# Kalman Filter Based Secure State Estimation and Individual Attacked Sensor Detection in Cyber-Physical Systems[1]

Presented by: M. H. Basiri and J. G. Thistle and J. W. Simpson-Porco and S. Fischmeister at the 2019 American Control Conference (ACC)

Jonas Wagner

MECH 6325 Reaserch Seminar
Fall 2020

# Outline

# Outline

# Paper Information

**Title:** Kalman Filter Based Secure State Estimation and Individual Attacked Sensor Detection in Cyber-Physical Systems

**Authors:** M. H. Basiri and J. G. Thistle and J. W. Simpson-Porco and S. Fischmeister

**Conference:** 2019 American Control Conference (ACC)

**Main Contributions:** Development and simulation of two attack detection and secure state estimation algorithms - Rolling Window Detector (RWD) and Novel Residual Detector (NRD)

# Security of Cyber-Physical Systems

- Cyber-Physical Systems (CPS) allow for large scale wide spread control systems that can take advantages of wireless data communication
- CPSs are unfortunately vulnerable to malicious attacks
- Attacks on a subset of sensors may attempt to disrupt the performance of such system
- Detection of such attacks is important and it is difficult to distinguish between noise and attacks
- Many detection methods actually use Kalmen filter estimation and perform hypothetical testing base the residual vector to detect changes above the standard noise threshold

# Outline

# System and Attack Definitions

**Original System**

$$\mathcal{P}: \begin{cases} x_{k+1} = Ax_k + Bu_k + \nu_k, \\ y_k = Cx_k + w_k \end{cases} \quad (1)$$

**States, Inputs, and Outputs:**
$x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, and $y \in \mathbb{R}^q$

**Noise:** $\nu \sim \mathcal{N}(0,\ Q)$ and $w \sim \mathcal{N}(0, R)$

**Initial State:** $x_0 \sim \mathcal{N}(0, \Sigma)$

**Attacked System**

$$\mathcal{P}_a: \begin{cases} x^a_{k+1} = Ax^a_k + Bu_k + \nu_k, \\ y^a_k = Cx^a_k + w_k + Da_k \end{cases} \quad (2)$$

**States, Inputs, and Outputs:** $x^a_k$, $u_k$, and $y^a_k$

**Noise:** $\nu \sim \mathcal{N}(0,\ Q)$ and $w \sim \mathcal{N}(0, R)$

**Attacks:** $a_k$ is manipulated
$D$ s.t. $D_{jj} = 1$ if $j^{th}$ sensor is under attack, otherwise $D_{ij} = 0\ \forall i, j$

# Outline

# Kalman Filter Definition

UT DALLAS

**Measurement Update**

$$K_k = P_{k|k-1}C^T(CP_{k|k-1}C^T + R)^{-1} \quad (3)$$
$$P_{k|k} = P_{k|k-1} - K_k C P_{k|k-1} \quad (4)$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - C\hat{x}_{k|k-1}) \quad (5)$$

**Equivalent Equations:**

$$K_k = P_k^- H^T(HP_k^- H^T + R)^{-1}$$
$$P_k^+ = (I - K_k H_k)P_k^-$$
$$\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - H\hat{x}_k^-)$$

**Time Update**

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} + Bu_k \quad (6)$$
$$P_{k+1|k} = AP_{k|k}A^T + Q \quad (7)$$

$$\hat{x}_k^- = F\hat{x}_{k-1}^+ + Gu_{k-1}$$
$$P_k^- = FP_{k-1}^+ F^T + Q$$

**Steady-State:** $P =^{\triangle} \lim_{k\to\infty} P_{k|k-1}$ $\quad K =^{\triangle} \lim_{k\to\infty} K_k = PC^T(CPC^T + R)^-$

# Conventional $\chi^2$-Detector

Widely used in fault-detection and also applicable to CPS security.

