k)$ convergences to 1 very quickly and keeps convergence until the shiftpoint $k = 100$, and on the other hand, the conditional probability for the hacked case $p(\theta_2|Y_k)$ convergences to 0 very quickly and keeps convergence until the shiftpoint $k = 100$, which means that the system is not under attack when $k < 100$. When the shiftpoint

$k = 100$, which represents the system measurement is now replaced by the constant-type intrusion signal and now the conditional probability for the unhacked case $p(\theta_1|Y_k)$ convergences to 0 from 1 very quickly and keeps convergence when the shiftpoint $k > 100$, and on the other hand, the conditional probability for the hacked case $p(\theta_2|Y_k)$ convergences to 1 form 0 very quickly and keeps convergence when the shiftpoint $k > 100$, which means that the system is now under attack when $k > 100$. Note that the convergence time for $p(\theta_1|Y_k) = 1$ when $k < 100$ is $k = 2$, and the convergence time for $p(\theta_2|Y_k) = 1$ when $k > 100$ is $k = 102$, which means the algorithm designed in this thesis works very well under this condition.

4.2 First-Order System with Ramp-Type Intrusion Signal

Consider the first-order system (4.1a) and (4.1b), where the hacker enters the system, replace the system measurement to a step and ramp-type intrusion signal and in this case, the attack model of this first-order system can be known from (3.4a) and (3.4b) and it could be shown below

$$\begin{bmatrix} x_{k+1} \\ h_{k+1}^1 \\ h_{k+1}^2 \end{bmatrix} = \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ h_k^1 \\ h_k^2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} v_k \tag{4.19a}$$

$$y_k^2 = [0.05 \quad 1 \quad 1] \begin{bmatrix} x_k \\ h_k^1 \\ h_k^2 \end{bmatrix} + w_k \tag{4.19b}$$

From the previous chapter, it could be known that h_k is now a ramp-type intrusion signal where $h_{k+1} = \begin{bmatrix} h_{k+1}^1 \\ h_{k+1}^2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} h_k^1 \\ h_k^2 \end{bmatrix}$, and it can be found that the whole system is changed to a third-order system with its associated intrusion signal, where $A^2 = \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$, $B^2 = 0$, $C^2 = [0.05 \quad 1 \quad 1]$, $D^2 = 0$, $F^2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ and $G^2 = 1$. Similarly, superscript 2 represents the system is currently under attack, and the intrusion signal is now a ramp-type signal. Fig 4.4 shows the changes of the system measurement y when the step and ramp-type sensor intrusion happened at shiftpoint $k = 100$.

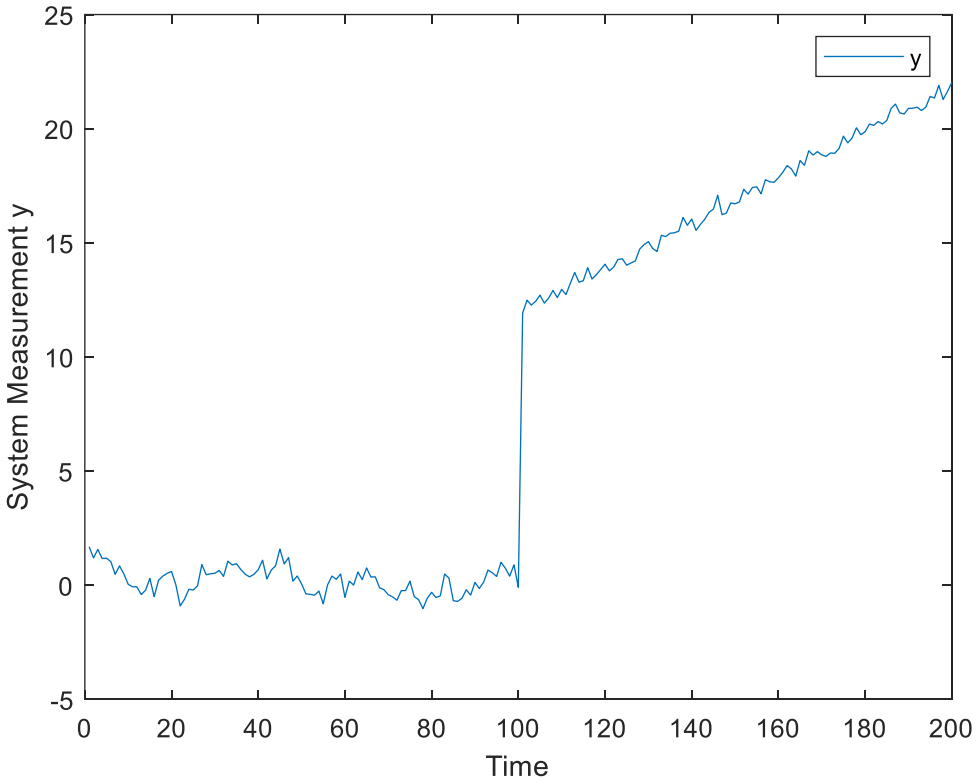

Figure 4.4: The First-Order Discrete-Time stochastic system measurement y with its initial state $x_0 = 2$ and the Step and Ramp-Type sensor intrusion happens at shiftpoint $k = 100$.

As usual, the observability of this third-order system needs to be checked by submitting A^2 and C^2 into O^2, where

$$O^2 = \begin{bmatrix} C^2 \\ C^2A^2 \\ C^2A^{2^2} \end{bmatrix} = \begin{bmatrix} 0.05 & 1 & 1 \\ 0.045 & 1 & 2 \\ 0.0405 & 1 & 3 \end{bmatrix}$$

Here, O_2 represents the observability matrix for this third-order attack model and it can be known that O_2 is full rank, which shows the system is observable.

Similarly, after knowing the system's observability, this third order system estimated state and measurement can be observed using the Kalman Filter by setting up the system's initial state estimate $\hat{x}_0$, initial intrusion signal estimate $\hat{h}_0$ and initial error covariance P_0^2 based on the system's uncertainty. According to (2.10), (2.11), (2.12) and (2.13), the system state estimate and measurement estimate could be calculated and after that, the system innovation terms $\tilde{y}_k^2$ can also be calculated with its initial state estimate $\begin{bmatrix} \hat{x}_0 \\ \hat{h}_0^1 \\ \hat{h}_0^2 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0 \\ 0 \end{bmatrix}$ and initial error covariance $P_0^2 = \begin{bmatrix} 7 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 7 \end{bmatrix}$.

Fig 4.5 shows the innovation terms $\tilde{y}_k^1$ and $\tilde{y}_k^2$ in time, note that the intrusion is the ramp-type signal now

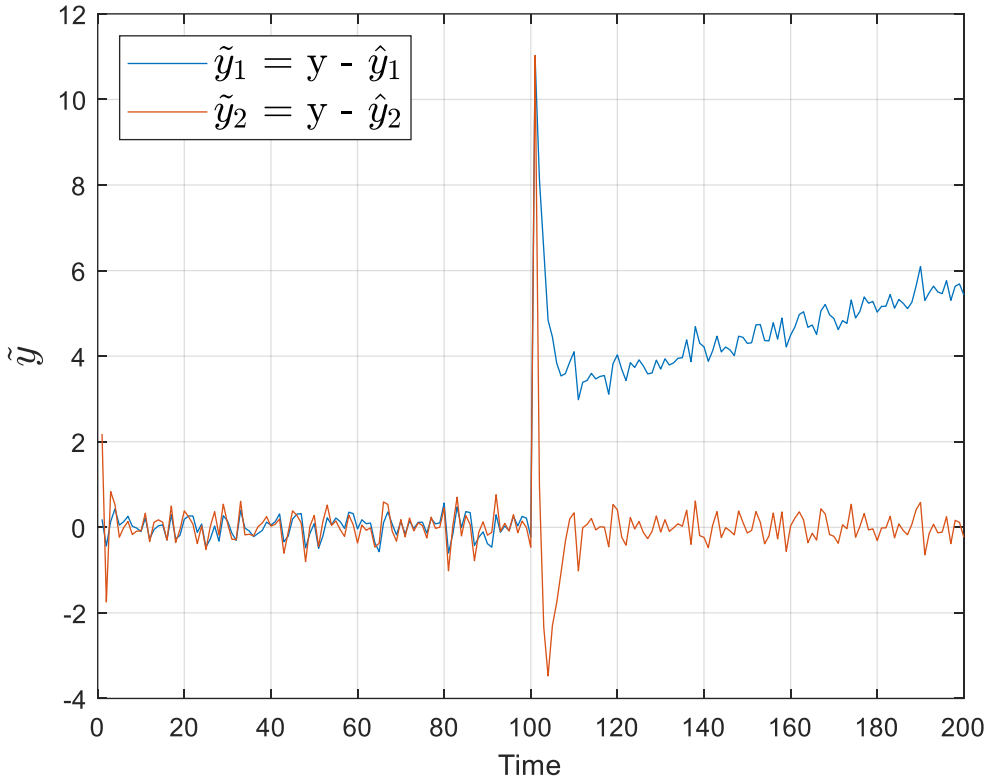

Figure 4.5: The innovation terms $\tilde{y}_k^1$ and $\tilde{y}_k^2$ when there is a step and ramp-type intrusion signal at shiftpoint $k = 100$

Similar to the constant-type intrusion cases, after finding the innovation term $\tilde{y}_k^1$ and $\tilde{y}_k^2$, the likelihood function (4.15) and (4.16) can be calculated and the conditional probabilities of each hypothesis $p(\theta_1|Y_k)$ and $p(\theta_2|Y_k)$ can be found by submitting the likelihood function (4.15) and (4.16) to (2.15) separately and by setting up both the initial probability $p(\theta_1|Y_0) = 0.5$ and $p(\theta_2|Y_0) = 0.5$ when $k = 0$, the conditional probabilities of each hypothesis can be shown as Fig 4.6, note that the intrusion is the step and ramp-type signal and the intrusion happens at shiftpoint $k = 100$

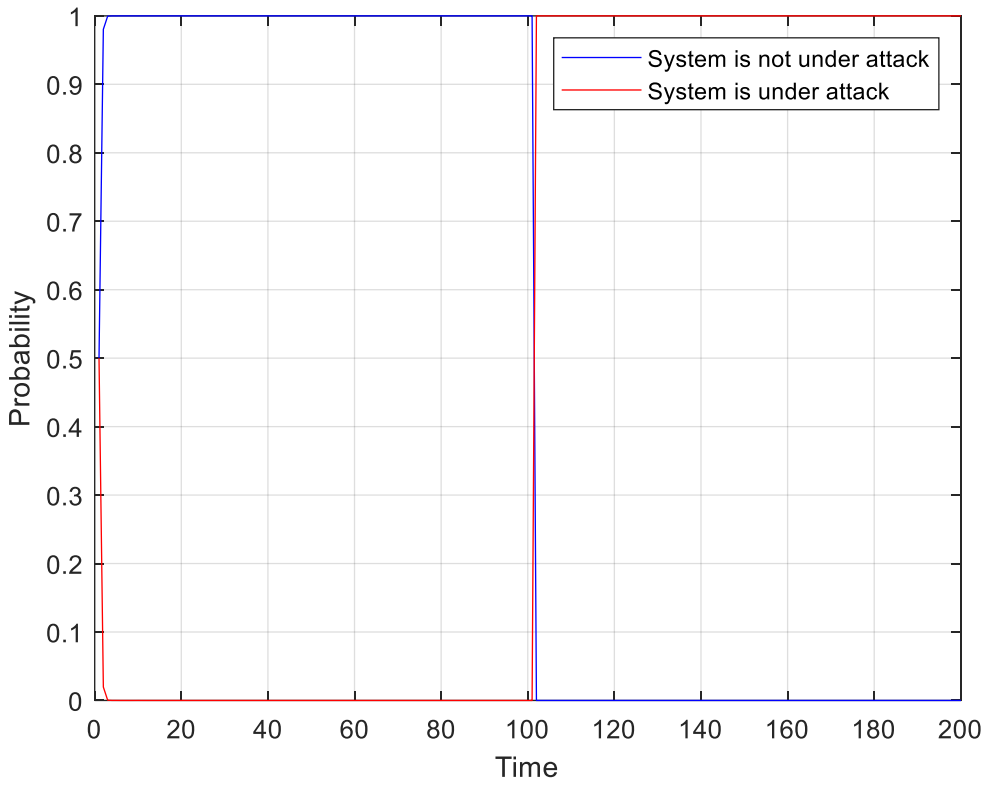

Figure 4.6: Conditional probabilities of each hypothesis $p(\theta_1|Y_k)$ (unhacked case) and $p(\theta_2|Y_k)$ (hacked case) when there is a step and ramp-type intrusion signal enters the system at shiftpoint $k = 100$, starting with each initial probability $p(\theta_1|Y_0) = 0.5$ and $p(\theta_2|Y_0) = 0.5$

From Fig 4.6, it can be noticed that both the hacked and unhacked cases are all start with the initial probability $p(\theta_i|Y_0) = 0.5$. The conditional probability for the unhacked case $p(\theta_1|Y_k)$ convergences to 1 very quickly and keeps convergence until the shiftpoint $k = 100$, and on the other hand, the conditional probability for the hacked case $p(\theta_2|Y_k)$ convergences to 0 very quickly and keeps convergence until the shiftpoint $k = 100$, which means that the system is not under attack when $k < 100$. When the shiftpoint $k = 100$, which represents the system measurement is now replaced by the step and ramp-type intrusion signal and now the conditional probability for the unhacked case

$p(\theta_1|Y_k)$ convergences from 1 to 0 very quickly and keeps convergence when the shiftpoint $k > 100$, and on the other hand, the conditional probability for the hacked case $p(\theta_2|Y_k)$ convergences from 0 to 1 very quickly and also keeps convergence, which means the system is now under attack when $k > 100$. Note that the convergence time for $p(\theta_1|Y_k) = 1$ when $k < 100$ is $k = 9$, and the convergence time for $p(\theta_2|Y_k) = 1$ when $k > 100$ is $k = 102$, which shows this estimation algorithm designed in this thesis works also well under this step and ramp-type intrusion condition.

