Improving the Detection Rate of Cosine Similarity detector in False Data Injection Attacks in Control Systems

Manchuri Yeswanth

A Thesis Submitted to
Indian Institute of Technology Hyderabad
In Partial Fulfillment of the Requirements for
The Degree of Master of Technology

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Department of Electrical Engineering

June 2018

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

_26/06/18_

(Signature)

(Manchuri Yeswanth)

EE15MTECH11025

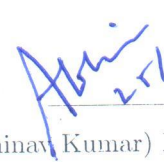(Roll No.)

Approval Sheet

This Thesis entitled Improving the Detection Rate of Cosine Similarity detector in False Data Injection Attacks in Control Systems by Manchuri Yeswanth is approved for the degree of Master of Technology from IIT Hyderabad

-25/06/2018

(Dr. Subrahmanyam Kalyanasundaram) Examiner
Dept. of Computer Science Eng
IITH

27/6/17

(Dr. Abhinav Kumar) Examiner
Dept. of Electrical Eng
IITH

(Dr. Mohammed Zafar Ali Khan) Adviser
Dept. of Electrical Eng
IITH

Acknowledgements

Dedication

To all my loved ones

Abstract

Cyber Physical Systems are more vulnerable to attacks than the conventional systems because of the integrated nature of the cyber as well as physical environment.Replay attacks and False data injection attacks are in particular harmful because of their deceptive nature to traditional detectors. A popular traditional detector is Chi Square Detector which detects based on the statistics of deviations of the residual i.e. difference of observed measurement and estimated measurement. Since the statistics is not changed in the attacks mentioned above, Chi square detector fails to detect these. However, the Cosine detector proposed by [1] also fails in detecting these attacks in control system scenario. So in this work, we will show why the cosine detector fails to detect them and design a method to improve the detection rate of the Cosine Detector.

Contents

Chapter 1

Introduction to False Data Injection Attacks

Over the past few decades, there has been a increasing concern over the reliability and security of the cyber physical systems (CPS). This concern can be attributed to the potential of the cyber physical systems that can change the present energy industry in terms of performance and economy. The Stuxnet attacks and American blackout have demonstrated the vulnerable nature of the SCADA systems (a type of CPS) that created a huge impact on economy.The term cyber-physical systems (CPS) refers to a new generation of systems with integrated computational and physical capabilities that can interact with humans through many new modalities [2] . These CPS have applications in the area of military, aerospace, autonomous systems etc. Example CPS's are the energy systems such as smart grids and the autonomous process control systems that can communicate.

Since the cyber Physical systems physically interact with the real world, securing them is very important. Any possible security lapse would incur in significant economic losses and social losses. The first ever attack identified was the Stuxnet attack on the Iranian Nuclear facilities and has attracted much attention from the research community.Since the CPS have interconnectivity, the inherent network can be prone to attacks. The popular attack schemes that leverage this network to create an attack are the Replay attacks and False Data Injection attacks. In replay attacks, the attacker replays the past data collected over a period of time to the control center to generate false control signals and that will consequently compromise the integrity of the system. On the other hand, False Data injection attacks, generates attack vector that are added to the observation vector. Despite the addition, these attacks remain hidden to the widely used Chi Square Detector. Since the Chi Square Detector is statistical based detector, the attacks were not detected [3].

The first paper that addressed the False data injection in the Control system and wireless network scenario was that of Yilin Mo [4] where he showed deviation in the estimated states of the attacked system compared to the normal system. The paper by Kwon [5] showed the design of the attack vectors for attacks on actuator as well as on the sensor. The paper by Miao [6] addressed improving detection of Chi Square Detector by using the Coding techniques. R.X Niu [7] studied the effect of Kalman filter performance on the attack. Then papers[8],[9] showed how to move a state of system to target state without being detected.

In the previous existing works, only one detector is used primarily i.e. Chi Square Detector. But

there is an increasing demand in research of cheap and reliable detectors that can detect these attacks that remained hidden to Chi Square. In this regard, the Cosine Similarity Detector was proposed by [1] for the smart grid communication system and he showed the improvement in detection of attacks by using the cosine similarity detector over the chi square. A stochastic based detector with a random threshold was also proposed by [10]. Among the detectors studied to detect the false data attacks the easier one was the cosine similarity detection technique. In this paper [1], the author used Kalman filter to predict the observations from the Smart Grid communication systems and then took cosine similarity between the observed measurement and the predicted measurements using Kalman Filter. Since Control systems will have Kalman filter in them to estimate the states of the system, we use cosine similarity detector that can use predicted observations from Kalman filter and detect the attacks.

The main contribution of this work is to prove that Cosine similarity detector fails to detect False data injection attacks designed CPS modelled as control system. We then propose a low cost encoding and decoding scheme to improve the detection rate of both the detectors and give an algorithm to generate the encoding and decoding matrix. We assume that actuators are secured and hence the attacker attacks only on the sensor measurements. We also assume that encoding and decoding matrices are secured and sent by a side channel before encoding and decoding starts. We give an algorithm to generate random encoding and decoding matrices so that attacker wont be able to learn the encoding and decoding schemes.

In this work we focus on improving the performance of Cosine similarity by transforming the observation vector by deterministic and random approaches. The dynamics of the plant is modelled by a linear time invariant (LTI) system equipped with a Kalman filter and Chi Square detector.

The rest is organised as follows. Chapter 2 decribes the system modelling and gives a clear explanation of design of False Data injection attacks and chapter 3 have effect of these attacks on chi square and cosine simialarity detector and in chapter 4 we worked on improving the detection rate of chi square and cosine similarity detector by proposing a simple cost effective method using encoding and decoding scheme and last chapter gives numerical example followed by results.

2

Chapter 2

System Modelling for FDI Attacks in Control Systems

In this chapter we model the physical plant as a Linear Control System. We have the plant equipped with Kalman filter for state estimation , Linear Quadratic Gaussian Controller(LQG) for generation of next input and a detector to detect the attacks.

2.1 Physical Plant Modelling

We assume that the physical plant follow the Linear Time invariant system dynamics that can be described with the following state space model.

$$x_{k+1} = Ax_k + Bu_k + w_k \tag{2.1}$$

where $x_k \in R^n$ denotes the vector of state variables at time k, $u_k \in R^p$ denotes the control input at time k, w_k denotes the process noise and x_o denotes the inital state and w_k, x_o are independent Gaussian Random variables and $x_o \sim \mathcal{N}(0, \Sigma)$ and $w_k \sim \mathcal{N}(0, Q)$

2.2 Kalman Filter

Since we have modelled our physical plant as state space system we have a wireless sensor network to monitor the plant and the outputs of the wireless sensor network is given by

$$y_k = Cx_k + v_k \tag{2.2}$$

where $y_k \in R^m$ denotes the measurements from the sensors and v_k denotes the measurement noise with $v_k \sim \mathcal{N}(0, R)$. These readings are received by the centralised controller through the sensor network. Also note that v_k is independent of x_o and w_k.

Now once these measurements are recieved the Kalman filter tries to estimate the states of the system from the received observations y_k.

The equations are given by

Initialization

$$\hat{x}_{0|-1} = 0, P_{0|-1} = \Sigma \tag{2.3}$$

Predict

$$\hat{x}_{k+1|k} = A\hat{x}_k + Bu_k \tag{2.4}$$

$$P_{k+1|k} = AP_kA^T + Q \tag{2.5}$$

Update

$$K_k = P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1} \tag{2.6}$$

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1}) \tag{2.7}$$

$$P_k = P_{k|k-1} - K_kCP_{k|k-1} \tag{2.8}$$

where $\hat{x}_{0|-1}$ denotes the state initialisation, $P_{0|-1}$ denotes the error covariance initialisation, $\hat{x}_{k+1|k}$ denotes the predicted state estimate, $P_{k+1|k}$ denotes the predicated error covariance estimate, $\hat{x}_k$ denotes the updated state estimate ,P_k denotes the updates error covariance.