**Residual Vector:**

$$r_k = y_k - \hat{y}_k = y_k - C\hat{x}_k \tag{8}$$

**Residual Covariance:**

$$\Sigma_{r,k} = Ck|kC^T + R \tag{9}$$

**Residual Power:**

$$g_k = r_k^T \Sigma_{r,k}^{-1} r_k \tag{10}$$

**Detector Statistical Test:**
A threshold determined using the desired confidence and degrees of freedom from the $chi^2$-distribution. An alarm is then triggered if:

$$g_k > threshold \tag{11}$$

# Outline

# Rolling Window Detector (RWD)

This detector works by comparing the measured residual cumulative sum $\bar{P}_{k|k}$ over a set period of time $T$ with the error covariance predicted by the Kalman filter, $P_{k|k}$.

**Test Definitions:**

$$\hat{\Sigma}_k = \frac{1}{T} \sum_{k=k_0}^{k_0+T-1} (y_k - \hat{y}_k)(y_k - \hat{y}_k)^T \quad (12)$$

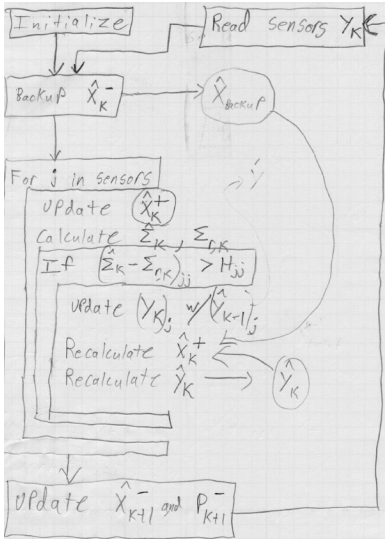
$$\Sigma_{r,k} = CP_{k|k}C^T + R \quad (13)$$

**Testing Statistic:** ($H$ is the threshold matrix)

$$S(T, k_0) = \hat{\Sigma}_k - \Sigma_{r,k} > H \quad (14)$$

**Initialize:**
$x_0 \sim \mathcal{N}(0, \Sigma)$, $\hat{x}_{0|-1} = \mathbf{0}$, $P_{0|-1} = \Sigma$, $Q$, $R$, $T$, $H$.
**while** $k \leq k_{\text{final}}$ **do**
  Make a backup of $\hat{x}_{k|k-1}$ in $\hat{x}_{k,\text{backup}}$.
  **for** Each sensor $j$ **do**
    Apply standard Kalman filter and estimate $\hat{x}_{k|k}$
    **calculate:** $\hat{\Sigma}_k = \frac{1}{T} \sum_{k_0}^{k_0+T-1} (y_k - \hat{y}_k)(y_k - \hat{y}_k)^T$
    **calculate:** $\Sigma_{r,k} = CP_{k|k}C^T + R$
    **if** $\left(\hat{\Sigma}_k - \Sigma_{r,k}\right)_{jj} > H_{jj}$ **then**
      Modify $j^{\text{th}}$ component of vector $y_k$ by replacing it with the $j^{\text{th}}$ component of vector $\hat{y}_{k-1}$.
      Re-estimate $\hat{x}_{k|k}$ via (6) using $\hat{x}_{k,\text{backup}}$ and the new $y_k$.
      Re-estimate $\hat{y}_k$ with the new $\hat{x}_{k|k}$.
    **end if**
  **end for**
  Predict $\hat{x}_{k+1|k}$ and $P_{k+1|k}$ via (7) and (8).
**end while**