4.3 Second-Order System with Constant-Type Intrusion Signal

Consider the second-order system mentioned in chapter 3, where the system is shown as (4.20a) and (4.20b),

$$x_{k+1} = \begin{bmatrix} 0 & 0.9 \\ -1 & -1 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 0 \end{bmatrix} v_k \qquad (4.20a)$$

$$y_k^1 = [1 \quad 1] x_k + w_k \qquad (4.20b)$$

where $A^1 = \begin{bmatrix} 0 & 0.9 \\ -1 & -1 \end{bmatrix}$, $B^1 = 0$, $C^1 = [1 \quad 1]$, $D^1 = 0$, $F^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $G^1 = 1$, and the covariance of the system state noise $V = 1$, the covariance of the system measurement noise $\mathrm{W} = 1$ and both system state noise and system measurement noise are Gaussian and by using the observability criteria, the observability of the system could be checked by submitting A^1 and C^1 into O^1

$$O^1 = \begin{bmatrix} C^1 \\ C^1A^1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -1 & -0.1 \end{bmatrix}$$

Here, O^1 represents the observability matrix for this second-order system and it can be known that O^1 is full rank, which shows the system is observable.

In the same time, the attack model of this second-order system can be known from (3.8a) and (3.8b)

$$\begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \\ h_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 0.9 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \\ h_k \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} v_k \qquad (4.21a)$$

$$y_k^2 = [0 \quad 0.1 \quad 1] x_k + w_k \qquad (4.21b)$$

Here, h_k is a constant-type intrusion signal where $h_k = h_{k+1} = 10$, and it can be found that the whole system is changed to a third-order system with its associated intrusion signal, where $A^2 = \begin{bmatrix} 0 & 0.9 & 0 \\ -1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $B^2 = 0$, $C^2 = [0 \quad 0.1 \quad 1]$, $D^2 = 0$, $F^2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ and $G^2 = 1$.

Same as mentioned before, the observability of this third-order system needs to be checked by submitting A^2 and C^2 into O^2

$$O^2 = \begin{bmatrix} C^2 \\ C^2C^2 \\ C^2A^{2^2} \end{bmatrix} = \begin{bmatrix} 0 & 0.1 & 1 \\ -0.1 & -0.1 & 1 \\ 0.1 & 0.01 & 1 \end{bmatrix}$$

Here, O^2 represents the observability matrix for this third-order attack model and it can be known that O^2 is full rank, which shows the system is observable. Fig 4.7 shows the system measurement y when the constant-type sensor intrusion happens at shiftpoint $k = 250$.

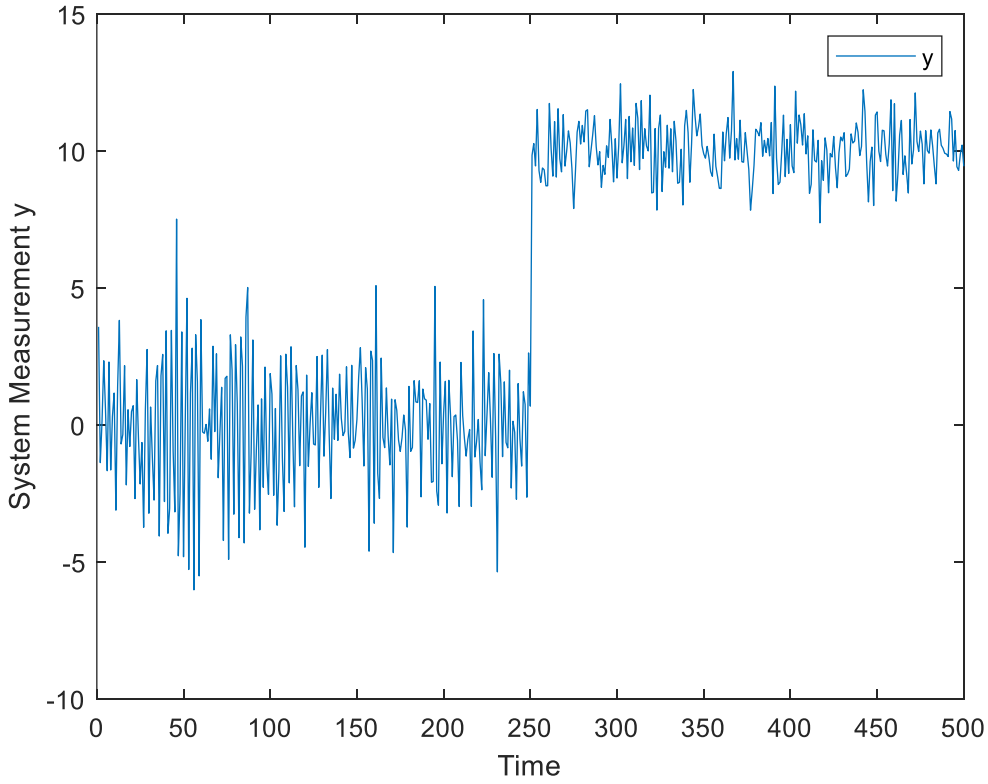

Figure 4.7: The Second-Order Discrete-Time stochastic system measurement y with its initial state $x_0 = [2 \quad 2]$ and the Constant-Type sensor intrusion happens at shiftpoint $k = 250$.

Before the constant-type intrusion signal enters the system, the system measurement is $y_k^1 = [1 \quad 1]x_k + w_k$ like Fig 4.7 shows before $k = 250$. After that, the intrusion signal replaces the system measurement so that $y_k^2 = [0 \quad 0.1 \quad 1]x_k + w_k$ like Fig 4.7 shows after $k = 250$.

After knowing the system measurement, the next step is to calculate the system state and measurement estimate by using a bank of Kalman Filters. By submitting (4.2), (4.3) and (4.4) to (4.5) and (4.5) with its initial state estimate $\hat{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and initial error covariance $P_0^1 = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$, the system measurement estimate $\hat{y}_k^1$ and the innovation term $\tilde{y}_k^1$ can be calculated. Similarly, by submitting (4.8), (4.9) and (4.10) to (4.11) and (4.12) with its initial state estimate $\begin{bmatrix} \hat{x}_0 \\ \hat{h}_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ and initial error covariance $P_0^2 = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}$, the system measurement estimate $\hat{y}_k^2$ and the innovation term $\tilde{y}_k^2$ can also be calculated. Fig 4.8 shows the innovation terms $\tilde{y}_k^1$ and $\tilde{y}_k^2$ in time

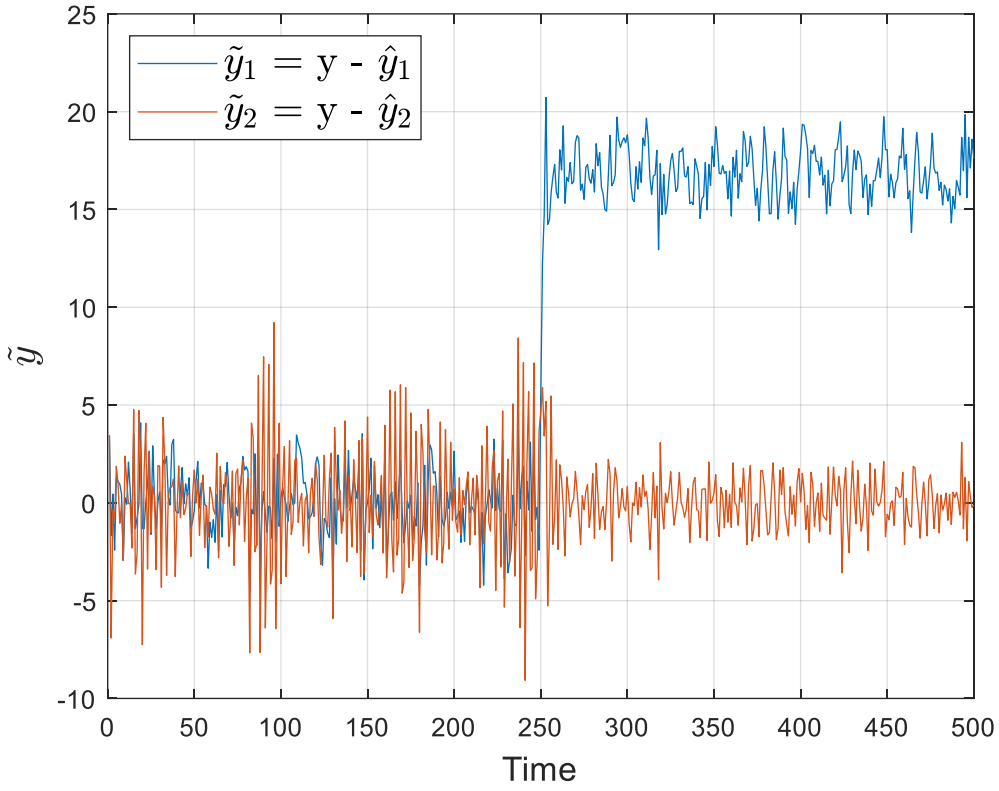

Figure 4.8: The innovation terms $\tilde{y}_k^1$ and $\tilde{y}_k^2$ for the second-order system when there is a constant-type intrusion signal at shiftpoint $k = 250$.

Like the first-order cases introduced previously in this chapter, after finding the innovation term $\tilde{y}_k^1$ and $\tilde{y}_k^2$, the likelihood function (4.15) and (4.16) can be calculated and the conditional probabilities of each hypothesis $p(\theta_1|Y_k)$ and $p(\theta_2|Y_k)$ can be found by submitting the likelihood function (4.15) and (4.16) to (2.15) separately and by setting up both the initial probability $p(\theta_1|Y_0) = 0.5$ and $p(\theta_2|Y_0) = 0.5$ when $k = 0$, the conditional probabilities of each hypothesis can be shown as Fig 4.9, note that the intrusion happens at shiftpoint $k = 250$

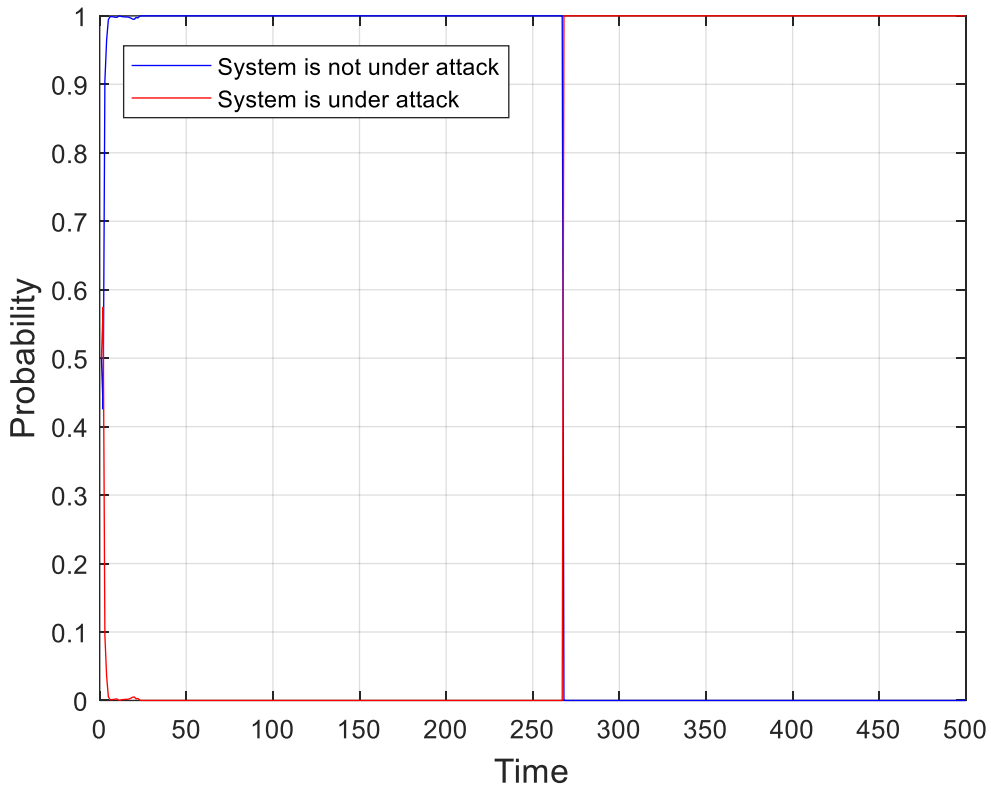

Figure 4.9: Conditional probabilities of each hypothesis $p(\theta_1|Y_k)$ (unhacked case) and $p(\theta_2|Y_k)$ (hacked case) when there is a constant-type intrusion signal enters the second-order system at shiftpoint $k = 250$, starting with each initial probability $p(\theta_1|Y_0) = 0.5$ and $p(\theta_2|Y_0) = 0.5$

From Fig 4.9, it can be noticed that both the hacked and unhacked cases are all start with the initial probability $p(\theta_i|Y_0) = 0.5$. The conditional probability for the unhacked case $p(\theta_1|Y_k)$ convergences to 1 very quickly and keeps convergence until the shiftpoint $k = 250$, and on the other hand, the conditional probability for the hacked case $p(\theta_2|Y_k)$ convergences to 0 very quickly and keeps convergence until the shiftpoint $k = 250$, which means that the system is not under attack when $k < 250$. When the shiftpoint $k = 250$, which represents the system measurement is now replaced by the constant-type intrusion signal and now the conditional probability for the unhacked case $p(\theta_1|Y_k)$

convergences from 1 to 0 very quickly and keeps convergence when the shiftpoint $k > 250$, and on the other hand, the conditional probability for the hacked case $p(\theta_2|Y_k)$ convergences from 0 to 1 very quickly and also keeps convergence, which means the system is now under attack when $k > 250$. Note that the convergence time for $p(\theta_1|Y_k) = 1$ when $k < 250$ is $k = 3$, and the convergence time for $p(\theta_2|Y_k) = 1$ when $k > 250$ is $k = 264$, which shows this estimation algorithm could also works well for the second-order system when there is a constant-type intrusion enters the system.