Even though Kalman gain varies with time initially it then converges to a constant value when (A, B) are controllable and (A, C) are observable. We assume that kalman filter has reached steady state so that K_k becomes constant.

$$K_k = P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1} \tag{2.9}$$

and $k \to \infty$ so that

$$K_k = K$$

and the state estimate update equation of the Kalman filter are given by

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + K(y_k - C\hat{x}_{k+1|k}) \tag{2.10}$$

now define the residue of the measurement to be z_{k+1}

$$z_{k+1} = y_{k+1} - C\hat{x}_{k+1|k} \tag{2.11}$$

So the above state estimate equation can be written as

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + Kz_k \tag{2.12}$$

Now define the estimation error to be

$$
\begin{aligned}
e_{k+1} &= x_{k+1} - \hat{x}_{k+1} \\
&= Ax_k + Bu_k + w_k - A\hat{x}_k - Bu_k - Kz_{k+1} \\
&= Ax_k - A\hat{x}_k + w_k - K(CAx_k + CBu_k + Cw_k + v_k - CA\hat{x}_k - CBu_k) \\
&= (A - KCA)e_k + (I - KC)w_k - Kv_k
\end{aligned}
\tag{2.13}
$$

4

now using the equation (2.1) and (2.10) we can manipulate e_{k+1} to be

$$e_{k+1} = (A - KCA)e_k + (I - KC)w_k - Kv_k \tag{2.14}$$

2.3   LQG Controller

An LQG controller is used to generate the control inputs to stabilize the system. It minimizes the following objective function

$$J = \lim_{T \to \infty} \min_{u_0 u_1 u_2 .. u_{T-1}} \mathcal{E}[\frac{1}{T} \sum_{k=0}^{k=T-1} (x_k^T W x_k + u_k^T U u_k)] \tag{2.15}$$

where $\mathcal{E}$ denotes the expectation operator and W, U are positive semi definite matrices.We knew that the optimal controller of the above minimization problem takes the following form

$$u_k = -(B^T S B + U)^{-1} B^T S A \hat{x}_k \tag{2.16}$$

where u_k is the optimal control input and S satisfies the following Ricatti equation

$$S = A^T S A + W - A^T S B (B^T S B + U)^{-1} B^T S A \tag{2.17}$$

if we define $L = -(B^T S B + U)^{-1} B^T S A$ then we will have $u_k = L\hat{x}_k$.

2.4   Attack Detector

A χ^2 is often employed to detect the failures in the control systems. It computes the following quantity

$$g_k = \sum_{i=0}^{\mathcal{T}-1} z_k^T \mathcal{P}^{-1} z_k \tag{2.18}$$

where $\mathcal{P}$ denotes the covariance matrix of the residue z_k. g_k is χ^2 distributed with $m\mathcal{T}$ degrees of freedom and $\mathcal{T}$ denotes the window of the detector. The detection is done by comparing g_k with some threshold. In certain, the detector triggers an alarm when

$$g_k > threshold \tag{2.19}$$

The Chi square detector is statistical based detector so when the system under attack have same statistical properties as healthy system i.e. system without having attack then the detector wont be able to detect the attacks.

2.5   False Data Injection Attacks

In this section we will consider the system under the attack. The plant can be modelled as

$$x'_{k+1} = Ax'_k + Bu'_k + w_k \tag{2.20}$$

The measurement vector can be given by

$$y'_{ak} = y'_k + a_k = Cx'_k + v_k + a_k \tag{2.21}$$

where $'$ denotes the attacked system and a_k denotes the attack vector added to the measurement at time instant k. The system block diagram can be given by Fig 2.1 Since we have defined the model
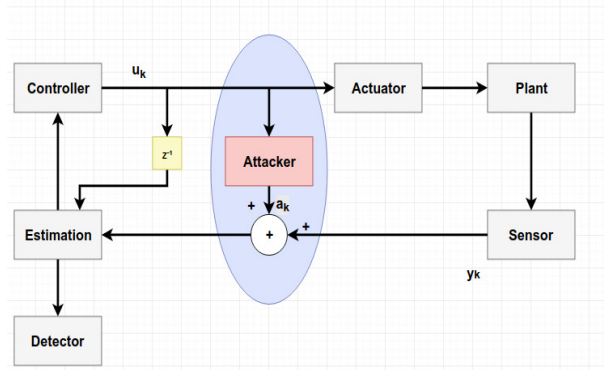
Figure 2.1: System Block Diagram

of the system under attack, we will see basic principles of generation of attack vector. Now let $\hat{x}'_k$ be the estimated state of the compromised system i.e. system under attack. Now let the residual of the compromised system would be given as similar to the (2.11)

$$z'_k = y'_{ak} - C[A\hat{x}'_k + Bu'_k] \tag{2.22}$$

Now define the various error quantites between the normal system and system under attack

$$e_k = x_k - \hat{x}_k \tag{2.23}$$
$$e'_k = x'_k - \hat{x}'_k \tag{2.24}$$
$$\Delta e_k = e'_k - e_k \tag{2.25}$$
$$\Delta z_k = z'_k - z_k \tag{2.26}$$

where Δe_k denotes the difference between the state estimation error of the compromised and the normal system and Δz_k denotes the residual difference between the compromised and the healthy system. Now the dynamics of the above difference vectors is given by

$$\begin{aligned}
e_{k+1} &= x_{k+1} - \hat{x}_{k+1} \\
&= Ax_k + Bu_k - [\hat{x}_{k+1|k} + Kz_{k+1}] \\
&= Ax_k + Bu_k - [A\hat{x}_k + Bu_k + Kz_{k+1}] \\
&= Ae_k - Kz_{k+1}
\end{aligned} \tag{2.27}$$

Thus the recursive equation of the difference of the states in case of unattacked system is given by

$$e_{k+1} = Ae_k + Kz_{k+1} \tag{2.28}$$

6

Now for the system under attacked case the dynamics is given by

$$
\begin{aligned}
e'_{k+1} &= x'_{k+1} - \hat{x}'_{k+1} \\
&= Ax'_k + Bu'_k - [A\hat{x}'_k + Bu'_k + Kz'_{k+1}] \\
&= Ae'_k - Kz'_{k+1}
\end{aligned}
\tag{2.29}
$$

Thus the recursive equation of the difference of the states in case of unattacked system is given by

$$
e'_{k+1} = Ae'_k - Kz'_{k+1}
\tag{2.30}
$$

Now the difference of equation (2.35) and (2.31) gives (2.25)

$$
\begin{aligned}
\Delta e_{k+1} &= e'_{k+1} - e_{k+1} \\
&= A\Delta e_k - K\Delta z_{k+1}
\end{aligned}
\tag{2.31}
$$

So the difference of error in the actual and estimates system in attacked and normal system follow the recursive equation given by

$$
\Delta e_{k+1} = A\Delta e_k - K\Delta z_{k+1}
\tag{2.32}
$$

So from the above equation (2.38) in order to understand it we have to see the dynamics of Δz_{k+1}.

$$
\begin{aligned}
\Delta z_{k+1} &= z'_{k+1} - z_{k+1} \\
&= (y'_{ak+1} - y_{k+1}) + C(\hat{x}_{k+1|k} - \hat{x}'_{k+1|k}) \\
&= a_{k+1} - CA(\hat{x}'_k - \hat{x}_k) + CA(x'_k - x_k) \\
&= CA(e'_k - e_k) + a_{k+1} \\
&= CA\Delta e_k + a_{k+1}
\end{aligned}
\tag{2.33}
$$

Thus the dynamics of the residual of the unattacked and the attacked system is given by

$$
\Delta z_{k+1} = CA\Delta e_k + a_{k+1}
\tag{2.34}
$$

Lemma1:*For a successful false data injection attack, the difference between the residuals of the difference of the normal and the attacked should tend to zero*

$$
\lim_{k \to \infty} \Delta z_{k+1} \to 0
$$

Proof : Since the detector used is chi square detector in order to detect the attack, by definition of chi square detection,

$$
g_{k+1} = \sum_{i=0}^{\mathcal{T}-1} z_{k+1}^T \mathcal{P}^{-1} z_{k+1}
\tag{2.35}
$$

and the above g_{k+1} when greater than a certain threshold detects the attack so when Δz_{k+1} is essentially small i.e. nearly zero then attack wont be detected and that results in a successful false data injection attack. This completes the proof

Since we noticed that $\Delta z_{k+1} \to 0$ the possible design of the attack vector is given by $a_{k+1} = -CA\Delta e_k + \epsilon$ where $\epsilon \leq \mathcal{M}$. So the attacker design the attack vector in this method for a successful

false data injection attack.