# RWD Procedure




**Initialize:**
$x_0 \sim \mathcal{N}(0, \Sigma)$, $\hat{x}_{0|-1} = \mathbf{0}$, $P_{0|-1} = \Sigma$, $Q$, $R$, $T$, $H$.
**while** $k \leq k_{\text{final}}$ **do**
  Make a backup of $\hat{x}_{k|k-1}$ in $\hat{x}_{k,\text{backup}}$.
  **for** Each sensor $j$ **do**
    Apply standard Kalman filter and estimate $\hat{x}_{k|k}$
    **calculate:** $\hat{\Sigma}_k = \frac{1}{T}\sum_{k_0}^{k_0+T-1}(y_k - \hat{y}_k)(y_k - \hat{y}_k)^T$
    **calculate:** $\Sigma_{r,k} = CP_{k|k}C^T + R$
    **if** $\left(\hat{\Sigma}_k - \Sigma_{r,k}\right)_{jj} > H_{jj}$ **then**
      Modify $j^{\text{th}}$ component of vector $y_k$ by replacing it with the $j^{\text{th}}$ component of vector $\hat{y}_{k-1}$.
      Re-estimate $\hat{x}_{k|k}$ via (6) using $\hat{x}_{k,\text{backup}}$ and the new $y_k$.
      Re-estimate $\hat{y}_k$ with the new $\hat{x}_{k|k}$.
    **end if**
  **end for**
  Predict $\hat{x}_{k+1|k}$ and $P_{k+1|k}$ via (7) and (8).
**end while**

# Novel Residual Detector (NRD)

This method extends the $\chi^2$-detector method to individual sensors using a similar modified Kalman filter to RWD.

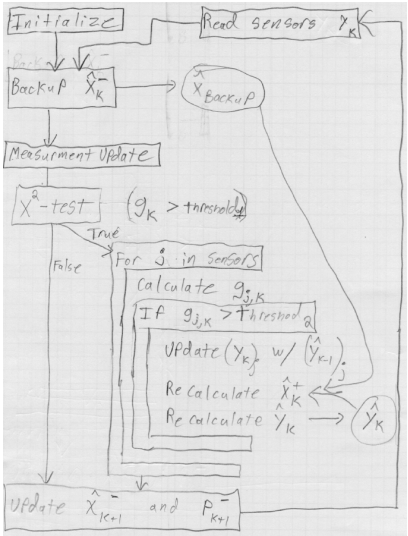
**Test Definitions:** (First is identical to $\chi^2$-test)

$$g_k = r_k^T \Sigma_{r,k}^{-1} r_k > threshold_1 \qquad (15)$$

$$g_{j,k} = \frac{r_k^2}{(\Sigma_{r,k})_{jj}} > threshold_2 \qquad (16)$$

**Initialize:**
$x_0 \sim \mathcal{N}(0, \Sigma)$, $\hat{x}_{0|-1} = \mathbf{0}$, $P_{0|-1} = \Sigma$, $Q$, $R$.
**while** $k \leq k_{\text{final}}$ **do**
  Make a backup of $\hat{x}_{k|k-1}$ in $\hat{x}_{k,\text{backup}}$.
  Apply standard Kalman filter and estimate $\hat{x}_{k|k}$
  **calculate:** $g_k = r_k^T \Sigma_{r,k}^{-1} r_k$
  **if** $g_k = r_k^T \Sigma_{r,k}^{-1} r_k > threshold_1$ **then**
    **for** Each sensor $j$ **do**
      **calculate:** $g_{j,k} = r_k^2 / (\Sigma_{r,k})_{jj}$
      **if** $g_{j,k} = r_k^2 / (\Sigma_{r,k})_{jj} > threshold_2$ **then**
        Modify $j^{\text{th}}$ component of vector $y_k$ by replacing it with the $j^{\text{th}}$ component of vector $\hat{y}_{k-1}$.
        Re-estimate $\hat{x}_{k|k}$ via (6) using $\hat{x}_{k,\text{backup}}$ and the new $y_k$.
        Re-estimate $\hat{y}_k$ with the new $\hat{x}_{k|k}$.
      **end if**
    **end for**
  **end if**
  Predict $\hat{x}_{k+1|k}$ and $P_{k+1|k}$ via (7) and (8).
**end while**