4.4 Second-Order System with Step and ramp-type Intrusion Signal

Consider the second-order system (4.20a) and (4.20b), where the hacker enters the system, replace the system measurement to a step and ramp-type intrusion signal and in this case, the attack model of this second-order system can be known from (3.9a) and (3.9b) and it could be shown below

$$\begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \\ h_{k+1}^1 \\ h_{k+1}^2 \end{bmatrix} = \begin{bmatrix} 0 & 0.9 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \\ h_k^1 \\ h_k^2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} v_k \tag{3.9a}$$

$$y_k^2 = [0 \quad 0.1 \quad 1 \quad 0]x_k + w_k \tag{3.9b}$$

From the previous chapter, it could be known that h_k is a step and ramp-type intrusion signal where $h_{k+1} = \begin{bmatrix} h_{k+1}^1 \\ h_{k+1}^2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} h_k^1 \\ h_k^2 \end{bmatrix}$, and it can be found that the whole

system is changed to a fourth-order system with its associated intrusion signal, where

$A^2 = \begin{bmatrix} 0 & 0.9 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, $B^2 = 0$, $C^2 = [0 \quad 0.1 \quad 1 \quad 0]$, $D^2 = 0$, $F^2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ and $G^2 =$ 1. Similarly, superscript 2 represents the system is currently under attack, and the intrusion signal is now a step and ramp-type signal. Fig 4.10 shows the system measurement y when the step and ramp-type sensor intrusion happened at shiftpoint $k = 250$.

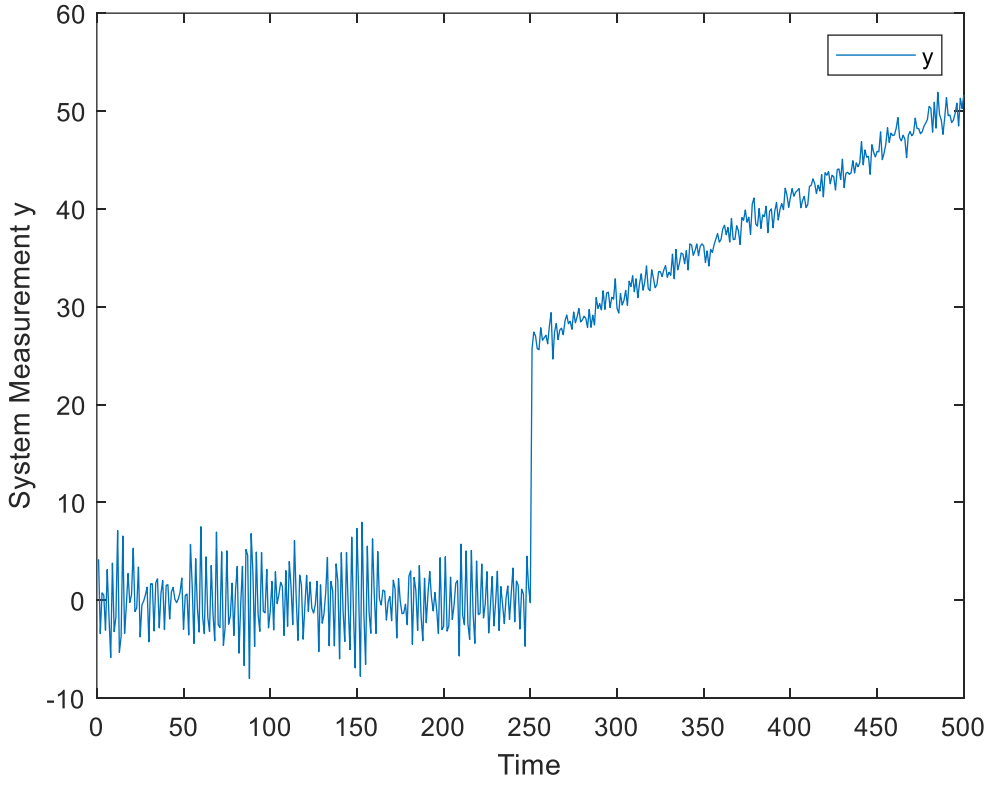

Figure 4.10: The Second-Order Discrete-Time stochastic system measurement y with its initial state $x_0 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$ and the Step and ramp-type sensor intrusion happens at shiftpoint $k = 250$.

Same as mentioned before, the observability of this fourth-order system needs to be checked by submitting A^2 and C^2 into O^2

$$O^2 = \begin{bmatrix} C^2 \\ C^2A^2 \\ C^2A^{2^2} \\ C^2A^{2^3} \end{bmatrix} = \begin{bmatrix} 0 & 0.1 & 1 & 0 \\ -0.1 & -0.1 & 1 & 1 \\ 0.1 & 0.01 & 1 & 2 \\ -0.01 & 0.08 & 1 & 3 \end{bmatrix}$$

Here, O^2 represents the observability matrix for this fourth-order attack model and it can be known that O^2 is a full rank fourth-order matrix, which shows the system is observable.

Knowing the system's observability, this fourth-order system estimated state and measurement can be monitored using the Kalman Filter by setting up the system's initial state estimate $\hat{x}_0$, initial intrusion signal estimate $\hat{h}_0$ and initial error covariance P_0^2 based on the system state uncertainty. According to (2.10), (2.11), (2.12) and (2.13), the system state estimate and measurement estimate can be calculated and after that, the system innovation terms $\tilde{y}_k^2$ can also be calculated with its initial state estimate $\begin{bmatrix} \hat{x}_0 \\ \hat{h}_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ and initial error covariance $P_0^2 = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}$. Fig 4.5 shows the innovation terms $\tilde{y}_k^1$ and $\tilde{y}_k^2$ in time

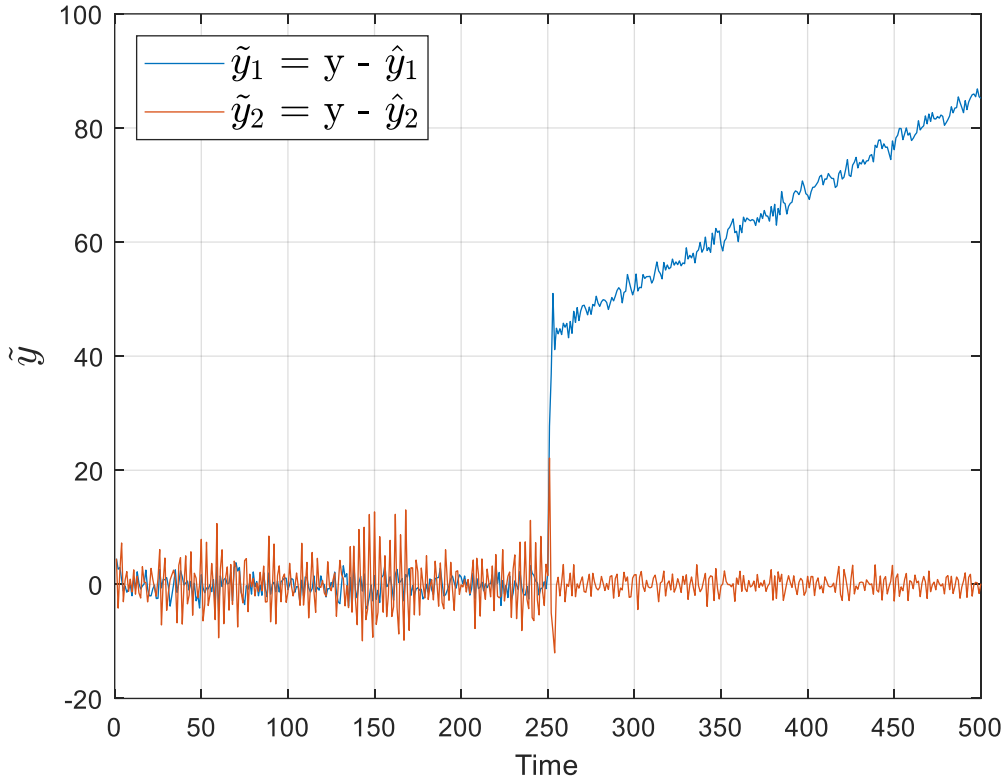

Figure 4.11: The innovation terms$\tilde{y}_k^1$ and $\tilde{y}_k^2$ for the second-order system when there is a step and ramp-type intrusion signal at shiftpoint $k = 250$

Similar with the constant-type intrusion cases, after finding the innovation term $\tilde{y}_k^1$ and $\tilde{y}_k^2$, the likelihood function (4.15) and (4.16) can be calculated and the conditional probabilities of each hypothesis $p(\theta_1|Y_k)$ and $p(\theta_2|Y_k)$ can be found by submitting the likelihood function (4.15) and (4.16) to (2.15) separately and by setting up both the initial probability $p(\theta_1|Y_0) = 0.5$ and $p(\theta_2|Y_0) = 0.5$ when $k = 0$, the conditional probabilities of each hypothesis can be shown as Fig 4.12, note that the intrusion happens at shiftpoint $k = 250$

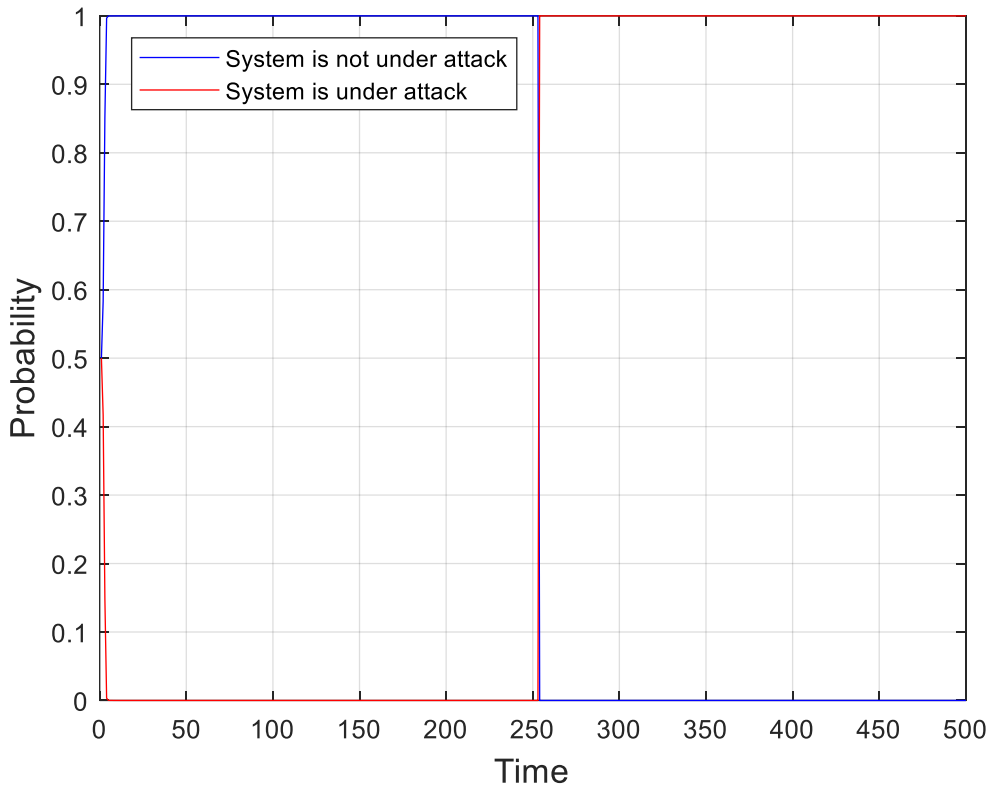

Figure 4.12: Conditional probabilities of each hypothesis $p(\theta_1|Y_k)$ (unhacked case) and $p(\theta_2|Y_k)$ (hacked case) when there is a step and ramp-type intrusion signal enters the second-order system at shiftpoint $k = 100$, starting with each initial probability $p(\theta_1|Y_0) = 0.5$ and $p(\theta_2|Y_0) = 0.5$

From Fig 4.12, it can be noticed that both the hacked and unhacked cases are all start with the initial probability $p(\theta_i|Y_0) = 0.5$. The conditional probability for the unhacked case $p(\theta_1|Y_k)$ convergences to 1 very quickly and keeps convergence until the shiftpoint $k = 250$, and on the other hand, the conditional probability for the hacked case $p(\theta_2|Y_k)$ convergences to 0 very quickly and keeps convergence until the shiftpoint $k = 250$, which means that the system is not under attack when $k < 250$. When the shiftpoint $k = 250$, the system measurement is replaced by the constant-type intrusion signal and the conditional probability for the unhacked case $p(\theta_1|Y_k)$ convergences from 1 to 0 very

fast and keeps convergence when the shiftpoint $k > 250$, and on the other hand, the conditional probability for the hacked case $p(\theta_2|Y_k)$ convergences from 0 to 1 very fast and also keeps convergence, which means the system is now under attack when $k > 250$. Note that the convergence time for $p(\theta_1|Y_k) = 1$ when $k < 250$ is $k = 5$, and the convergence time for $p(\theta_2|Y_k) = 1$ when $k > 250$ is $k = 255$, which shows this estimation algorithm could also work for the second-order system when there is a step and ramp-type intrusion enters the system.