Now coming back to the equation (2.38) and substituting equation (2.44) in (2.38) we will get

$$\Delta e_{k+1} = A\Delta e_k - K(CA\Delta e_k + a_{k+1})$$
$$= (A - KCA)\Delta e_k - Ka_{k+1} \tag{2.36}$$

Now the dynamics of the error differences in states is given by

$$\Delta e_{k+1} = (A - KCA)\Delta e_k - Ka_{k+1} \tag{2.37}$$

Since we saw that the error dynamics is given by above equation (2.48) the attacker with the knowledge of the all system matrices tries to compute Δe_k and sends as attack vector at each instant. Now we set conditions on the nature of matrix A in order to drive Δe_k to ∞. By substituting the value of a_{k+1} in equation (2.48) we get

$$\Delta e_{k+1} = (A - KCA)\Delta e_k - K(-CAe_k + \epsilon)$$
$$= A\Delta e_k + \epsilon \tag{2.38}$$

So by observing the equation (2.50), in order to drive Δe_k to infinity,matrix A should have atleast one unstable eigen value and error goes in the direction of the eigen vector of that unstable eigen value.

2.5.1   Contruction of Attack Vector

Since from the previous section we know that the attack vector is of form

$$a_{k+1} = -CA\Delta e_k \tag{2.39}$$

and also we have the difference of the dynamics of the state error equation

$$\Delta e_{k+1} = (A - KCA)\Delta e_k - Ka_{k+1} \tag{2.40}$$

if we expand the above error dynamics equation with the attack vector we will notice that the attack vector takes the form

$$a_{k+p} = -CA^p\Delta e_k \tag{2.41}$$

where p denotes any time instant from k. Now if A has full rank and A is of order n then by Caley hamilton theorem

$$A^n = -c_{n-1}A^{n-1} + ... + c_1 A + (-1)^n det(A)I_n \tag{2.42}$$

By comparing the above equation to controllability grammian, we can compute a_k. If v is the unstable eigen vector of A and it should be reachable state of the equation(2.52). Thus if rank of A is n then we can find first n attack vectors by

$$a_{k=1,2...n} = \mathcal{C}^{-1}v \tag{2.43}$$

8

where $\mathcal{C}$ denotes the controllability grammian.

$$\mathcal{C} = \begin{bmatrix} K & (A - KCA)K & (A - KCA)^2 K \dots (A - KCA)^{n-1} A \end{bmatrix}$$

where n denotes the order of matrix A. Thus we are essentially finding the inputs for the above system that make the error dynamics go to v i.e.unstable vector in n steps. Now for $k > n$ the attack vector is given by using the Caley Hamilton theorem.

$$a_{k+n} = c_n a_n + c_{n-1} a_{n-1} + \dots\dots + c_1 a_1 + c_o$$

Thus the attack vector for the next time step beyond n is linear combination of the n attack vectors generated using equation (2.55).

Inorder to simplify the generation process, since we have generated the attack inputs so as to drive the Δe_k to v in n steps, let the state of Δe_{k+n} at time instant $k + n$ be v. Now we can see that

$$\begin{aligned} a_{k+n+1} &= -CA\Delta e_{k+n} \\ &= -CAv \\ &= -C\lambda v \end{aligned} \tag{2.44}$$

where λ denotes the unstable eigen value of A and v denotes the unstable eigen vector of A. Thus for any time greater than n, the attack vector will have the form

$$\begin{aligned} a_{k+n+p} &= -CA\Delta e_{k+n+p-1} \\ &= -CA^p \Delta e_{k+n} \\ &= -CA^p v \\ &= -C\lambda^p v \end{aligned} \tag{2.45}$$

Now using the above definition of attack vector for any time greater than $k+n$ we will see the design of attack vector for any time greater than $k + n$ instant

$$\begin{aligned} a_{k+n+1} - a_{k+1} &= -CA[\Delta e_{k+n} - \Delta e_k] \\ &= -CA[v - 0] \qquad = -C\lambda v \end{aligned} \tag{2.46}$$

The above is from assumption that $\Delta e_k = 0$ initially and we have driven to v from origin. So by manipulating above equation (2.62) we get,

$$a_{k+n+1} = -C\lambda v + a_{k+1} \tag{2.47}$$

Similarly by above principles,

$$\begin{aligned} a_{k+n+2} - a_{k+2} &= -CA[\Delta e_{k+n+1} - \Delta e_{k+1} \\ &= -CA.A[\Delta e_{k+n} - \Delta e_k] \qquad = -CA^2[v] \end{aligned} \tag{2.48}$$

Thus the resultant attack vector becomes

$$a_{k+n+2} = -CA^2v + a_{k+2} \qquad (2.49)$$

Now generalising the above expression and n denotes the order of matrix A then

$$a_{k+n+i} = -C\lambda^i v + a_{k+i} \qquad (2.50)$$

and if we make $k = 0$ we get

$$a_{n+i} = -C\lambda^i v + a_i \qquad (2.51)$$

*where i=0,1,2...n-1.*So by the above equation we will generate the attack vector for any time greater than n. Summarising the construction of attack vectors, the algorithm for generation of attack vectors is given by

---

Algorithm 1 Construction of Attack Vector Algorithm

Inputs: $A, B, C, K,$ v-unstable eigen vector,n-order of matrix A, k-number of time steps

Outputs: Attack Vectors $a_0, a_1, a_2........a_k$

PROCEDURE

1. Construct Controllability Matrix

$$\mathcal{C} = \begin{bmatrix} K & (A-KCA)K & (A-KCA)^2K \ldots (A-KCA)^{n-1}K \end{bmatrix}$$

2. Get Attack vector $a_0, a_1....a_{n-1}$

$$\bar{a} = \mathcal{C}^{-1}v$$

where $\bar{a}$ denotes the attack vector

3. For any time step $k \geq n$, generate by

$$a_k = -C\lambda^k v + a_{k-n}$$

In this way we can generate all the attack vectors for any number of time instants.

So here we conclude on the generation of the attack vectors for the false data injection attacks that can bypass the traditional chi square detector.

2.6   Cosine Similarity Detector

Cosine similarity detector is popular method of measurement of correlation between two vectors. It calculates the cosine of the angle between the two vectors. If A, B are two vectors then the cosine of the angle is given by

$$cos(\theta) = \frac{\bar{A}.\bar{B}}{||A||.||B||}. \qquad (2.52)$$

The cosine similarity detector is normally used in text recognition and in machine learning and in the field of false data injection attacks, it was first used by [8]. In his work , the cosine similarity was calculated in the measurement vector at time instant y_{k+1} and with the estimated measurement

10

$C\hat{x}_{k+1|k}$.. It is defined as

$$Sim = \frac{y_{k+1}.C\hat{x}_{k+1|k}}{||y_{k+1}||.||C\hat{x}_{k+1|k}||} \qquad (2.53)$$

If there are no attacks, then the Kalman filter estimates perfectly so that the similarity becomes 1 and hence no attack. We can define attack detector (AD) using cosine similarity detector as

$$AD = 1 - Sim$$

The attack and no attack condition are given by $AD = 1$ and 0 respectively.