# NRD Procedure




**Initialize:**
$x_0 \sim \mathcal{N}(0, \Sigma)$, $\hat{x}_{0|-1} = \mathbf{0}$, $P_{0|-1} = \Sigma$, $Q$, $R$.
**while** $k \leq k_{\text{final}}$ **do**
  Make a backup of $\hat{x}_{k|k-1}$ in $\hat{x}_{k,\text{backup}}$.
  Apply standard Kalman filter and estimate $\hat{x}_{k|k}$
  **calculate:** $g_k = r_k^T \Sigma_{r,k}^{-1} r_k$
  **if** $g_k = r_k^T \Sigma_{r,k}^{-1} r_k > threshold_1$ **then**
    **for** Each sensor $j$ **do**
      **calculate:** $g_{j,k} = r_k^2 / (\Sigma_{r,k})_{jj}$
      **if** $g_{j,k} = r_k^2 / (\Sigma_{r,k})_{jj} > threshold_2$ **then**
        Modify $j^{\text{th}}$ component of vector $y_k$ by replacing it with the $j^{\text{th}}$ component of vector $\hat{y}_{k-1}$.
        Re-estimate $\hat{x}_{k|k}$ via (6) using $\hat{x}_{k,\text{backup}}$ and the new $y_k$.
        Re-estimate $\hat{y}_k$ with the new $\hat{x}_{k|k}$.
      **end if**
    **end for**
  **end if**
  Predict $\hat{x}_{k+1|k}$ and $P_{k+1|k}$ via (7) and (8).
**end while**

# Outline

# IEEE 14-bus Power Grid System [2]

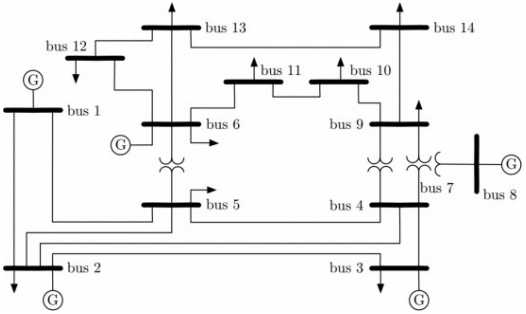
Effectiveness of RWD and NRD is tested on the IEEE 14-bus power grid system

- 5 generators and 14 buses
- 14 power injection and 20 power flow sensors

**System Parameters:** $n = 10$ and $p = 35$

**System States:**

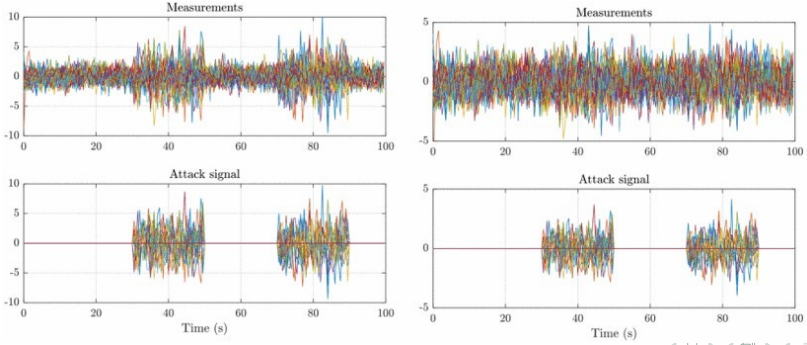
$\delta_i$ (rotor angle), $\omega_i$ (frequencies) $\forall\ i = \{1, \cdots, \frac{n}{2}\}$