4.5 Analysis of Simulation Results

Considering the four cases mentioned above, the algorithm using a bank of Kalman filters in this thesis can detect the intrusion signal when it enters the system measurement based on the intrusion model considered. For example, when a constant-type signal enters the second-order system mentioned in section 4.3, The conditional probability for the hypothesis 1, which is the unhacked case $p(\theta_1|Y_k)$, convergences to 1 very quickly and keeps convergence until the shiftpoint $k = 250$ when there is no intrusion signal enters the system, note that the convergence time for $p(\theta_1|Y_k) = 1$ is $k = 3$ and the probability for the hacked case $p(\theta_2|Y_k) = 0$. When $k > 250$, the probability of the hacked hypothesis $p(\theta_2|Y_k)$ convergences from 0 to 1 with the convergence time $k = 264$. Note that the shorter convergence time the better it will be, especially for the convergence time when the hypothesis $p(\theta_2|Y_k)$ (hacking happens) convergences from 0 to 1. Since the different signal to noise ratios could influence the

convergence time, it is necessary to do some analysis between the system signal to noise ratio (SNR) and the convergence time. Here, the SNR is given as below

$$\text{SNR} = \frac{CC^T tr(X_\infty)}{W} \tag{4.22}$$

Where

$$X_\infty = AX_\infty A^T + FVF^T$$

From (4.22) it can be found that the system SNR increases by increasing the value of system state noise covariance V or decreasing the value of W. Here, Table 4.1 shows the changes of the convergence time for unhacked case for $p(\theta_1|Y_k)$ to converge to one before the intrusion happens as a function of time the covariance of the system state noise V.

Table 4.1: Changes of the convergence time for the second-order system with constant-type intrusion signal when increasing the SNR from 3 to 30

V	W	SNR	Convergence Time
0.1	1	2.8352	3
0.5	1	14.1762	3
1	1	28.3525	4
2	1	56.7050	5
3	1	85.0575	5
4	1	113.4100	5
5	1	141.7625	5

From Table 4.1, it can be noted that the changes of the system state noise can barely influence the convergence time for the probability of the hypothesis for $p(\theta_1|Y_k)$

converge to one before the intrusion happens. While if the system output is replaced by the intrusion signal, the SNR is presented as below

$$\text{SNR} = \frac{d^2}{W} \tag{4.23}$$

where d is the intrusion signal and in section 4.3 $d = h_k = h_{k+1}$ which is a constant-type intrusion signal and it could be noticed that the value of d could influence the value of SNR from (4.23). Here, Table 4.2 shows the changes of the convergence time of the hypothesis $p(\theta_2|Y_k)$ to one after the intrusion happens as a function of time the SNR from 3 to 30.

Table 4.2: Changes of the convergence time for the second-order system with constant-type intrusion signal when increasing the SNR from 3 to 30

d	W	SNR	Convergence Time
3	1	9	369
5	1	25	278
10	1	100	266
15	1	225	257
20	1	400	254
25	1	625	254
30	1	900	254

From Table 4.2, it can be noted that increasing the SNR of the system can lead to a shorter detection time of intrusion. This might imply that might because the larger SNR could lead to the bank of Kalman filters working faster. However, it is necessary to do some study regarding the relationship between the SNR and the convergence time.

4.6 An Alternative Detection Algorithm for the Intrusion Problem

In this section, an alternative detection method is where the computed throretical mean value of the system measurement is compared to the actual value of the measurement.

Consider the first-order system

$$x_{k+1} = 0.9x_k + v_k \tag{4.24a}$$

$$y_k^1 = 3x_k + w_k \tag{4.24b}$$

where $\mathrm{A} = 0.9$, $\mathrm{B} = 0$, $\mathrm{C} = 3$, $\mathrm{D} = 0$, $\mathrm{F} = 1$, $\mathrm{G} = 1$, and the covariance of the system state noise $V = 0.01$, the covariance of the system measurement noise $W = 0.01$ and both system state noise and system measurement noises are aero mean and Gaussian. Note that this system's eigenvalue is inside the unit circle and the superscript 1 represents the system is currently not under attack. After the shiftppoint $k = 100$, there is a constant-type intrusion signal $h_k = h_{k+1}$ enters the system and replaces the system measurement and in this case, the system measurement becomes

$$y_k^2 = h_k + w_k \tag{4.25}$$

and the system state and measurement could be shown as Fig 4.13

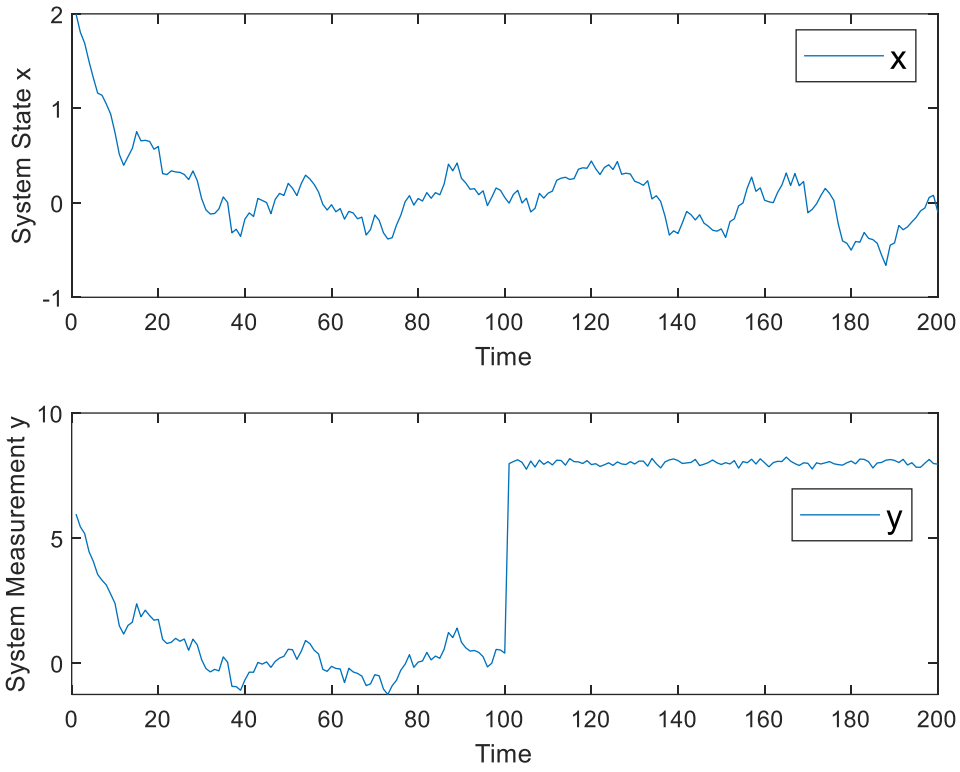

Figure 4.13: The first-order Discrete-Time stochastic system state x_k and measurement y_k with its initial state $x_0 = 2$ and the Constant-Type sensor intrusion $h_k = 2$ happens at shiftpoint $k = 100$.

The sample mean value of the system measurement $\bar{y}_k$ can be approximately found using the sample mean method by setting up the initial state's mean value $\bar{x}_0$. If there is a control signal u_k, then we have

$$\bar{y}_k = CA^k\bar{x}_0 + C\sum_{i=0}^{k-1} A^{k-i-1}Bu_i \tag{4.26}$$

Since there is no control signal in this system, (4.26) could be simplified as below

$$\bar{y}_k = CA^k\bar{x}_0 \tag{4.27}$$

And the sample mean value of the system measurement could be shown as Fig 4.14.

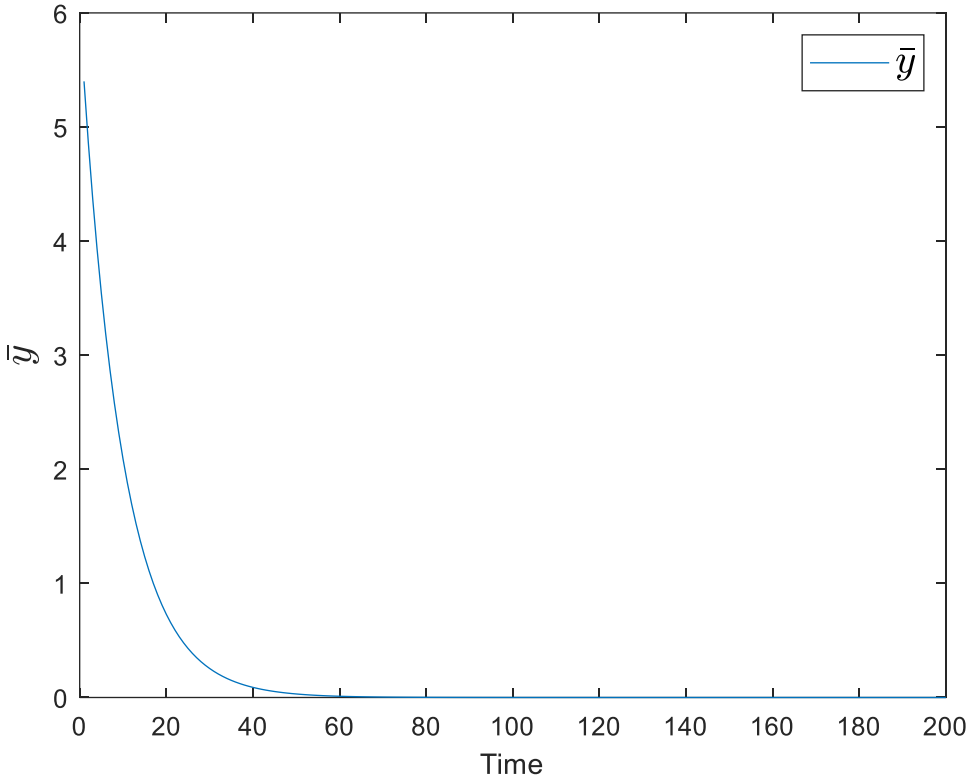

Figure 4.14: The first-order Discrete-Time stochastic system sample mean value of the system measurement $\bar{y}_k$ with its initial mean value of the system state $\bar{x}_0 = 2$ and the Constant-Type sensor intrusion signal $h_k = 2$ happens at shiftpoint $k = 100$.

After calculating the sample mean value of the system measurement $\bar{y}_k$, the sample mean detection algorithm can be expressed as Fig 4.15

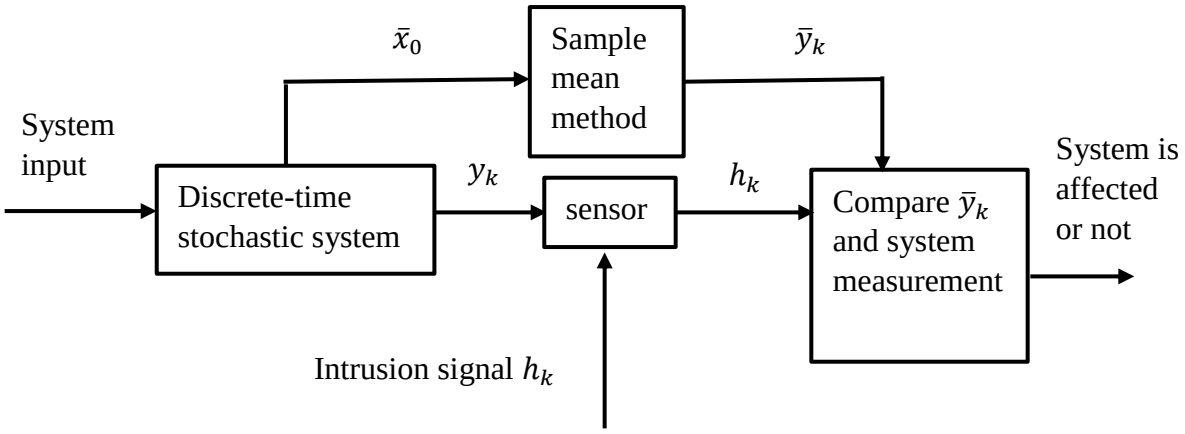

Figure 4.15: Flowchart of the sample mean detection algorithm

As mentioned in Fig 4.15, there is a comparison between the system measurement and the theoretical sample mean value of the system measurement, and it is defined as $\tilde{y}_k = y_k - \bar{y}_k$. Before the intrusion enters the system, the value of $\tilde{y}_k$ should be close to zero because the value of $\bar{y}_k$ is the computed mean value of y_k at each time step *k*. After the intrusion signal h_k enters the system and replaces the system measurement, the comparison becomes $\tilde{y}_k = h_k - \bar{y}_k$, which would not be close to zero because $\bar{y}_k$ is no longer the intrusion signal's mean value. Fig 4.16 shows the value of $\tilde{y}_k$ when the constant-type intrusion signal h_k enters the system at the shiftpiont $k = 100$.

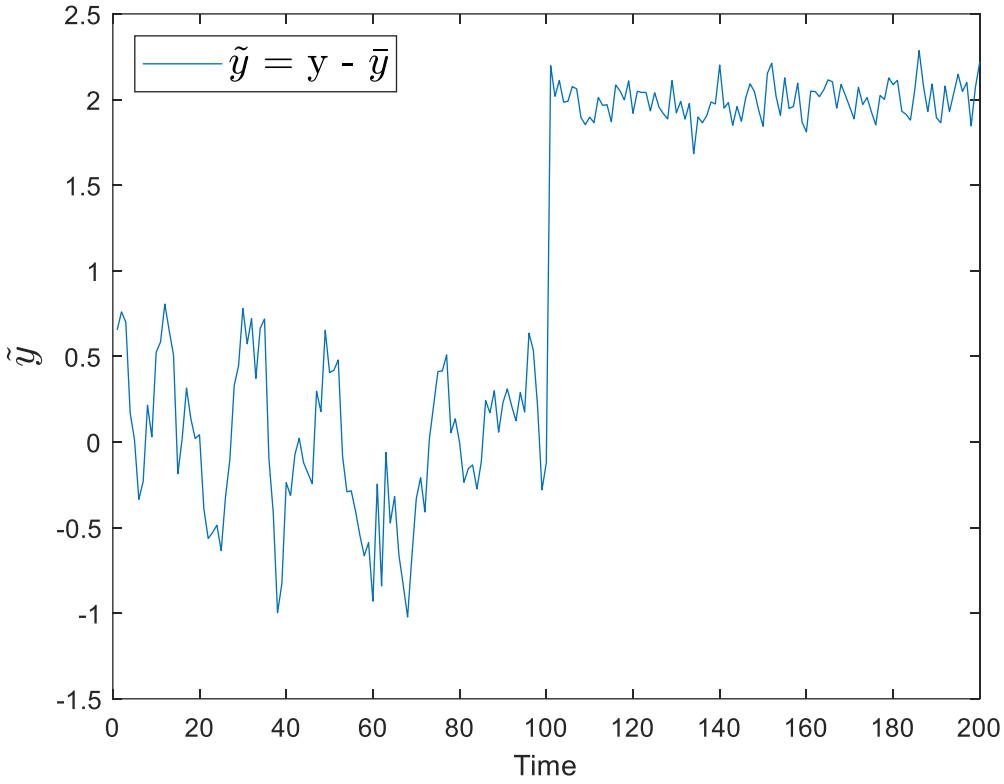

Figure 4.16: The value of $\tilde{y}_k$ when the Constant-Type sensor intrusion $h_k = 2$ happens at shiftpoint $k = 100$.