2.7 False Data Injection Attacks on Actuators

In the previous section, we considered False data injection attacks on the sensor measurements. A question might come regarding the nature of the FDI on the actuator. So in this section we analyse the FDI attack on the actutator and draw some conclusions on it. The system model in which the attacker is doing attacks on the actuator is given by

$$x_{k+1} = Ax_k + Bu_k + Ba_k + w_k$$
$$y_k = Cx_k + v_k \qquad (2.54)$$

where a_k denotes the attack vector that is injected into the actuator and remaining follows similar to the system described in the previous sections. Now lets us assume that the attacker has access to all the actuators so a_k is scaled by B matrix. We have a Kalman filter setup to estimate the states of the system and a Chi Square detector to detect the attacks. Now the residual is given by

$$\begin{aligned} z_{k+1} &= y_{k+1} - C\hat{x}_{k+1|k} \\ &= Cx_{k+1} + v_k - C\hat{x}_{k+1|k} \\ &= CAx_k + CBu_k + CBa_k + Cw_k + v_k - CA\hat{x}_k - CBu_k \\ &= CAe_k + Cw_k + v_k + CBa_k \end{aligned} \qquad (2.55)$$

so from the above equation the dynamics of the residue is given by

$$z_{k+1} = CAe_k + Cw_k + v_k + CBa_k \qquad (2.56)$$

The error in the estimation of the states is given by

$$\begin{aligned} e_{k+1} &= x_{k+1} - \hat{x}_{k+1} \\ &= Ax_k + Bu_k + Ba_k - A\hat{x}_k - Bu_k - K(CAx_k + CBu_k + CBa_k - CA\hat{x}_k - CBu_k) \quad (2.57) \\ &= (A - KCA)e_k + (B - KCB)a_k \end{aligned}$$

Thus the error dynamics in this case is given by

$$e_{k+1} = (A - KCA)e_k + (B - KCB)a_k \qquad (2.58)$$

11

Now in order for a successful false data injection attack the necessary conditions are given by

$$\lim_{k \to \infty} ||z_{k+1}|| \leq \mathcal{M}_\infty$$
$$\lim_{k \to \infty} ||e_{k+1}|| \to \infty \tag{2.59}$$

which means that the error states of the system should be bounded while maintaining residual bounded. This makes a successful attack. So (2.58) can be rewritten as

$$e_{k+1} = Ae_k + Ba_k - Kz_{k+1} \tag{2.60}$$

we have neglected noise in above equation. Multiplying the above equation by C we get,

$$Ce_{k+1} = z_{k+1} - KCz_{k+1} \tag{2.61}$$

Since the pair (A, C) is observable, the quantity that takes the state estimation error to infinity is z_{k+1} i.e. residual . So it is not possible to induce infinite estimation error by keeping the residue to be bounded. Thus we can induce a finite estimation error by doing attack on the actuators.

Thus we conclude that infinie estimation error by keeping the residual bounded is not possible in FDI attacks on the actuator. So we limit ourselves to doing attacks on the sensor measurements.

Chapter 3

Effect of False Data Injection attacks on Chi Square and Cosine Detector

Since the basic principle involved in the generation of the attack vector makes the False data injection attacks remaining stealthy to the Chi square detector, our simulations showed that even cosine similarity detector failed in detecting the false data injection attacks. So we will see how each detector behaves when the attacks are done.

3.1   Chi Square Detector

Let the system is given by

$$x_{k+1} = Ax_k + Bu_k + w_k$$
$$\bar{y}_k = Cx_k + v_k + a_k$$

(3.1)

where $\bar{y}_k$ denotes the measurement in the presence of noise, w_k denotes process noise and v_k denotes measurement noise and a_k denotes the attack vector. Now the estimation of Kalman filter is given by

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + K(\hat{y}_{k+1} - CA\hat{x}_k - CBu_k)$$
$$= A\hat{x}_k + Bu_k + K(\bar{y}_{k+1} - C\hat{x}_{k|k-1})$$

(3.2)

now define

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

.

The residual of the measurement is given by

$$
\begin{aligned}
z_{k+1} &= y_{k+1} - C\hat{x}_{k+1|k} \\
&= Cx_{k+1} + v_{k+1} + a_{k+1} - C(A\hat{x}_k + Bu_k) \\
&= CAe_k + v_{k+1} + a_{k+1}
\end{aligned}
\tag{3.3}
$$

Now we look at the evolution of state estimation error

$$
\begin{aligned}
e_{k+1} &= x_{k+1} - \hat{x}_{k+1} \\
&= (A - KCA)e_k + (I - KC)w_k - Kv_k - Ka_k
\end{aligned}
\tag{3.4}
$$

3.1.1   No Attack Case

Now under no attack $a_{k+1} = 0$ so

$$
\mathcal{E}[z]_{k+1} = CA\mathcal{E}[e_k] + \mathcal{E}[v_{k+1}] = 0
\tag{3.5}
$$

where $\mathcal{E}$ denotes the Expectation operator. In the Chi square detection , the detector is given by

$$
g_k = \sum_{i=0}^{\mathcal{T}-1} z_k^T \mathcal{P}^{-1} z_k
\tag{3.6}
$$

where $\mathcal{P}$ denotes the covariance matrix of the residue z_k. g_k is χ^2 distributed with m degrees of freedom and $\mathcal{T}$ denotes the window of the detector. The detection is done by comparing g_k with some threshold. If there is no attack then

$$
g_k = \sum_{i=0}^{\mathcal{T}-1} (y_k + a_k - C\hat{x}_{k|k-1})^T \mathcal{P}^{-1} (y_k + a_k - C\hat{x}_{k|k-1})
\tag{3.7}
$$

the above equation with $a_k = 0$ becomes

$$
g_k = \sum_{i=0}^{\mathcal{T}-1} (y_k - C\hat{x}_{k|k-1})^T \mathcal{P}^{-1} (y_k - C\hat{x}_{k|k-1})
\tag{3.8}
$$

if kalman filter estimates perfectly, then $y - C\hat{x}_{k|k-1} = v_k$ i.e just gaussian noise. So the g_k becomes sum of squared of gaussian random variables which is chi square and hence it will be less than threshold so giving no attack output.

3.1.2   Attack Case

Under the attack case , the $a_{k+1} = -CAe_k$ from the previous chapter. So by substituting this value in equation (3.12) gives

$$
\begin{aligned}
g_{k+1} &= \sum_{i=0}^{\mathcal{T}-1} (y_{k+1} + a_{k+1} - C\hat{x}_{k+1|k})^T \mathcal{P}^{-1} (y_{k+1} + a_{k+1} - C\hat{x}_{k+1|k}) \\
&= \sum_{i=0}^{\mathcal{T}-1} (CAe_k + v_{k+1} + a_{k+1})^T \mathcal{P}^{-1} (CAe_k + v_{k+1} + a_{k+1})
\end{aligned}
\tag{3.9}
$$

by writing $a_{k+1} = -CAe_k$ we get

$$g_{k+1} = \sum_{i=0}^{\mathcal{T}-1} (v_{k+1})^T \mathcal{P}^{-1}(v_{k+1}) \tag{3.10}$$

Thus the above quantity is again Gaussian hence the false data injection attacks are not detected by the chi square detector. Now by similar principles we will verify the performance of Cosine detector in detecting false data injection attacks.

3.2 Cosine Similarity Detector

We will verify the performance of the cosine similarity detector in normal case and attack case.

3.2.1 No Attack

In the cosine similarity detector, we define a new quantity attack detector given by

$$AD = 1 - \frac{y_{k+1}.C\hat{x}_{k+1|k}}{||y_{k+1}||.||C\hat{x}_{k+1|k}||} \tag{3.11}$$

where $AD = 1$ under attack and 0 under no attack. Now if the Kalman filter estimates correctly then $Cx_{k+1} = Cx_{k+1|k}$ so that the above $AD = 1 - 1 = 0$ hence no attack will be detected.