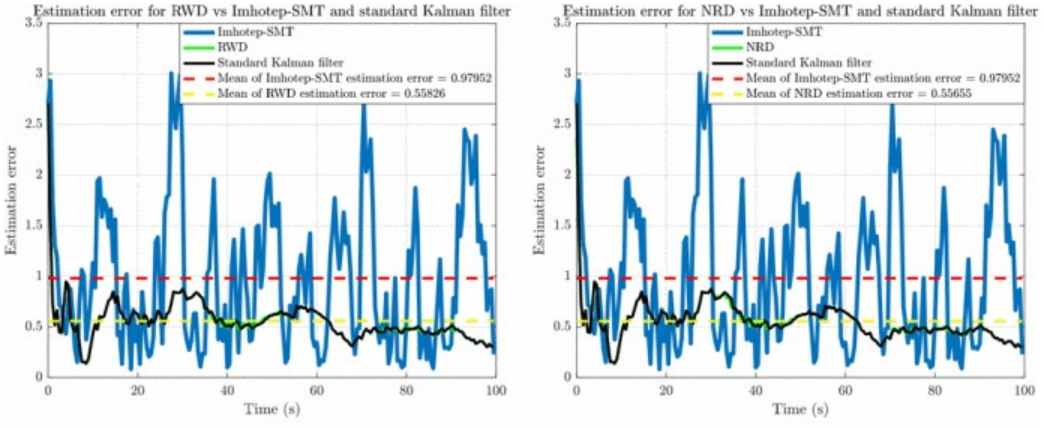
# Simulation Results Overview

The simulation tested the two methods for 3 cases:

- The Unstealthy Case - Attempts to disrupt system without remaining undetected
- The Stealthy Case - Injects false data upon the same order as the noise
- The Very Unstealthy Case - An unstealthy attack much larger then the noise

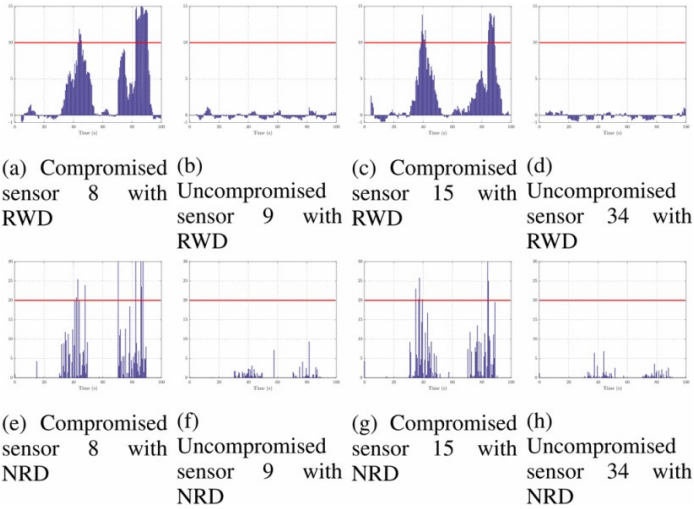


# The Unstealthy Case




Estimation error compared with the Imhotep-SMT tool [3] and a standard Kalman Filter.