From Fig 4.16, changes of $\tilde{y}_k$ show that there is a constant-type intrusion signal h_k enters the system at shiftpoint $k = 100$ and changes the system measurement.

Consider the first-order system with a control signal u_k

$$x_{k+1} = 0.9x_k + u_k + v_k \qquad (4.28a)$$

$$y_k^1 = 2x_k + u_k + w_k \qquad (4.28b)$$

where $\mathrm{A} = 0.9,\ \mathrm{B} = 0, \mathrm{C} = 2, \mathrm{D} = 0, \mathrm{F} = 1, \mathrm{G} = 1$, the constant-type control signal $u_k = 1$, and the covariance of the system state noise $V = 0.01$, the covariance of the system measurement noise $W = 0.01$ and both system state noise and system measurement noises are zero mean and Gaussian. Note that this system's eigenvalue is inside the unit circle and the superscript 1 represents the system is currently not under attack. Similarly, after the shiftppoint $k = 100$, there is a constant-type intrusion signal $h_k = h_{k+1}$ enters the system and replaces the system measurement and in this case, the system measurement becomes

$$y_k^2 = h_k + w_k \tag{4.29}$$

and the system state and measurement are shown as Fig 4.17.

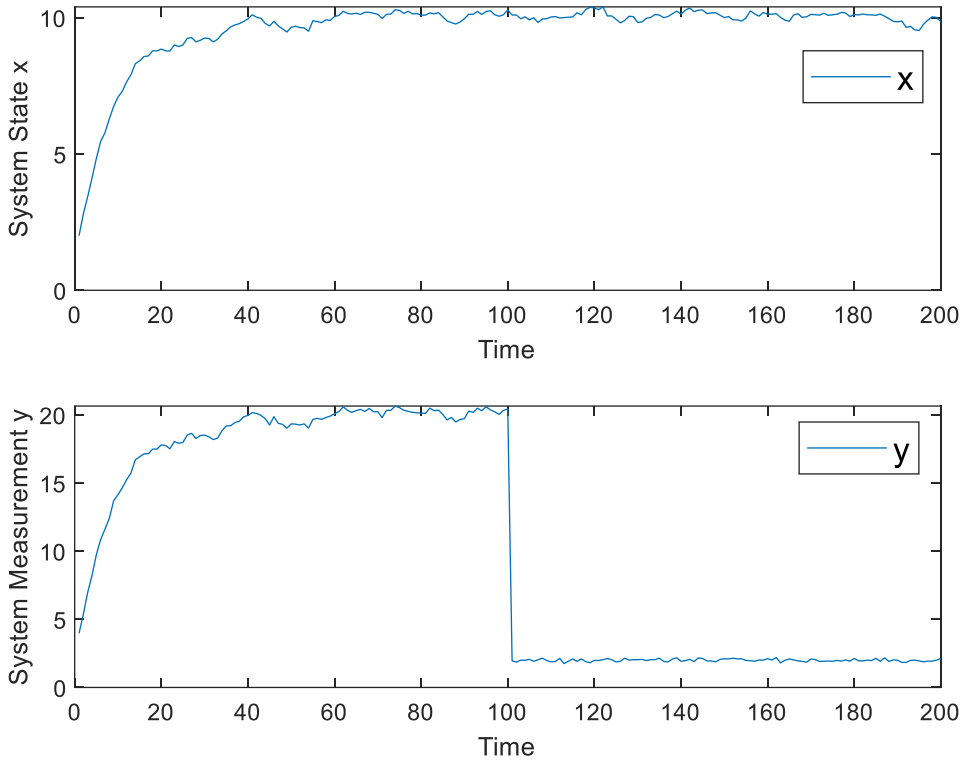

Figure 4.17: The first-order Discrete-Time stochastic system with a constant-type control signal $u_k = 1$, where its state x_k and measurement y_k with its initial state $x_0 = 2$ and the Constant-Type sensor intrusion $h_k = 2$ happens at shiftpoint $k = 100$.

Similarly, the throretical mean value of the system measurement $\bar{y}_k$ can be found using the sample mean method using (4.26) by setting up the system's initial state mean value $\bar{x}_0$ and the control signal u_k. Knowing that u_k is a constant control signal where $u_k = 1$, (4.26) can be simplified as below

$$\bar{y}_k = C\left(A^k \bar{x}_0 + \sum_{i=0}^{k-1} A^{k-i-1} B\right) + 1 \tag{4.30}$$

and the theoretical mean value of the system measurement is shown as Fig 4.18

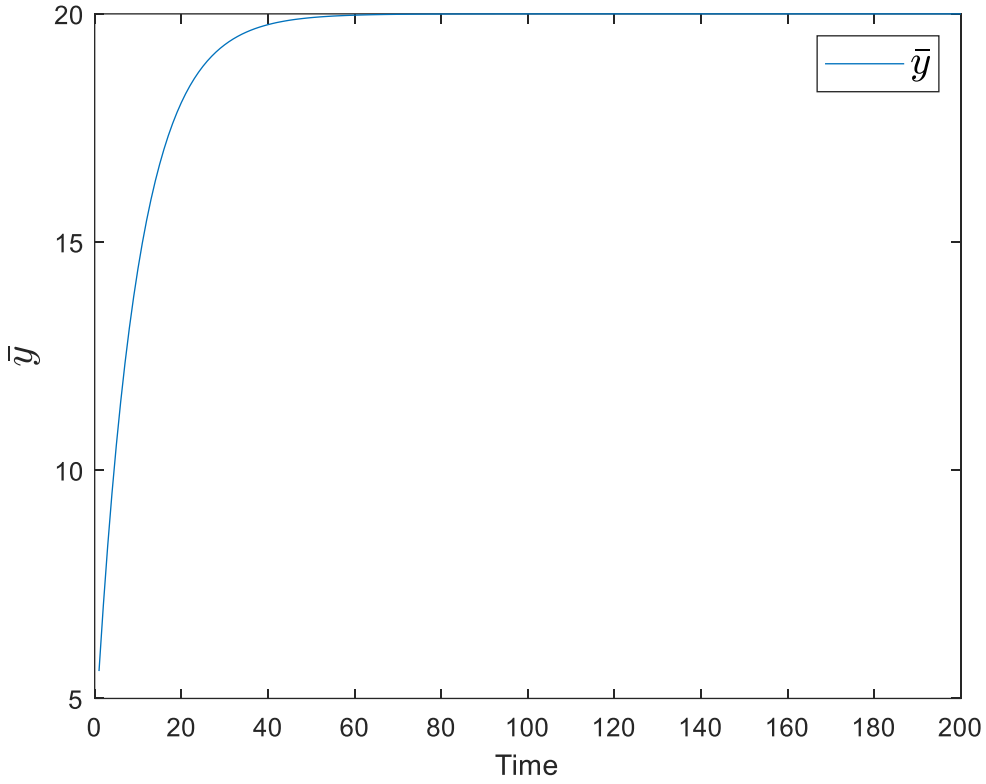

Figure 4.18: The first-order Discrete-Time stochastic system with a constant contorl signal $u_k = 1$, where its theoretical sample mean value of the system measurement $\bar{y}_k$ with its initial mean value of the system state $\bar{x}_0 = 2$ and the Constant-Type sensor intrusion signal $h_k = 2$ happens at shiftpoint $k = 100$.

Similarly, using the algorithm mentioned in Fig 4.15, there is a comparison between the system measurement and the theoretical sample mean value of the system measurement, and it is defined as $\tilde{y}_k = y_k - \bar{y}_k$. Before the intrusion enters the system, the value of $\tilde{y}_k$ should be close to zero because the value of $\bar{y}_k$ is the mean value of y_k at each time step *k*. After the intrusion signal h_k enters the system and replaces the system measurement, the comparision becomes $\tilde{y}_k = h_k - \bar{y}_k$, which would not be close to zero because $\bar{y}_k$ is no longer the intrusion signal's mean value. Fig 4.19 shows the value of $\tilde{y}_k$ when the constant-type intrusion signal h_k enters the system at the shiftpiont $k = 100$.

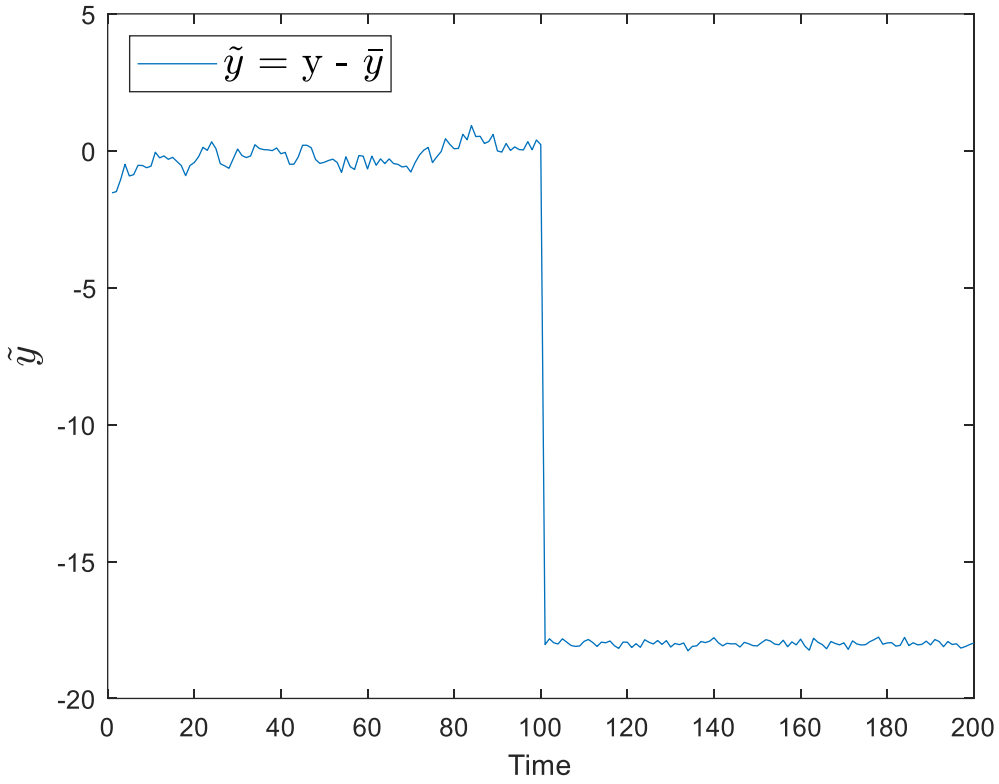

Figure 4.19: The value of $\tilde{y}_k$ when the Constant-Type sensor intrusion $h_k = 2$ happens at shiftpoint $k = 100$.

From Fig 4.19, changes of $\tilde{y}_k$ could show that there is a constant-type intrusion signal h_k enters the system at shiftpoint $k = 100$ and changes the system measurement.

5 SUMMARY, CONCLUSIONS AND FUTURE WORK

5.1 Summary

In this thesis, an estimation algorithm based on a bank of Kalman Filters was designed that is capable of detecting sensor intrusion problem in industrial control systems. It is shown that when a hacker replaces the system measurement by different types of the intrusion signals, the estimation algorithm designed in this thesis can detect the changes of the system measurement by calculating the system state and measurement estimates based on different intrusion possibilities. This was achieved by designing a bank of Kalman filters together with calculating the probabilities of different hypotheses on the system measurements. To set up the bank of Kalman filters, it is necessary to know the system's state and measurement equations. Step and ramp types of intrusion signals either partially or totally replace the measurement signal at a certain time point so that the system measurement does not give information about the system state. Thus, a bank of Kalman filters was implemented, to calculate the system measurement with and without the measurement being replaced by the intrusion signal.

After receiving the system measurement, the probabilities of each hypothesis (unhacked and hacked with step or ramp) can be calculated using the Bayesian Estimation algorithm. As mentioned in chapter 2, the initial probabilities for the hacked and unhacked cases both need to be assumed so that the algorithm could be initialized. After setting the initial probabilities for both assumptions, the probabilities can be

calculated recursively with the given system measurement data. The convergence time for the probabilities converging to one was measured and this was used as the speed criterion of the estimation method. This performance of the algorithm was also tested using different measurement signal to noise ratios.

A simpler estimation method, which is called the sample mean method, was also implemented for the same system when there is a step-type intrusion signal replacing the measurement. Since the system state and measurement mean value can be calculated using this technique, the residual between the actual value of the measurement and the theoretical mean value of the system measurement can be found and this can be used to detect if there's an intrusion signal. When there is no intrusion signal, the residual should be close to zero because the theoretical mean value of the system measurement should be close enough to the actual value of the system measurement. If the measurement is replaced by the intrusion signal, the residual would be relatively larger, and the intrusion signal can be detected.