3.2.2 Under Attack

The cosine detector is given by

$$AD = 1 - \frac{\bar{y}_{k+1}.C\hat{x}_{k+1|k}}{||\bar{y}_{k+1}||.||C\hat{x}_{k+1|k}||} \tag{3.12}$$

and we know that

$$\bar{y}_{k+1} = Cx_{k+1} + a_{k+1} + v_{k+1} \tag{3.13}$$

$$a_{k+1} = -CAe_k \tag{3.14}$$

$$\begin{aligned} AD &= 1 - \Big[\frac{[CAx_k + CBu_k + v_{k+1} - CA(x_k - \hat{x}_k)]^T [CA\hat{x}_k + CBu_k]}{||CAx_k + CBu_k + v_{k+1} - CAx_k + CA\hat{x}_k||||CA\hat{x}_k + CBu_k||} \Big] \\ &= 1 - \Big[\frac{[CBu_k + v_{k+1} + CA\hat{x}_k]^T [CA\hat{x}_k + CBu_k]}{||CAx_k + CBu_k + v_{k+1}||||CA\hat{x}_k + CBu_k||} \Big] \end{aligned} \tag{3.15}$$

Adding Noise term to attack vector we can make $v_{k+1} \to 0$.
Thus

$$AD = 1 - \frac{(CA\hat{x}_k + CBu_k)^T(CA\hat{x}_k + CBu_k)}{||CA\hat{x}_k + CBu_k||||CA\hat{x}_k + CBu_k||} = 1 - 1 = 0 \tag{3.16}$$

Thus,we have seen that the Cosine Similarity detector gives a zero indicating no attack in case of actually an attack. So Cosine similarity detector also fails in detecting the False data injection attacks.

Chapter 4

Improving the Detection of Cosine Detector

In the previous chapter, we have seen that the Cosine similarity detector wont be able to detect the False data injection so here we look into the way to improve the detection rate and formulate a simple method to enhance the detection rate.

4.1 Frame Work

From chapter 2, we seen that the error dynamics in the case of the attack is given by

$$e_{k+1} \qquad = Ae_k - Kz_k \tag{4.1}$$

$$= (A - KCA)e_k - Ka_{k+1} + \epsilon \tag{4.2}$$

where ϵ denotes some bounded term.

$$z_{k+1} = CAe_k + v_{k+1} + a_{k+1} \tag{4.3}$$

and by substituting $a_{k+1} = -CAe_k$ in the above error dynamics and residual equation we get

$$e_{k+m} = A^m e_k + \epsilon_2 \tag{4.4}$$

where ϵ_2 denotes some bounded term. Thus at any time k the error term can be decomposed as

$$e_k = c_k v + \epsilon_3 \tag{4.5}$$

where c_k denotes coefficient and v denotes the unstable eigen vector of matrix A and ϵ_3 is some bounded term.Since we know the difference of the residual of the normal and attacked system is given by

$$\Delta z_{k+1} = CA\Delta e_k + a_{k+1} \tag{4.6}$$

So if we make this difference of residue grow larger then we can detect the attack vector. In the attacked case the attack vector would be $a_{k+1} = -CA\Delta e_k$, So substituting that

$$\Delta z_{k+1} = CA\Delta e_k(I - I) \tag{4.7}$$

in the above equation if we have attack vector such that $(I - I) \neq 0$. The easiest way to detect is to multiply attack vector with a matrix instead of I. This will make detector to detect the attack.

$$\Delta z_{k+1} = CA\Delta e_k(I - D) \tag{4.8}$$

where D denotes some matrix multiplying the attack vector so that the residual is non zero. So by this method we will make the residual to be non zero so that the detector detects the attack. In previous case, $D = I$ so residual was not getting detected and if $D \neq I$ then we can detect the attack.

4.2   Inability of WaterMarking in case of FDI attacks

Replay attacks [11] are also kind of false data injection attacks where the attacker replays the past data collected over a period of time. In this context, we can add a watermarking term to our measurement inorder to detect the attacks. But this comes at expense of loss of optimality. As in absence of attacks, mere adding of watermarking term makes the estimation inoptimal. So dewatermarking procedure needs to be followed. Under no attack,

$$y_{watermarked} = y + \alpha$$
$$y_{dewatermarked} = y_{watermarked} - \alpha \tag{4.9}$$

Under attack

$$y_{watermarked} = y_k + \alpha + a_k$$
$$y_{dewatermarked} = y_{watermarked} - \alpha + a_k \tag{4.10}$$
$$= y_k + a_k$$

still attack vector is present in measurement. and α is any watermarking term and if measurement is not dewatermarked then, it will have effect on state estimation performance. Rather than losing performance of estimator by watermarking we use encoding and decoding method that will detect the attacks without losing any performance. So the easiest way to counter FDI attacks is to encode and decode the measurements

4.3   Encoding and Decoding Method

Here we will use popular encoding and decoding method to make the detector detect the attacks. Instead of transmitting measurement directly we will encode the measurement by a matrix and again

at the receiver so that the attack vector will be scaled by a matrix not I.

$$y_k = Cx_k + v_k$$
$$yenc_k = My_k$$
$$y_{attack} = \bar{y} = yenc_k + a_k \tag{4.11}$$
$$ydecode_k = M^{-1}\bar{y}$$
$$ydecode_k = y_k + M^{-1}a_k$$

This decoded term would be used in the Kalman filter estimation so that the difference of the residue would become not equal to zero i.e.

$$\Delta z_{k+1} = CA\Delta e_k(I - M^{-1}) \tag{4.12}$$

If we denote $M^{-1} = D$ then for $D \neq I$ we can detect the residue.

So we choose a encoding matrix M and a decoding matrix M^{-1} in order to detect the attack. The block diagram for this can be represented as Fig 4.1. So the block diagram is same with encoding
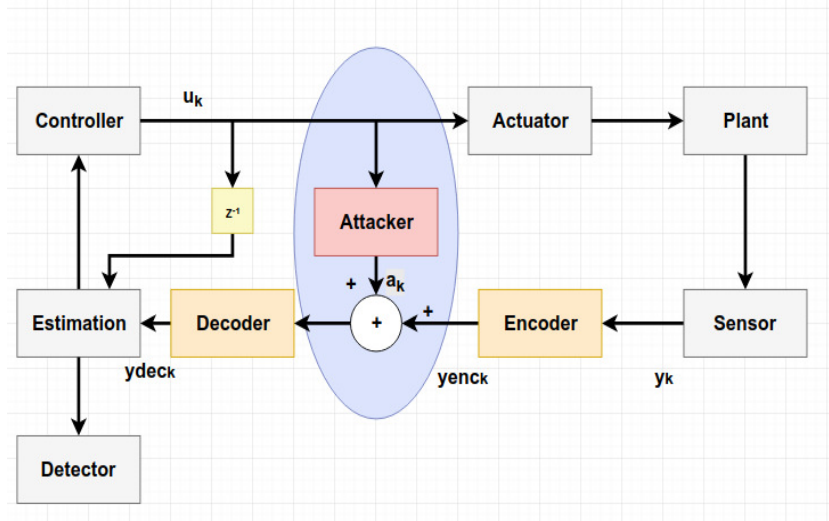

Figure 4.1: Modified System Block Diagram

and decoding components added and it requires minimal effort to place these components in the transmitter stage and the reception stage.

4.4   Choice of Encoding and Decoder Matrix

In the previous sections, we have seen that any matrix D that is non identity matrix will make the residue to be non zero and hence the detection can be improved. So we need to select a appropriate matrix to do encoding and decoding part. There are innumerable choices for these matrices. So we set some conditions in order to choose a appropriate matrix. We will first try to find the decoder matrix and then use its inverse to find the encoder matrix.

- Matrix inverse should exist. This is because if the inverse dont exist then we cant find a

appropriate encoder matrix. This condition gives the easiest choice for selecting D would be a diagnol matrix. However, zeroes are not permitted in the diagnol.

- The matrix should be random in nature. If we use deterministic matrices then the attacker can learn about the matrix and then he can also send encoded attack.