# The Unstealthy Case - Individual Sensors



(a) Compromised sensor 8 with RWD

(b) Uncompromised sensor 9 with RWD

(c) Compromised sensor 15 with RWD

(d) Uncompromised sensor 34 with RWD

(e) Compromised sensor 8 with NRD

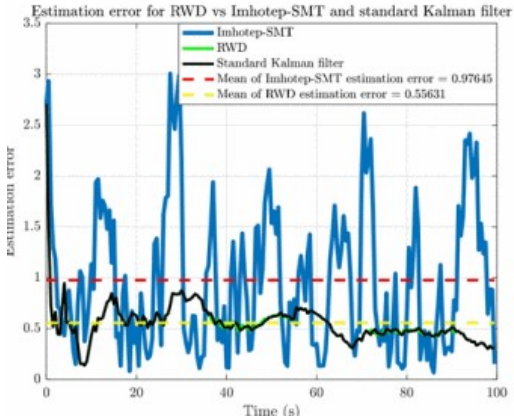
(f) Uncompromised sensor 9 with NRD

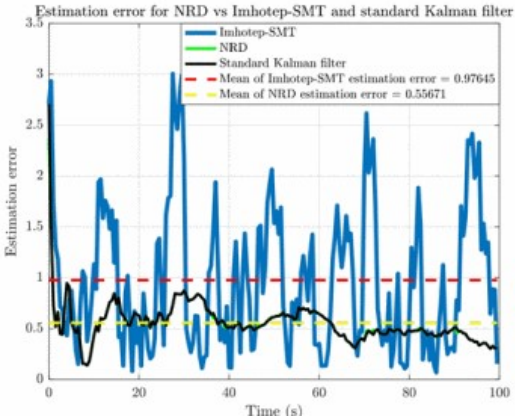
(g) Compromised sensor 15 with NRD

(h) Uncompromised sensor 34 with NRD

Residuals calculated for individual sensors for RWD and NRD (unstealthy attacks).

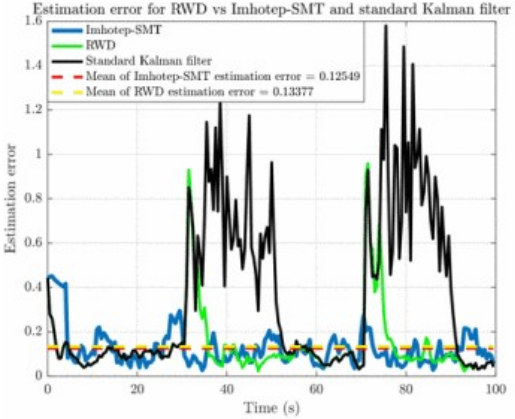
# The Stealthy Case

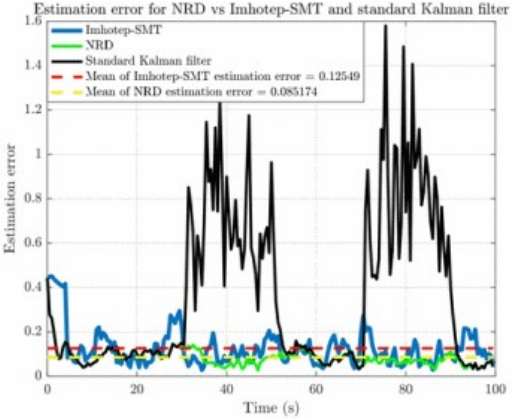



Estimation error compared with the Imhotep-SMT tool [3] and a standard Kalman Filter.

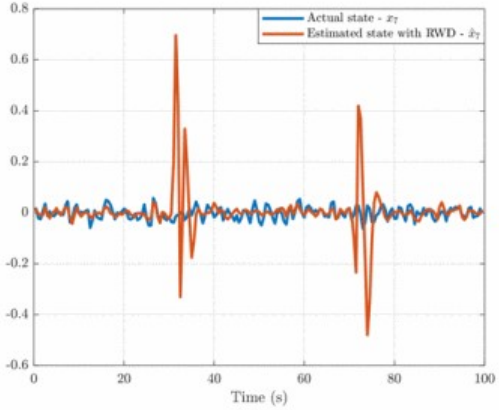
# The Very Unstealthy Case

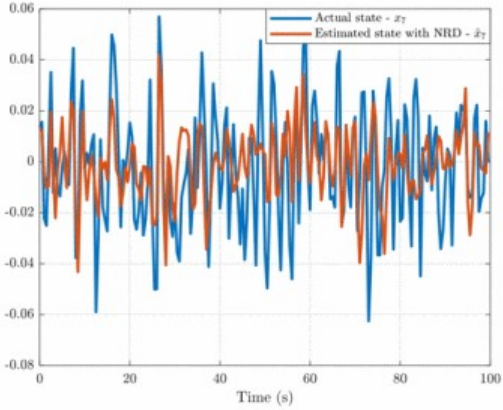



Estimation error compared with the Imhotep-SMT tool [3] and a standard Kalman Filter.