5.2 Conclusions

The estimation algorithms implemented in this thesis were applied to two different systems, a first-order system and a second-order system, with an additive white noise component in the measurement. Two different type of intrusion signals are considered, step and ramp replacing partially or totally the signal component of the measurement in our attack model.

In our scenarios, first, a first-order system was attacked by the step-type intrusion signal, replacing the measurement by that intrusion signal. The simulations showed that the intrusion was detected and the probability of each hypothesis (hypotheses 1: no intrusion, hypoehtses 2: most of the signal component of intrusion) was calculated as expected. Then, we focused on the step and ramp-type intrusion on the same first-order system and after simulating the attack using step and ramp-type intrusion signal on the first-order system, the result was also positive.

Next, the constant-type and the step and ramp-type intrusion targeting the second-order system were also simulated. Similarly, the intrusion signal replaced modt of the signal component of the system measurement by the intrusion signal at a certain time point. For both cases, the simulations showed the attack can be detected and the probability of each hypothesis was calculated and converged to correct values sufficiently fast.

Furthermore, a short study on how the signal to noise ratio influences the speed performance of the algorithm was also done, and this short study shows the behavior effect of large signal to noise ratios on the speed of detection of the algorithm.

Lastly, a new estimation method, which is named the sample mean method, was developed to detect the sensor intrusions when the intrusion signal replaces the measurement. Using this sample mean technique, the theoretical sample mean value of the system state and measurement can be calculated in time and by forming the residual

between the system actual measurement and theoretical value of its sample mean, the intrusion can be effectively detected.

5.3 Future Work

This work brings out new ideas in the detection of sensor intrusion using estimation theory. The main objective of this work was to achieve a successful detection of certain types of the sensor intrusion targeting the industrial control systems, where two types of the intrusion signal were tested in this work. Other types of intrusion signals can be applied to control systems and the performance of these estimation algorithms can be tested.

The main technique developed in this work using a bank of Kalman filters only applied to first order and second order linear, time-invariant systems with additive white noise, that is of zero mean and Gaussian distributed, but can be applied to nonlinear (e.g. with estended Kalman filters), time-varying systems of larger order affected by noises with various other characteristics.

Furthermore, the extension of this work only used sample mean technique for the first order system with constant-type intrusion and step and ramp-type intrusion and this technique can also be applied to higher order systems.

Also, the relationship between the signal noise to ratio and the speed of the first detection algorithm was only studied briefly. This relationship might be analyzed further

to be able to find optimal signal to noise ratios for the best performance of these intrusion detection estimation algorithms.

REFERENCES

[1] W.Greg, B.Gary (2001), "*An Introduction to the Kalman Filter"*, University of North Carolina, Chapel Hill, NC Available: http://www.cs.unc.edu/~welch [September 21, 2012].

[2] Alvaro A. Cárdenas, Saurabh Amin, ZongSyun Liny, Yu-Lun Huangy, Chi-Yen Huangy and Shankar Sastry, March 22–24, 2011, "Attacks Against Process Control Systems: Risk Assessment, Detection, and Response", ASIACCS '11

[3] Jill Slay and Michael Miller. Lessons learned from the Maroochy water breach. In Critical Infrastructure Protection, volume 253/2007, pages 73-82. Springer Boston, November 2007.

[4] M. S. Ayas and S. M. Djouadi, "Undetectable sensor and actuator attacks for observer based controlled Cyber-Physical Systems," 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, 2016, pp. 1-7.

[5] S.M.Kay, "Fundamentals of Statistical Signal Processing: Estimation Theory," Upper Saddle River, NJ: Prentice Hall, 1993.

[6] M.J. Wenzel, "Polymer-Coated and Polymer-Based Microcantilever Chemical Sensors: Analysis and Sensor Signal Processing," Ph.D. Dissertation, Marquette University, Milwaukee, WI, U.S.A, 2009.

[7] T.Roberto (2005, August 30), "Estimation Theory for Engineers," Available: http://www.ee.uwa.edu.au/~roberto/teach/Estimation_Theory.pdf [Accessed: September 20, 2013].

[8] R. N. Clark, "Instrument Fault Detection," IEEE Transactions on Aerospace and Electronic Systems, vol. AES-14, no. 3, pp. 456-465, May 1978.

[9] Kobayashi T, Simon DL. Application of a Bank of Kalman Filters for Aircraft Engine Fault Diagnostics. ASME. Turbo Expo: Power for Land, Sea, and Air, Volume 1: Turbo Expo 2003

[10] W. Xue, Y. Guo and X. Zhang, "A Bank of Kalman Filters and a Robust Kalman Filter Applied in Fault Diagnosis of Aircraft Engine Sensor/Actuator," *Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007)*, Kumamoto, 2007, pp. 10-10.

[11] D. H. Trinh and H. Chafouk, "Fault detection and isolation using Kalman filter bank for a wind turbine generator," *2011 19th Mediterranean Conference on Control & Automation (MED)*, Corfu, 2011, pp. 144-149.

[12] G. Rigatos, D. Serpanos and N. Zervos, "Detection of Attacks Against Power Grid Sensors Using Kalman Filter and Statistical Decision Making," *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7641-7648, 1 Dec.1, 2017.

[13] M. Rezaee, N. Sadeghzadeh-Nokhodberiz and J. Poshtan, "Kalman filter based sensor fault detection and identification in an electro-pump system," *2017 5th International Conference on Control, Instrumentation, and Automation (ICCIA)*, Shiraz, 2017, pp. 12-17.

[14] Y. Chen, S. Kar and J. M. F. Moura, "Cyber-Physical Attacks With Control Objectives," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1418-1425, May 2018.

[15] E. Yaz, EECE 6340, Class Lecture Notes, Topic: "Stochastic Systems Estimation and Control," Faculty of Electrical and Computer Engineering, Marquette University, Milwaukee, WI, Spring 2018.

[16] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems" Transactions of the ASME - Journal of Basic Engineering, vol. 82, pp. 35-45, 1960.

[17] R.G. Brown, P.Y.C. Hwang, "Introduction to Random Signals and Applied Kalman Filtering," 2nd Ed. New York: John Wiley and Sons, Inc, 1992.

[18] K. Sothivelr, "Analysis of Sensor Signals and Quantification of Analytes Based on Estimation Theory". Department of EECE, Marquette University, 2014, Master's Thesis.

[19] A. R. Strandt, A. P. Strandt, S. C. Schneider and E. E. Yaz, "Stator Resistance Estimation Using Adaptive Estimation via a Bank of Kalman Filters," *2018 Annual American Control Conference (ACC)*, Milwaukee, WI, 2018, pp. 1078-1083.

[20] B. D. O. Anderson and J. B. Moore, Optimal Filtering. Mineola, NY: Dover Publications, Inc., 2005.

[21] T. Dursun, "Kalman Filter and Its Applications in Navigation," M.S. Thesis, California State University, Northridge, CA, December 2012.

[22] A.Michael, "The Importance of Kalman Filtering Methods for Economic Systems" in Annals of Economic and Social Measurement, Volume 3, number 1, 1974, pp.49-64 Available: http://www.nber.org/chapters/c9994.pdf [September 22, 2013].

APPENDIX: MATLAB CODES

The MATLAB codes used to implement the bank of Kalman filters algorithm and the Sample Mean algorithm in this thesis are given in this Appendix.

1 MATLAB Code for the First-Order Discrete-Time system

```
%%
% Author: Jiayi Su
%%
% Description: MATLAB Code for the First-Order Discrete-Time system

%% Cleaning
clear all
close all
clc

%% Systen matrices for the First-Order system
A = 0.9;
B = 0;
C  = 1;
D = 0;
F = 1;
G = 1;

kmax = 200; % Set up the time step kmax

x= zeros(1,kmax); % Create an x vector of length kmax and fill it with 0s
y = zeros(kmax,1); % Create an y vector of width kmax and fill it with 0s

vd = 0.1; % Set up the covariance of the system state noise
wd = 0.05; % Set up the covariance of the system measurement noise

%% Creating the noise for system state and measurement, which are distrubuted as
Gaussian
V=sqrt(vd)*randn(1,kmax);
V1 = mean(V);
V2 = V-V1;

W=sqrt(wd)*randn(1,kmax);
W1 = mean(W);
W2 = W-W1;
```

```
%% Initial value of system state
x(1) = 2;

%% Simulating the system with its initial value for kmax times
for k = 1:kmax

    x(k+1) = A*x(k)+F*V2(k);
    y(k) = C*x(:,k)+G*W2(k);

end

%% Plot of results
figure,
plot(x) % Plot of the system state
xlabel('Time')
ylabel('System State x')
legend('x')

figure,
plot(y) % Plot of the system measurement
xlabel('Time')
ylabel('System Measurement y')
legend('y')
```

2 MATLAB Code for the First-Order system with the Constant-Type intrusion signal enters the system.

```
%%
% Author: Jiayi Su
%%
% Description: MATLAB Code for the First-Order Discrete-Time system with the
%constant-type intrusion signal enters the system
%% Cleaning
clear all
close all
clc
%% Systen matrices for the First-Order system
A = 0.9;
B = 0;
C  = 1;
D = 0;
F = 1;
G = 1;

kmax = 200; % Set up the time step kmax

x1= zeros(1,kmax); % Create an x vector of length kmax and fill it with 0s
y = zeros(kmax,1); % Create an y vector of width kmax and fill it with 0s

vd = 0.1; % Set up the covariance of the system state noise
wd = 0.05; % Set up the covariance of the system measurement noise

%% Creating the noise for system state and measurement, which are distrubuted as
Gaussian
V=sqrt(vd)*randn(1,kmax);
V1 = mean(V);
V2 = V-V1;

W=sqrt(wd)*randn(1,kmax);
W1 = mean(W);
W2 = W-W1;

%% Initial value of system state
x1(1) = 2;


%% Attack model when the constant-type intrusion signal enters the system
A2 = [A 0; 0 1];
B2 = B;
```

```
C2 = [0.05 C];
D2 = D;
F2 = [F 0]';

OBSERVABILITY = obsv(A2,C2) % Checking the system observability

ShiftPoint = 100; % Set up the shiftpoint where the intrusion signal enters the system
%randi([80 120],1,1); % The shiftpoint can be select arbitrary using this function


p1(:,:,1)=100; % Set pup the initial value of the unhacked system error covariance P1
xhat1(:,1)=1; % Set up the initial unhacked estimated state x1_0_hat

p2(:,:,1) = p1(:,:,1)*eye(2); % Set pup the initial value of the hacked system error
covariance P2
xhat2(:,1) = [0,0]'; % Set up the initial hacked estimated state x2_0_hat

t = 1:kmax+1; % Set up the total time step for the system

ip = 0.5;   % Set up the initial probability for the unhacked hypothesis

% Set up the space vector the storing the probability of the two hypothesis
pThetaZk1 = [ip NaN(1,length(t)-1)];
pThetaZk2 = [1-ip NaN(1,length(t)-1)];

%% Bank of Kalman Filter scheme
for k=1:kmax

   if k< ShiftPoint

      x1(k+1) = A*x1(k)+F*V2(k);
      y(k) = C*x1(:,k)+G*W2(k);

   else

      h(k) = 10; % Intrusion signal enters the system

      x2(:,k+1) = A2*[x1(k) h(k)]' + F2*V2(k);
      y(k) = C2*x2(:,k) + G*W2(k);

   end

      % Estimator
      p1(:,:,k+1)=A*p1(:,:,k)*A' -
(A*p1(:,:,k)*C'*C*p1(:,:,k)*A')/(C*p1(:,:,k)*C'+wd)+vd; % Error covariance update
```

```
        Kk1(:,k)=(A*p1(:,:,k)*C')/(C*p1(:,:,k)*C'+G*wd*G'); % Kalman Gain update

        xhat1(:,k+1)=A*xhat1(:,k)+Kk1(:,k)*(y(k)-C*xhat1(:,k)); % State update


        p2(:,:,k+1)=A2*p2(:,:,k)*A2'-
(A2*p2(:,:,k)*C2'*C2*p2(:,:,k)*A2')/(C2*p2(:,:,k)*C2'+wd)+vd;  % Error covariance
update

        Kk2(:,k)=(A2*p2(:,:,k)*C2')/(C2*p2(:,:,k)*C2'+G*wd*G');% Kalman Gain update

        xhat2(:,k+1)=A2*xhat2(:,k)+Kk2(:,k)*(y(k)-C2*xhat2(:,k));% State update

      % Calculating the covariance for each Kalman Filter with its
      % corresponding hypothesis omega in time
      omega_k_1(k) = C * p1(:,:,k+1) * C' + G*wd*G';

      omega_k_2(k) = C2 * p2(:,:,k+1) * C2' + G*wd*G';

      % Calculating the system measurement estimate y_hat and the system
      % innovation term y_tilde in time
      yhat1(k) = C*xhat1(:,k);

      y_tilde1(k)=y(k)-C*xhat1(:,k);

      yhat2(k) = C2*xhat2(:,k);

      y_tilde2(k)=y(k)-C2*xhat2(:,k);

      % Likelihood function of each hypothesis
      pzkTheta1 = (2*pi)^(-1/2)*sqrt(1/det(omega_k_1(k)))...
       *exp(-0.5*y_tilde1(k)'*eye/omega_k_1(k)*y_tilde1(k));

      pzkTheta2 = (2*pi)^(-1/2)*sqrt(1/det(omega_k_2(k)))...
       *exp(-0.5*y_tilde2(k)'*eye/omega_k_2(k)*y_tilde2(k));


   % Weight update equations
   denom = pzkTheta1*pThetaZk1(k) + pzkTheta2*pThetaZk2(k);

   % Conditional probability for each hypothesis
   pThetaZk1(k+1) = pzkTheta1*pThetaZk1(k)/denom;
   pThetaZk2(k+1) = pzkTheta2*pThetaZk2(k)/denom;

end
% Plot of results
```

```
figure,
plot(y) % System ture output
xlabel('Time')
ylabel('System Measurement y')
legend('y')


tt = 1:kmax;
figure,
plot(tt,y_tilde1,tt,y_tilde2) % System innovation terms for both hypothesis
xlabel('Time')
ylabel('y tilde')
legend('y tilde1 = y - yhat1','y tilde2 = y - yhat2')
grid on

figure,
plot(t,pThetaZk1,'b',t,pThetaZk2,'r') % Conditional probability for each hypothesis
xlabel('Time')
ylabel('Probability')
legend('unhacked system','hacked system')
grid on

%% Convergence time for the conditional probability goes to 1
thresh = 0.99;
convergenceIndex = [find(pThetaZk1 > thresh,1);find(pThetaZk2 > thresh,1)];
disp('Convergence time:')
t(convergenceIndex)
```