Keeping the above two conditions in mind, we proceed to design the decoder matrix. The analysis of the cosine detector in the presence of encoding and decoding helps us to understand the problem better. The cosine similarity detector in the presence of encoding and decoding is given by

$$AD = 1 - \frac{(y_k + Da_k)^T(C\hat{x_{k|k-1}})}{||(y_k + Da_k)||.||(C\hat{x_{k|k-1}}))||} \tag{4.13}$$

where D denotes the decoder matrix i.e. M^{-1}. In the above equation if the Da_K dominates then we can detect the attacks. From the previous sections, we know that the error dynamics in the estimation of states can be given as

$$\begin{aligned} e_{k+1} &= (A - KCA)e_k - Ka_{k+1} \\ &= c_k v + \epsilon 1_{k+1} \end{aligned} \tag{4.14}$$

where the second equation in the above is obtained by substituting attack vector $a_{k+1} = -CAe_k$ and c_k denotes coefficient of the error dynamics and v denotes the unstable eigen vector and $||\epsilon 1|| < \mathcal{M}$ i.e. some bounded term. and the residue dynamics is given by

$$z_{k+1} = CAe_k + a_{k+1} \tag{4.15}$$

The above equation is with out encoding and decoding and since we are having attack vector multiplied with decoder matrix D the above equation becomes

$$z_{k+1} = CAe_k + Da_{k+1} \tag{4.16}$$

and if we substitute (4.18) in above equation we get

$$z_{k+1} = CAc_k v + \epsilon 2_{k+1} + Da_{k+1} \tag{4.17}$$

where $\epsilon 2 \le \mathcal{M}_2$ i.e. some bounded term. Since the attacker injects the false data injection attacks in the direction of unstable eigen vector of matrix A the above equation can be written as

$$z_{k+1} = c_k \lambda [Cv - DCv] + \epsilon 3_k \tag{4.18}$$

where $\epsilon 3$ denotes another bounded term. This equation is very important equation as it gives the information how the residue can be made to grow by using appropriate D matrix.

The residue will increase in the direction of the ustable eigen vector v and since $\lambda \ge 1$ for that eigen vector v, the elements of D should be in such a way that their magnitude is greater than 1 $d_i > 1$ along the standard basis of the spanned space of measurement matrix along unstable eigen vectors

of A. Now construct a matrix X with columns as given by

$$X = span\left(\begin{bmatrix} Cv_1 & Cv_2 & \ldots Cv_n \end{bmatrix}\right)$$

where $v_1, v_2, v_3 \ldots v_n$ denote the unstable eigen vectors of matrix A. The order of the matrix X would be mxn and $rank(X) \leq n$. Now we will consider two cases depending on rank

4.4.1   rank = n and m=n

When the rank of the matrix X is n , then A have all eigen values greater than 1 in that case the residue grows in the direction of all the eigen vectors.

$$z_{k+1} = \sum_{i=0}^{i=m} c_{ki}\lambda_i[Cv_i - d_iCv_i] + \epsilon_i \tag{4.19}$$

where c_{ki} denotes the coefficient of the error dynamics , d_i denotes the elements of the diagnol matrix, ϵ_i is bounded term , v_i denotes the unstable eigen vectors of matrix A. From the above expression we can see that if the term $\lambda_i d_i$ is maximised then the residue grows and then we can detect the attacks.

4.4.2   rank < n

In this case A will have unstable eigen values less than n so even if the residue grows in direction of other stable eigen values, λ_i^k term goes to zero when $\lambda_i < 1$. So only effect will be because of unstable eigen values. By formulating residue dynamics like equation (4.23) we can see that maximising $\lambda_i d_i$ in the direction of unstable eigen vectors will lead to good detection.So the resulting conditions for the design of the D matrix can be summarized by

- D matrix is a diagnol matrix with no zeros in diagnol entries.

- D matrix should be random in nature

- The decoder matrix designed should maximise $\lambda_i d_i$

From these conditions we move to the next step where we use popular water filling method to design the D matrix.

4.5   Design of Decoder matrix

In this section , we will show how the decoder matrix D will be designed. Since the unstable eigen values in the matrix will not be same we will keep elements d_i in proportion to the λ_i. Here we make use of the random property of the decoder matrix by keeping a constraint on trace of D matrix.

The problem now can be defined by water filling problem as

$$\underset{d_i}{\text{maximize}} \qquad log(1 + \lambda_i d_i)$$

$$\text{subject to} \qquad d_i \geq 0 \ i = 1, \ldots, m.$$

$$\sum_{i=1}^{i=m} d_i = QQ$$

where QQ denotes the trace constraint imposed on matrix D to have random nature and QQ will be random quantity and should be choosen to have $QQ > 0$ to make residue grow to infitnity. However ,QQ shouldnt be choosen too much high and too much low. A value of QQ between 5-30 will work effectively. The solutin of the above problem is given by

$$d_i = (\mu_i - \frac{1}{\lambda_i})^+$$

$$\sum_{i=1}^{i=m} d_i = QQ \tag{4.20}$$

$$\mu \geq 0$$

where $(x)^+ = max(x, \epsilon)$ here we take ϵ to avoid non existence of inverse and μ is obtained by using the (4.24) and substituting in equation (4.25) with out taking $()^+$.

The algorithm can be summarised as

Algorithm 2 Construction of Attack Vector Algorithm

Inputs: $[v_1, v_2, v_3...v_n],[\lambda_1, \lambda_2...\lambda_n],C,\epsilon_1,QQ$

Outputs: DD matrix

PROCEDURE

1. Construct X matrix.

$$X = \begin{bmatrix} Cv_1 & Cv_2 & ... & Cv_n \end{bmatrix}$$

2. Reduce X into standard basis form

$$V = \begin{bmatrix} e_1 & e_2 & e_3 & ... & e_n \end{bmatrix}$$

where $e_i \in \mathbf{R}^n, e_i = 1$ for i^{th} position and zero else where.

3. for i in 1 to n

  if $e_i \in V$

    λ_i unchanged

  else

    $\lambda_i = \epsilon_1$

  end

4. Get μ using equation (33)

5. Get d_i using $d_i = (\mu - \frac{1}{\lambda_i})^+$

The numerical example along with the results is explained in the next chapter.

Chapter 5

Numerical Example and Simulation Results

In this we will show with a numerical example in the improvement of the performance of both the cosine and chi square detector. We use the same example of [3] and do the attack on the position sensor $\gamma = diag(0, 1)$ Consider a vehicle moving along the x- axis. The state space gives the position x and velocity $\dot{x}$ of the vehicle. A control input will be sent to the actuator to control the speed of the vehicle. So the resultant state space equation is given by

$$\dot{x}_{k+1} = \dot{x}_k + u_k + w_{k,1} \tag{5.1}$$

$$x_{k+1} = x_k + (\dot{x}_{k+1} + (\dot{x}_k)/2 + w_{k,2} \tag{5.2}$$

$$y_k = X_k + v_k \tag{5.3}$$

For the above system the set of matrices are given by

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

$$C = I_2, D = 0 \tag{5.4}$$

We further impose following parameters on the system

$$Q = R = W = I_2 \tag{5.5}$$

The steady state Kalman gain and LQG control gain are obtained by

$$K = \begin{bmatrix} 0.5939 & 0.0793 \\ 0.0793 & 0.6944 \end{bmatrix}, L = \begin{bmatrix} -1.0285 & -0.4345 \end{bmatrix}$$