# The Very Unstealthy Case - Individual Sensor

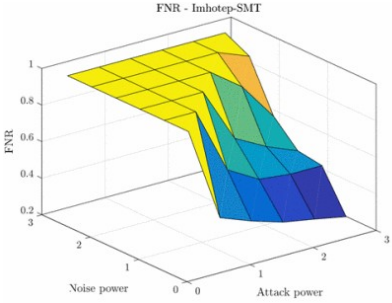


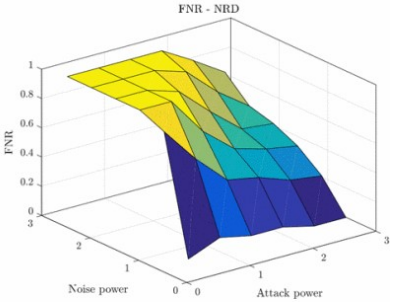

Estimation error of state $x_7$ for RWD and NRD (very unstealthy attacks).

# Detection Rate Analysis

**False Positive Rate (FPR):** $FNR = \frac{FN}{FP+TP}$



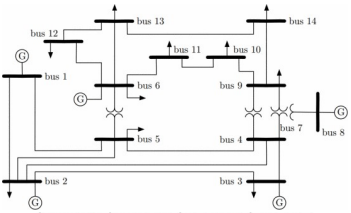

(a) FNR for Imhotep-SMT




(b) FNR for NRD

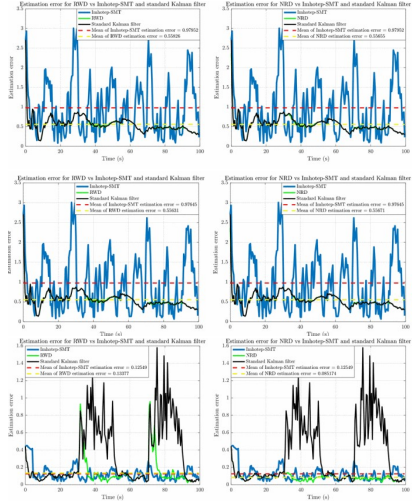
FNR for Imhotep-SMT vs NRD. NRD clearly performs better then Imhotep-SMT when the attack and noise power decrease (aka stealthy case).

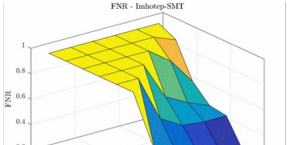
# Questions?



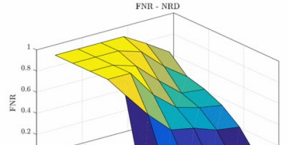

(a) Compromised sensor 8 with RWD (b) Uncompromised sensor 9 with RWD (c) Compromised sensor 15 with RWD (d) Uncompromised sensor 34 with RWD

(e) Compromised sensor 8 with NRD (f) Uncompromised sensor 9 with NRD (g) Compromised sensor 15 with NRD (h) Uncompromised sensor 34 with NRD




(a) FNR for Imhotep-SMT




(b) FNR for NRD

# Bibliography I


M. H. Basiri, J. G. Thistle, J. W. Simpson-Porco, and S. Fischmeister.
Kalman filter based secure state estimation and individual attacked sensor detection in cyber-physical systems.
In *2019 American Control Conference (ACC)*, pages 3841–3848, 2019.

F. Pasqualetti, F. Dörfler, and F. Bullo.
Cyber-physical attacks in power networks: Models, fundamental limitations and monitor design.
In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 2195–2201, 2011.

# Bibliography II


Y. Shoukry, P. Nuzzo, A. Puggelli, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and P. Tabuada.
Secure state estimation for cyber-physical systems under sensor attacks: A satisfiability modulo theory approach.
*IEEE Transactions on Automatic Control*, 62(10):4917–4932, 2017.