3 MATLAB Code for First-Order system with the step and ramp-type intrusion signal enters the system.

```
%%
% Author: Jiayi Su
%%
% Description: MATLAB Code for the First-Order Discrete-Time system with Ramp-
%Type intrusion siganl

%% Cleaning
clear all
close all
clc
%% Systen matrices for the First-Order system
A = 0.9;
B = 0;
C  = 1;
D = 0;
F = 1;
G = 1;

kmax = 200; % Set up the time step kmax

x1= zeros(1,kmax); % Create an x vector of length kmax and fill it with 0s
y = zeros(kmax,1); % Create an y vector of width kmax and fill it with 0s

vd = 0.1; % Set up the covariance of the system state noise
wd = 0.05; % Set up the covariance of the system measurement noise

%% Creating the noise for system state and measurement, which are distrubuted as
Gaussian
V=sqrt(vd)*randn(1,kmax);
V1 = mean(V);
V2 = V-V1;

W=sqrt(wd)*randn(1,kmax);
W1 = mean(W);
W2 = W-W1;

%% Initial value of system state
x1(1) = 2;

%% Attack model when the step and ramp-type intrusion signal enters the system
   H = [1 1; 0 1];
   A2 = [A 0 0; 0 H(1,:); 0 H(2,:)];
   B2 = B;
```

```
    C2 = [0.05 1 1];
    D2 = D;
    F2 = [F 0 0]';

    OBSERVABILITY = obsv(A2,C2) % Check the observability for the attack model
    RANK_OBSV_MODEL_2 = rank(OBSERVABILITY)

    ShiftPoint = 100; % Set up the shiftpoint where the intrusion signal enters the system
    %randi([80 120],1,1); % The shiftpoint can be select arbitrary between 80 and 120
using this function

p1(:,:,1)=100; % Set up the initial value of the unhacked system error covariance P1
xhat1(:,1)=1; % Set up the initial unhacked estimated state x1_0_hat

p2(:,:,1) = 7*eye(3); % Set up the initial value of the hacked system error covariance P2
xhat2(:,1) = [0.1,0,0]'; % Set up the initial estimated state x2_0_hat for the attack model

t = 1:kmax+1;

ip = 0.5;   %initial probability for the unhacked hypothesis

% Set up the space vector for storing the probability of the two hypothesis
pThetaZk1 = [ip NaN(1,length(t)-1)];
pThetaZk2 = [1-ip NaN(1,length(t)-1)];

%% Bank of Kalman Filter scheme
for k=1:kmax

        % set up the step and ramp-type intrusion signal
        h(:,1) = [1 0.1]';
        h(:,k+1) = H*h(:,k);

    if k< ShiftPoint

        x1(k+1) = A*x1(k)+F*V2(k);
        y(k) = C*x1(:,k)+G*W2(k);

    else

        % Intrusion signal enters the system
        x2(:,k+1) = A2*[x1(k); h(:,k)] + F2*V2(k);
        y(k) = C2*x2(:,k) + G*W2(k);

    end

        % Estimator
```

```
        p1(:,:,k+1)=A*p1(:,:,k)*A'-
(A*p1(:,:,k)*C'*C*p1(:,:,k)*A')/(C*p1(:,:,k)*C'+wd)+vd; % Error covariance  update

        Kk1(:,k)=(A*p1(:,:,k)*C')/(C*p1(:,:,k)*C'+G*wd*G'); % Kalman Gain update

        xhat1(:,k+1)=A*xhat1(:,k)+Kk1(:,k)*(y(k)-C*xhat1(:,k)); % Estimated state update


        p2(:,:,k+1)=A2*p2(:,:,k)*A2'-
(A2*p2(:,:,k)*C2'*C2*p2(:,:,k)*A2')/(C2*p2(:,:,k)*C2'+wd)+vd;  % Error covariance
update

        Kk2(:,k)=(A2*p2(:,:,k)*C2')/(C2*p2(:,:,k)*C2'+G*wd*G'); % Kalman Gain update

        xhat2(:,k+1)=A2*xhat2(:,k)+Kk2(:,k)*(y(k)-C2*xhat2(:,k)); % Estimated state
update

      % Calculating the covariance for each Kalman Filter with its
      % corresponding hypothesis omega in time
      omega_k_1(k) = C * p1(:,:,k+1) * C' + G*wd*G';

      omega_k_2(k) = C2 * p2(:,:,k+1) * C2' + G*wd*G';

      % Calculating the system measurement estimate y_hat and the system
      % innovation term y_tilde in time
      y_tilde1(k)=y(k)-C*xhat1(:,k);

      y_tilde2(k)=y(k)-C2*xhat2(:,k);

      % Likelihood function of each hypothesis
      pzkTheta1 = (2*pi)^(-1/2)*sqrt(1/det(omega_k_1(k)))...
       *exp(-0.5*y_tilde1(k)'*eye/omega_k_1(k)*y_tilde1(k));

      pzkTheta2 = (2*pi)^(-1/2)*sqrt(1/det(omega_k_2(k)))...
       *exp(-0.5*y_tilde2(k)'*eye/omega_k_2(k)*y_tilde2(k));

   % Weight update equations
   denom = pzkTheta1*pThetaZk1(k) + pzkTheta2*pThetaZk2(k);

   % Conditional probability for each hypothesis
   pThetaZk1(k+1) = pzkTheta1*pThetaZk1(k)/denom;
   pThetaZk2(k+1) = pzkTheta2*pThetaZk2(k)/denom;

end

% Plot of results
```

```
figure,
plot(y) % System ture output
xlabel('Time')
ylabel('System Measurement y')
legend('y')

tt = 1:kmax;
figure,
plot(tt,y_tilde1,tt,y_tilde2) % System innovation terms for both hypothesis
xlabel('Time')
ylabel('y tilde')
legend('y tilde1 = y - yhat1','y tilde2 = y - yhat2')
grid on

figure,
plot(t,pThetaZk1,'b',t,pThetaZk2,'r') % Conditional probability for each hypothesis
xlabel('Time')
ylabel('Probability')
legend('unhacked system','hacked system')
grid on

%% Convergence time for the conditional probability goes to 1
thresh = 0.99;
convergenceIndex = [find(pThetaZk1 > thresh,1);find(pThetaZk2 > thresh,1)];
disp('Convergence time:')
t(convergenceIndex)
```

4 MATLAB Code for the second-order discrete-time system

```
%%
% Author: Jiayi Su
%%
% Description: MATLAB Code for modeling the Second-Order Discrete-Time system

%% Cleaning
clear all
close all
clc
%%
% Second-order Discrete-Time model
A1 = [0 0.9; -1 -1];
F1 = [1 0]';
C1 = [1 1];
G1 = 1;

EIGENVALUE_SYS_1 = eig(A1) % Check the system stability
OBSERVABILITY_SYS_1 = obsv(A1,C1) % check the system observability
RANK_OBSV_SYS_1 = rank(OBSERVABILITY_SYS_1)

kmax = 200;

% Error covariance for the systen state noise and measuremnt noise
vd = 1;
wd = 1;

%% Creating the noise for system state and measurement, which are distrubuted as
Gaussian
V1=sqrt(vd)*randn(1,kmax);
V2 = mean(V1);
V = V1-V2;

W1=sqrt(wd)*randn(1,kmax);
W2 = mean(W1);
W = W1-W2;

x1 = zeros(2,kmax); % Create an x1 vector of length kmax, width 2 and fill it with 0s
y = zeros(1,kmax); % Create an y vector of width kmax and fill it with 0s

% Set up the initial value for the second-order system
x1(:,1) = [2,2]';

%% Simulating the system with its initial value for kmax times
```

```
for k = 1:kmax

    x1(:,k+1) = A1*x1(:,k) + F1*V(k);
    y(k) = C1*x1(:,k) + G1*W(k);

end

%% Plot of results
figure,
plot(x1(1,:)) % Plot of the system first state
legend('x1')
xlabel('Time')
ylabel('System State x_1')

figure,
plot(x1(2,:)) % Plot of the system second state
legend('x2')
xlabel('Time')
ylabel('System State x_2')

figure,
plot(y) % Plot of the system measurement
legend('y')
xlabel('Time')
ylabel('System Measurement y')
```

5 MATLAB Code for the second-order discrete-time system with constant-type intrusion signal

```
%%
% Author: Jiayi Su
%%
% Description: MATLAB Code for the Second-Order Discrete-Time system with
% Constant-Type intrusion signal

%% Cleaning
clear all
close all
clc
%% Systen matrices for the Second-Order system
A1 = [0 0.9; -1 -1];
F1 = [1 0]';
C1 = [1 1];
G1 = 1;

EIGENVALUE_SYS_1 = eig(A1) % Check the system eigenvalue for the system
stability
OBSERVABILITY_SYS_1 = obsv(A1,C1) % Check the system observability
RANK_OBSV_SYS_1 = rank(OBSERVABILITY_SYS_1)

kmax = 500; % Set up the time step kmax

vd = 1; % Set up the covariance of the system state noise
wd = 1; % Set up the covariance of the system measurement noise

%% Creating the noise for system state and measurement, which are distrubuted as
Gaussian
V1=sqrt(vd)*randn(1,kmax);
V2 = mean(V1);
V = V1-V2;

W1=sqrt(wd)*randn(1,kmax);
W2 = mean(W1);
W = W1-W2;

x1 = zeros(2,kmax);
y = zeros(1,kmax);

%% Initial value of system state
x1(:,1) = [2,2]';

%% Attack model when the constant-type intrusion signal enters the system
```

```
A2 = [A1(1,:)  0; A1(2,:)  0; 0 0 1 ];
F2 = [F1' 0]';
C2 = [0 0.1 1];
G2 = G1;

EIGENVALUE_SYS_2 = eig(A2) % Check the eigenvalue for the attack model for the system stability
OBSERVABILITY_ATTACK_MODEL = obsv(A2,C2) % Check the observability for the attack model
RANK_OBSV_ATTACK_MODEL = rank(OBSERVABILITY_ATTACK_MODEL)

p1(:,:,1) = 100*eye(2); % Set up the initial value of the unhacked system error covariance P1
xhat1(:,1) = [0 0]'; % Set up the initial unhacked estimated state x1_0_hat

p2(:,:,1) = 100*eye(3); % Set up the initial value of the hacked system error covariance P2
xhat2(:,1) = [0 0 0]'; % Set up the initial hacked estimated state x2_0_hat

t = 1:kmax+1;

ip = 0.5;   % Set up the initial probability for the unhacked system

% Set up the space vector the storing the probability of the two hypothesis
pThetaZk1 = [ip NaN(1,length(t)-1)];
pThetaZk2 = [1-ip NaN(1,length(t)-1)];

ShiftPoint = 250;% Set up the shiftpoint where the intrusion signal enters the system
%randi([200 300],1,1); % The shiftpoint can be select arbitrary between 200 and 300 using this function

% Bank of Kalman Filter scheme
for k = 1:kmax


   if k< ShiftPoint

      x1(:,k+1) = A1*x1(:,k) + F1 *V(k);
      y(k) = C1*x1(:,k) + G1*W(k);

   else

      h(k) = 20;% the constant-type intrusion signal enters the system
      x2(:,k+1) = A2*[x1(:,k); h(k)] + F2 *V(k);
      y(k) = C2*x2(:,k) + G2*W(k);
```

```
    end

        % Estimator
        p1(:,:,k+1)=A1*p1(:,:,k)*A1'-
(A1*p1(:,:,k)*C1'*C1*p1(:,:,k)*A1')/(C1*p1(:,:,k)*C1'+wd)+vd; % Error covariance
update

        Kk1(:,k)=(A1*p1(:,:,k)*C1')/(C1*p1(:,:,k)*C1'+G1*wd*G1'); % Kalman Gain
update

        xhat1(:,k+1)=A1*xhat1(:,k)+Kk1(:,k)*(y(k)-C1*xhat1(:,k)); % State update


        p2(:,:,k+1)=A2*p2(:,:,k)*A2'-
(A2*p2(:,:,k)*C2'*C2*p2(:,:,k)*A2')/(C2*p2(:,:,k)*C2'+wd)+vd; % Error covariance
update

        Kk2(:,k)=(A2*p2(:,:,k)*C2')/(C2*p2(:,:,k)*C2'+G2*wd*G2'); % Kalman Gain
update

        xhat2(:,k+1)=A2*xhat2(:,k)+Kk2(:,k)*(y(k)-C2*xhat2(:,k)); % State update

      % Calculating the covariance for each Kalman Filter with its
      % corresponding hypothesis omega in time
      omega_k_1(k) = C1 * p1(:,:,k+1) * C1' + G1*wd*G1';

      omega_k_2(k) = C2 * p2(:,:,k+1) * C2' + G2*wd*G2';

      % Calculating the system innovation term in time using system true
      % measurment and the system estimated measurement
      y_tilde1(k)=y(k)-C1*xhat1(:,k);

      y_tilde2(k)=y(k)-C2*xhat2(:,k);

      % Likelihood functions for each hypothesis
      pzkTheta1 = (2*pi)^(-1/2)*sqrt(1/det(omega_k_1(k)))...
       *exp(-0.5*y_tilde1(k)'*eye/omega_k_1(k)*y_tilde1(k));

      pzkTheta2 = (2*pi)^(-1/2)*sqrt(1/det(omega_k_2(k)))...
       *exp(-0.5*y_tilde2(k)'*eye/omega_k_2(k)*y_tilde2(k));
     % Weight update equations
     denom = pzkTheta1*pThetaZk1(k) + pzkTheta2*pThetaZk2(k);

     % Conditional probability for each hypothesis
     pThetaZk1(k+1) = pzkTheta1*pThetaZk1(k)/denom;
     pThetaZk2(k+1) = pzkTheta2*pThetaZk2(k)/denom;
```

```
end

% Calculate the system Signal Noise Ration before the system is intruded
X = dlyap(A1,F1*vd*F1');
SNR = (C1*C1'*trace(X))/wd


% Plot of the results
figure,
plot(y) % System true output
xlabel('Time')
ylabel('System Measurement y')
legend('y')



tt = 1:kmax;
figure,
plot(tt,y_tilde1,tt,y_tilde2) % System innovation terms for each hypothesis
xlabel('Time')
ylabel('y tilde')
legend('y tilde1 = y - yhat1','y tilde2 = y - yhat2')
grid on

figure,
plot(t,pThetaZk1,'b',t,pThetaZk2,'r') % Conditional probability for each hypothesis
xlabel('Time')
ylabel('Probability')
legend('unhacked system','hacked system')
grid on

%% Convergence time for the conditional probability goes to 1
thresh = 0.99;
convergenceIndex = [find(pThetaZk1 > thresh,1);find(pThetaZk2 > thresh,1)];
disp('Convergence time:')
t(convergenceIndex)
```