Using the same principles as described in the [8] we will first generate the attack vectors and cross check with the [8] for state deviation. Using the algorithm explained in the chapter 2 we will generate attack vectors and the attack vectors are validated by the following result. Hence with the above graph, we can say that the attack vector that we have generated increases the states of the position
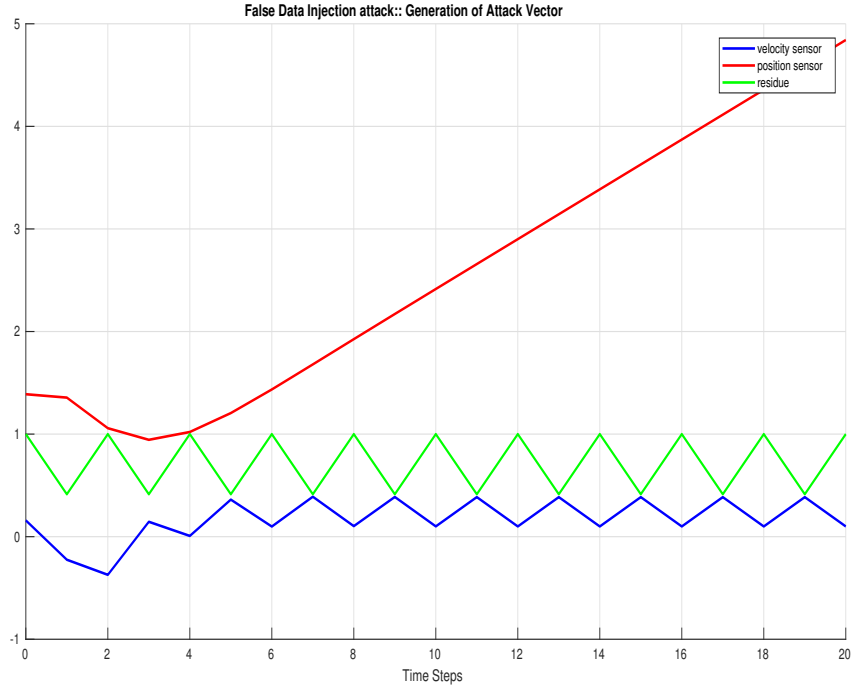
Figure 5.1: False Data injection attack vector generation

and no change in velocity and also the residue is bounded so that the detector wont detect an attack.

5.1 Design of Decoder Matrix

With the method explained in the previous chapter, we find that the unstable eigen value of matrix A is 1 and the unstable eigen vector is $[01]'$. Now the X matrix is given by

$$X = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Since the X matrix is in standard basis form no need to reduce it to basis form and we can see that residue grows along direction of basis of X i.e. $\begin{bmatrix} 0 & 1 \end{bmatrix}^T$. Intuitionally we should have a large value in D matrix along this eigen vector direction and less values in other directions.

Let us choose $QQ = 10$ and apply the Algorithm 2. We get $\lambda_1 = \epsilon_1 = 0.1$ say and $\lambda_2 = 1$ unchanged as $e_2 \in X$. Now calculating μ by using condition 4.18 we get $\mu = 10.5$. So the resultant d_i are obtained from (4.18). The solution given by water filling problem by the above values is

$$D = \begin{bmatrix} 0.5 & 0 \\ 0 & 9.5 \end{bmatrix}$$

23

From above , we can see that D matrix have its element in unstable eigen vector direction so when this multiplied with unstable eigen value , they grow the residue and hence the detector would be able to detect the attack.

Since we have diagnol matrix , the encoder matrix M will be inverse of the decoder matrix D obtained above.

$$M = D^{-1}$$

So in this way we design the encoder and decoder matrix.

5.2 Results

5.2.1 Performance of Chi Square and Cosine detector without Encoding and Decoding

In this section we see the simulation results of the chi square and cosine detector without encoding and decoding.We have conducted the simulation on Matlab R2017b and by changing the variance of the measurement noise. We will investigate the ROC curves. We can see from the Figure 5.2,
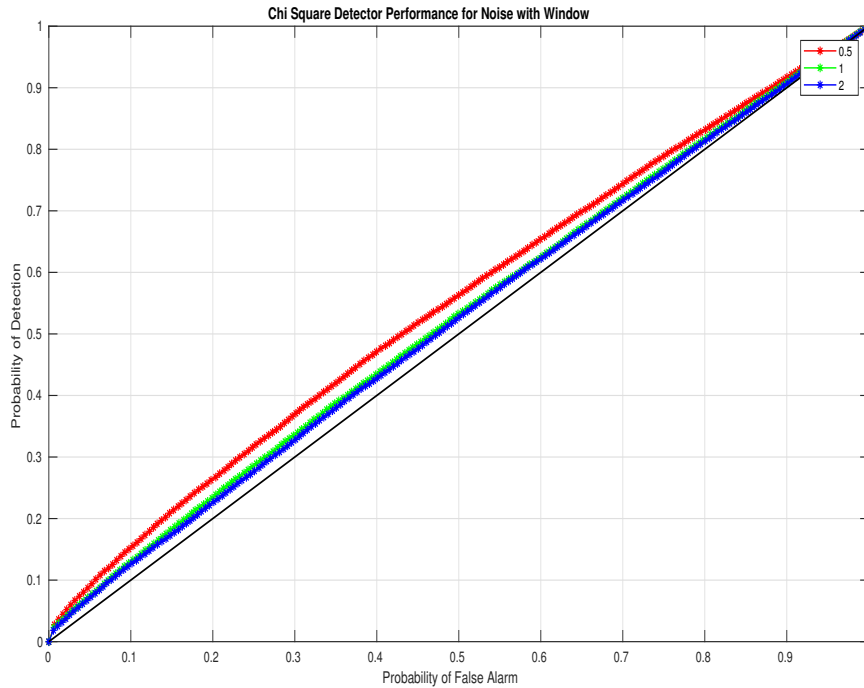

Figure 5.2: Chi Square Detector w/o Encoding and Decoding

the ROC curves are nearly along $y = x$ that shows the inability of the Chi Square detector. In the Figure 5.3 , the ROC points lie on the extreme diagnol regions because the Cosine detector is a "Always No Attack" detector in case of False Data Injection attacks.
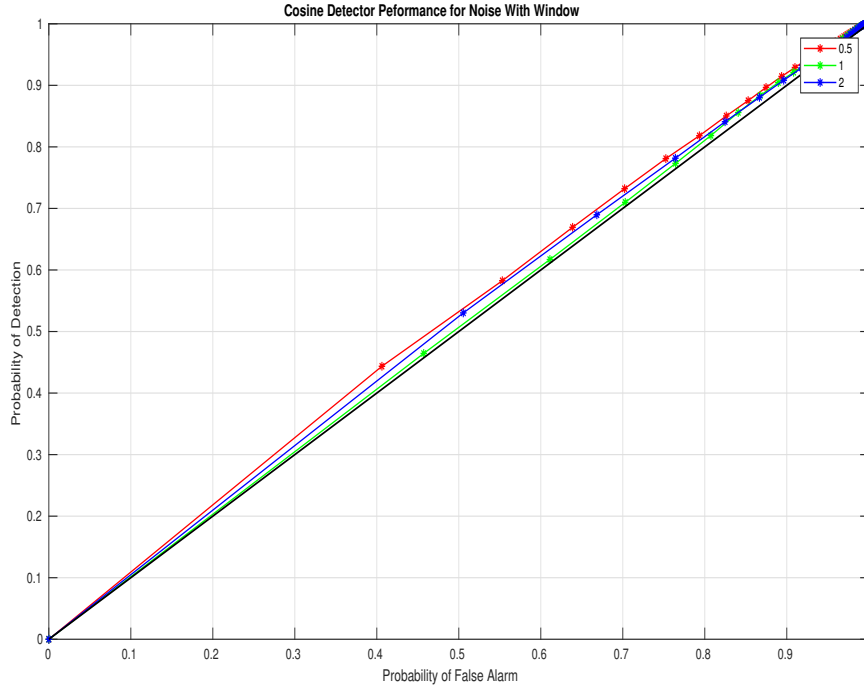
Figure 5.3: Cosine Detector w/o Encoding and Decoding

5.2.2 Explanation of ROC Curves

Chi Square Detector

In the case of Chi Square Detector, under attack scenario the equations of the dynamics of the state estimation error and residue are given by

$$e_{k+1} = (A - KCA)e_k - Ka_{k+1}$$
$$z_{k+1} = CAe_k + a_{k+1}$$

(5.6)

Since we use Chi Square detector, the mean of the residual is given by

$$\mathcal{E}(z_{k+1}) = CA\mathcal{E}(e_k) + (-CA\mathcal{E}(e_k)) \quad = 0$$

(5.7)

where $\mathcal{E}$ denotes the Expectation operator and we have substituted the attack design condition in place of a_{k+1}.

From chapter 2 we have seen that under no attack, the mean is zero and under attack the mean of the residual is still zero. So, the conditional probability density functions under attack and under without attack, overlap on each other so just by varying threshold we will have ROC curves along the $y = x$ line which means probability of detection equal to probability of false alarm $P_f = P_D$.

Cosine Similarity Detector

Now coming to the Cosine similarity detector by similar analysis as in the above section, the mean of the cosine similarity measure is given by

$$
\begin{aligned}
\mathcal{E}(AD) &= \mathcal{E}(1 - \frac{(y_{k+1} + a_{k+1})'(C\hat{x}_{k+1|k})}{||(y_{k+1} + a_{k+1})||.||(C\hat{x}_{k+1|k}||}) \\
&= 1 - \mathcal{E}(\frac{(y_{k+1} + a_{k+1})'(C\hat{x}_{k+1|k})}{||(y_{k+1} + a_{k+1})||.||(C\hat{x}_{k+1|k}||}) \\
&= 1 - (\approx 1) \\
&\approx 0
\end{aligned}
\tag{5.8}
$$

where $\mathcal{E}$ denotes the expectation operator and we get expectation of similarity nearly equal to zero by substituting $a_{k+1} = -CAe_k$.

5.2.3 Performance of Chi Square and Cosine Detector with Encoding and Decoding

The ROC curves shown in the previous subsection implied that the Chi square and Cosine detectors were not able to recognise the attacks because of the nature in which the false data injection attacks are designed.
Here we will show the results of both the detectors in the presence of encoding and decoding.
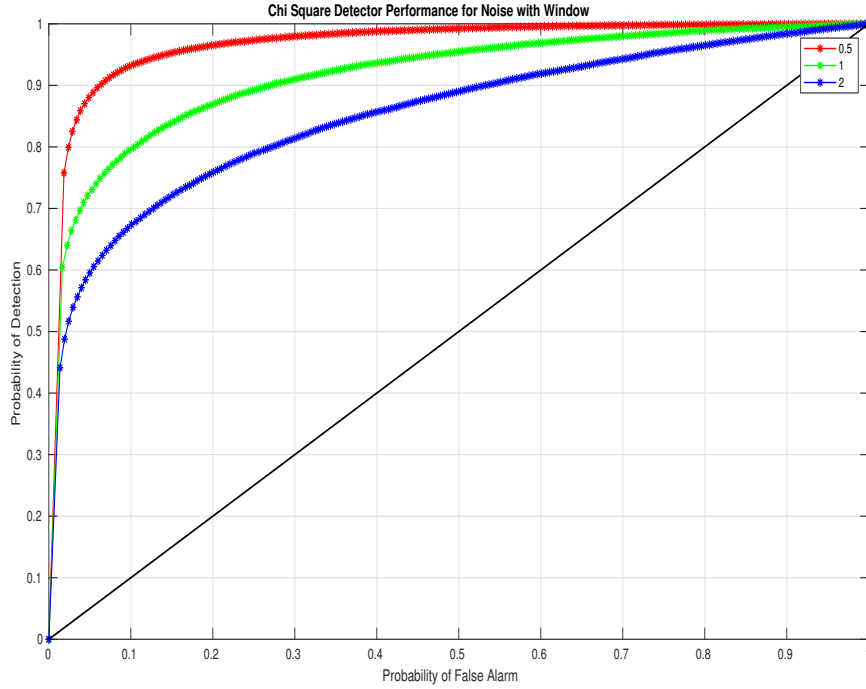

Figure 5.4: Chi Square Detector with Encoding and Decoding

The legend of these curves indicates the measurement noise variance. As the measurement noise
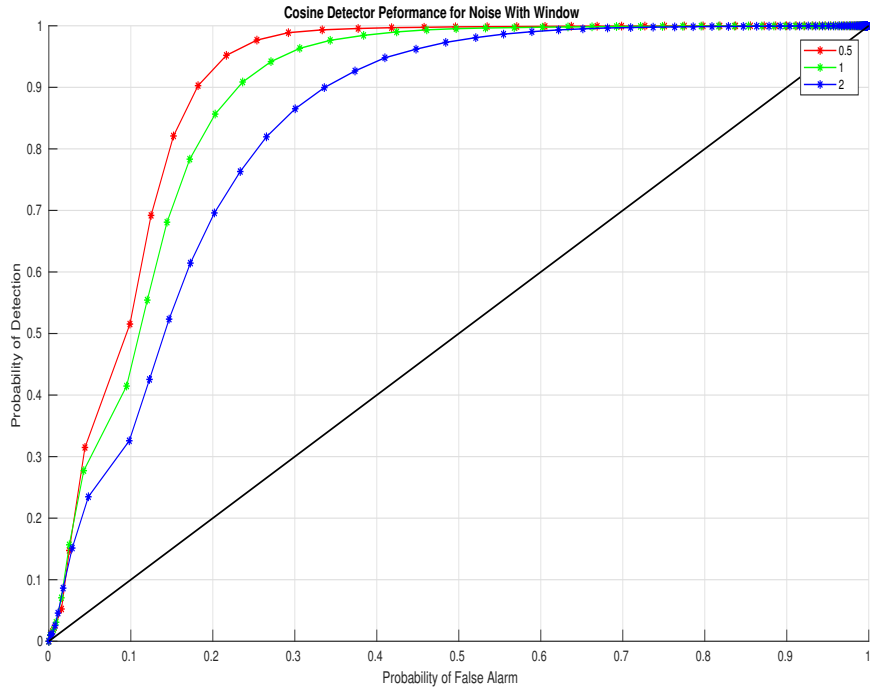
Figure 5.5: Cosine Detector with Encoding and Decoding

increases, the received observation becomes more noisy and hence detector performance decreases. So we have improved the performance capability of the Chi square and cosine similarity detector using the encoding and decoding methods.

Chapter 6

Conclusion

The work shown in the previous chapters clearly explains the design of false data injection attack vectors and gives detailed explanation of behaviour of chi square and cosine similarity detector in the presence of the attacks. Then we have shown the proof why the cosine similarity detector fails to detect the false data injection attacks.We analysed the behaviour of detectors using ROC curves. Then we have shown a simple encoding and decoding method to improve the performance of both the detectors. This method is cost effective and does not require any complex calculations so it can be implementable in networks to detect the false data injection attacks.

References

[1] D. B. Rawat and C. Bajracharya. Detection of false data injection attacks in smart grid communication systems. *IEEE Signal Processing Letters* 22, (2015) 1652–1656.

[2] R. Baheti and H. Gill. Cyber-physical systems. *The impact of control technology* 12, (2011) 161–166.

[3] Y. Mo, E. Garone, A. Casavola, and B. Sinopoli. False data injection attacks against state estimation in wireless sensor networks. 2010.

[4] Y. Mo and B. Sinopoli. False data injection attacks in control systems. In Preprints of the 1st Workshop on Secure Control Systems. 2010 .

[5] C. Kwon, W. Liu, and I. Hwang. Security analysis for cyber-physical systems against stealthy deception attacks. In American Control Conference (ACC), 2013. IEEE, 2013 3344–3349.

[6] F. Miao, Q. Zhu, M. Pajic, and G. J. Pappas. Coding sensor outputs for injection attacks detection. In Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on. IEEE, 2014 5776–5781.

[7] R. Niu and L. Huie. System state estimation in the presence of false information injection. In Statistical Signal Processing Workshop (SSP), 2012 IEEE. IEEE, 2012 385–388.

[8] P.-Y. Chen, S. Yang, J. A. McCann, J. Lin, and X. Yang. Detection of false data injection attacks in smart-grid systems. *IEEE Communications Magazine* 53, (2015) 206–213.

[9] Y. Chen, S. Kar, and J. M. Moura. Cyber physical attacks constrained by control objectives. In American Control Conference (ACC), 2016. IEEE, 2016 1185–1190.

[10] Y. Li and T. Chen. Stochastic Detector against linear deception attacks on remote state estimation. In Decision and Control (CDC), 2016 IEEE 55th Conference on. IEEE, 2016 6291–6296.

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