6 MATLAB Code for the second-order discrete-time system with step and ramp-type intrusion signal

```
%%
% Author: Jiayi Su
%%
% Description: MATLAB Code for the Second-Order Discrete-Time system with
% Step and ramp-type intrusion signal

%% Cleaning
clear all
close all
clc
%% Systen matrices for the Second-Order system
A1 = [0 0.9; -1 -1];
F1 = [1 0]';
C1 = [1 1];
G1 = 1;

EIGENVALUE_SYS_1 = eig(A1) % Check the system eigenvalue for the stability
OBSERVABILITY_SYS_1 = obsv(A1,C1) % Check the system observability
RANK_OBSV_SYS_1 = rank(OBSERVABILITY_SYS_1)

kmax = 500; % Set up the time step kmax

vd = 1; % Set up the covariance of the system state noise
wd = 1; % Set up the covariance of the system measurement noise

%% Creating the noise for system state and measurement, which are distrubuted as
Gaussian
V1=sqrt(vd)*randn(1,kmax);
V2 = mean(V1);
V = V1-V2;

W1=sqrt(wd)*randn(1,kmax);
W2 = mean(W1);
W = W1-W2;

x1 = zeros(2,kmax); % Create an x vector of length kmax, width 2 and fill it with 0s
y = zeros(1,kmax); % Create an y vector of width kmax and fill it with 0s

x1(:,1) = [2,2]'; % Set up the system initial state

%% Attack model when the step and ramp-type intrusion signal enters the system
H = [1 1; 0 1];
A2 = [A1(1,:) 0 0; A1(2,:) 0 0; 0 0 H(1,:); 0 0 H(2,:) ];
```

```
F2 = [F1' 0 0]';
C2 = [0 0.1 1 0];
G2 = G1;

OBSERVABILITY_ATTACK_MODEL = obsv(A2,C2)  % Check the observability for
the attack model
RANK_OBSV_ATTACK_MODEL = rank(OBSERVABILITY_ATTACK_MODEL)


p1(:,:,1) = 100*eye(2); % Set up the initial value of the unhacked system error covariance
P1
xhat1(:,1) = [0 0]'; % Set up the initial unhacked estimated state x1_0_hat

p2(:,:,1) = 100*eye(4); % Set up the initial value of the hacked system error covariance
P2
xhat2(:,1) = [0 0 0 0]'; % Set up the initial estimated state x2_0_hat for the attack model

t = 1:kmax+1;

ip = 0.5;   %initial probability for the unhacked hypothesis

% Set up the space vector for storing the probability of the two hypothesis
pThetaZk1 = [ip NaN(1,length(t)-1)];
pThetaZk2 = [1-ip NaN(1,length(t)-1)];

ShiftPoint = 250;% Set up the shiftpoint where the intrusion signal enters the system
   %randi([200 300],1,1); % The shiftpoint can be select arbitrary between 200 and 300
using this function


%% Bank of Kalman Filter scheme
for k = 1:kmax
      % set up the step and ramp-type intrusion signal
      h(:,1) = [1 0.1]';
      h(:,k+1) = H*h(:,k);

   if k< ShiftPoint

      x1(:,k+1) = A1*x1(:,k) + F1 *V(k);
      y(k) = C1*x1(:,k) + G1*W(k);

   else

      % Intrusion signal enters the system
      x2(:,k+1) = A2*[x1(:,k); h(:,k)] + F2 *V(k);
      y(k) = C2*x2(:,k) + G2*W(k);
```

```
    end

        % Estimator
        p1(:,:,k+1)=A1*p1(:,:,k)*A1'-
(A1*p1(:,:,k)*C1'*C1*p1(:,:,k)*A1')/(C1*p1(:,:,k)*C1'+wd)+vd; % Error covariance
update

        Kk1(:,k)=(A1*p1(:,:,k)*C1')/(C1*p1(:,:,k)*C1'+G1*wd*G1'); % Kalman Gain
update

        xhat1(:,k+1)=A1*xhat1(:,k)+Kk1(:,k)*(y(k)-C1*xhat1(:,k)); % State Estimate
update


        p2(:,:,k+1)=A2*p2(:,:,k)*A2'-
(A2*p2(:,:,k)*C2'*C2*p2(:,:,k)*A2')/(C2*p2(:,:,k)*C2'+wd)+vd; % Error covariance
update

        Kk2(:,k)=(A2*p2(:,:,k)*C2')/(C2*p2(:,:,k)*C2'+G2*wd*G2'); % Kalman Gain
update

        xhat2(:,k+1)=A2*xhat2(:,k)+Kk2(:,k)*(y(k)-C2*xhat2(:,k)); % State Estimate
update

      % Calculating the covariance for each Kalman Filter with its
      % corresponding hypothesis omega in time
      omega_k_1(k) = C1 * p1(:,:,k+1) * C1' + G1*wd*G1';

      omega_k_2(k) = C2 * p2(:,:,k+1) * C2' + G2*wd*G2';

      % Calculating the system measurement estimate y_hat and the system
      % innovation term y_tilde in time
      y_tilde1(k)=y(k)-C1*xhat1(:,k);

      y_tilde2(k)=y(k)-C2*xhat2(:,k);

      % Likelihood function of each hypothesis
      pzkTheta1 = (2*pi)^(-1/2)*sqrt(1/det(omega_k_1(k)))...
       *exp(-0.5*y_tilde1(k)'*eye/omega_k_1(k)*y_tilde1(k));

      pzkTheta2 = (2*pi)^(-1/2)*sqrt(1/det(omega_k_2(k)))...
       *exp(-0.5*y_tilde2(k)'*eye/omega_k_2(k)*y_tilde2(k));

     % Weight update equations
     denom = pzkTheta1*pThetaZk1(k) + pzkTheta2*pThetaZk2(k);
```

```
    % Conditional probability for each hypothesis
    pThetaZk1(k+1) = pzkTheta1*pThetaZk1(k)/denom;
    pThetaZk2(k+1) = pzkTheta2*pThetaZk2(k)/denom;

end

% Plot of results
figure,
plot(y) % System ture output
xlabel('Time')
ylabel('System Measurement y')
legend('y')

tt = 1:kmax;
figure,
plot(tt,y_tilde1,tt,y_tilde2) % System innovation terms for both hypothesis
xlabel('Time')
ylabel('y tilde')
legend('y tilde1 = y - yhat1','y tilde2 = y - yhat2')
grid on

figure,
plot(t,pThetaZk1,'b',t,pThetaZk2,'r') % Conditional probability for each hypothesis
xlabel('Time')
ylabel('Probability')
legend('unhacked system','hacked system')
grid on

%% Convergence time for the conditional probability goes to 1
thresh = 0.99;
convergenceIndex = [find(pThetaZk1 > thresh,1);find(pThetaZk2 > thresh,1)];
disp('Convergence time:')
t(convergenceIndex)
```

7 MATLAB Code for the Sample Mean algorithm

```
%Cleaning
clear all
close all
clc
%%
% Author: Jiayi Su
%%
% Description: MATLAB Code for the First-Order system without control signal with
constant-type intrusion signal using Sample Mean algorithm
%% Systen matrices for the First-Order system
A = 0.9;
B = 0;
C  = 3;
D = 0;
F = 1;
G = 1;

kmax = 200; % Set up the time step kmax

x= zeros(1,kmax);% Create an x vector of length kmax and fill it with 0s
y = zeros(kmax,1);% Create an y vector of width kmax and fill it with 0s

x_mean = NaN(1,kmax);% Create an x_mean vector of length kmax and fill it with 0s
y_mean = NaN(kmax,1);% Create an y_mean vector of length kmax and fill it with 0s

vd = 0.01;% Set up the covariance of the system state noise
wd = 0.01;% Set up the covariance of the system measurement noise

EIGENVALUE = eig(A) % Calculate the system eigenvalues
%% Creating the noise for system state and measurement, which are distrubuted as
Gaussian
V=sqrt(vd)*randn(1,kmax);
V1 = mean(V);
V2 = V-V1;

W=sqrt(wd)*randn(1,kmax);
W1 = mean(W);
W2 = W-W1;


x(1) = 2;% Set up the system initial state

shiftpoint = 100;% Set up the shiftpoint where the intrusion signal enters the system
```

```
%Sasmple Mean scheme
for k = 1:kmax

    x(k+1) = A*x(k)+F*V2(k);
    d = 2; % set up the constant-tyoe intrusion signal
   % calculating the sample mean value of the system state and system measurement
       x_mean(k) = (A^k)*x(1);
       y_mean(k) = C*x_mean(k);

   if k<= shiftpoint

       y(k) = C*x(:,k)+G*W2(k);

       % Comparision between the ture measurement and the sample mean
       % measurement
       y_tilde(k) = y(k) - y_mean(k);

   else
       % intrusion signal enters the system
       y(k) = d + G*W2(k);

       % Comparision between the ture measurement and the sample mean
       % measurement
       y_tilde(k) = y(k) - y_mean(k);

   end
end


% Plot of results
figure,
subplot(2,1,1) % system state
plot(x)
xlabel('Time')
ylabel('System State x')
legend('x')
subplot(2,1,2) % system measurement
plot(y)
xlabel('Time')
ylabel('System Measurement y')
legend('y')


figure, % sample mean value of the system measurement
plot(y_mean)
```

```
xlabel('Time')
ylabel('y mean')
legend('y mean')


figure, % residule between the system ture measurement and the system sample mean
measurement
plot(y_tilde)
xlabel('Time')
ylabel('y tilde')
legend('y tilde = y - y mean')
```

8 MATLAB Code for the Sample Mean algorithm

```
%Cleaning
clear all
close all
clc
%%
% Author: Jiayi Su
%%
% Description: MATLAB Code for the First-Order system with constant control signal
with constant-type intrusion signal using Sample Mean algorithm
%% Systen matrices for the First-Order system
A = 0.9;
B = 1;
C  = 2;
D = 1;
u = 1; % constant control signal u = 1
F = 1;
G = 1;

kmax = 200;% Set up the time step kmax


x= zeros(1,kmax);% Create an x vector of length kmax and fill it with 0s
y = zeros(kmax,1);% Create an y vector of width kmax and fill it with 0s


y_mean = NaN(kmax,1);% Create an y_mean vector of length kmax and fill it with 0s


vd = 0.01;% Set up the covariance of the system state noise
wd = 0.01;% Set up the covariance of the system measurement noise


EIGENVALUE = eig(A)% Calculate the system eigenvalues
%% Creating the noise for system state and measurement, which are distrubuted as
Gaussian
V=sqrt(vd)*randn(1,kmax);
V1 = mean(V);
V2 = V-V1;

W=sqrt(wd)*randn(1,kmax);
W1 = mean(W);
W2 = W-W1;

x(1) = 2;% Set up the system initial state
```

```
shiftpoint = 100;% Set up the shiftpoint where the intrusion signal enters the system

syms i
%Sasmple Mean scheme
for k = 1:kmax

    x(k+1) = A*x(k) + B*u + F*V2(k);
    d = 2;% set up the constant-tyoe intrusion signal
    % calculating the sample mean value of the system state and system measurement
    y_mean(k) = C*(A^k)*x(1) + C*symsum(A^(k-i-1)*B*u, 0,k-1);

   if k<= shiftpoint

      y(k) = C*x(:,k)+G*W2(k);

      % Comparision between the ture measurement and the sample mean
      % measurement
      ytilde(k) = y(k) - y_mean(k);

   else
      % intrusion signal enters the system
      y(k) = d + G*W2(k);
      % Comparision between the ture measurement and the sample mean
      % measurement
      ytilde(k) = y(k) - y_mean(k);

   end
end


% Plot of results
figure,
subplot(2,1,1)% system state
plot(x)
xlabel('Time')
ylabel('System State x')
legend('x')
subplot(2,1,2)% system measurement
plot(y)
xlabel('Time')
ylabel('System Measurement y')
legend('y')


figure,% sample mean value of the system measurement
plot(y_mean)
```

```
xlabel('Time')
ylabel('y mean')
legend('y mean')

figure, % residule between the system ture measurement and the system sample mean measurement
plot(ytilde)
xlabel('Time')
ylabel('y mean')
legend('y tilde = y - y mean')
```