Model-based Attack Detection Scheme for Smart Water Distribution Networks

Chuadhry Mujeeb Ahmed
Carlos Murguia
Singapore University of Technology and Design
chuadhry@mymail.sutd.edu.sg
murguia_rendon@sutd.edu.sg

Justin Ruths
University of Texas Dallas
Dallas, USA
jruths@utdallas.edu

ABSTRACT

In this manuscript, we present a detailed case study about model-based attack detection procedures for Cyber-Physical Systems (CPSs). In particular, using EPANET (a simulation tool for water distribution systems), we simulate a Water Distribution Network (WDN). Using this data and sub-space identification techniques, an input-output Linear Time Invariant (LTI) model for the network is obtained. This model is used to derive a Kalman filter to estimate the evolution of the system dynamics. Then, residual variables are constructed by subtracting data coming from EPANET and the estimates of the Kalman filter. We use these residuals and the Bad-Data and the dynamic Cumulative Sum (CUSUM) change detection procedures for attack detection. Simulation results are presented - considering false data injection and zero-alarm attacks on sensor readings, and attacks on control input - to evaluate the performance of our model-based attack detection schemes. Finally, we derive upper bounds on the estimator-state deviation that zero-alarm attacks can induce.

1. INTRODUCTION

Cyber Physical Systems (CPSs) are the combination of computing resources and physical processes [14]. In the past, process control systems were completely isolated in the sense that they were not connected to the cyber space. However, with the arrival of new networking technologies, physical processes are being controlled and monitored through communication networks. These advancements have greatly improved the performance of our public infrastructures–e.g., transportation, smart grid, and water treatment facilities– but have also led to increased vulnerabilities against failures and attacks at the communication networks, which may serve as new access points for malicious agents trying to disrupt the system. Attacks on such systems may result in anything from performance degradation to physical damange, depending on the knowledge, capabilities, resources, and goals of the attacker.

ASIA CCS '17, April 02-06, 2017, Abu Dhabi, United Arab Emirates

DOI: http://dx.doi.org/10.1145/3052973.3053011

Water distribution networks are spread over hundreds of miles. These distributed systems are controlled by Programmable Logic Controllers (PLCs) and monitored by sensors and smart meters. To operate these networks, an operator supervises the system through a centralized computer using a Supervisory Control and Data Acquisition (SCADA) system [8]. PLCs gather data coming from the sensors and use this information to compute control actions to be sent to the actuators. This automation relies on the cyber infrastructure to exchange information between devices. The addition of this cyber layer makes the water system vulnerable to different types of cyber-physical attacks. For instance, in 2000, Maroochy Shire (Australia) sewage system was hacked by a disgruntled employee. This hacking led to the spillage of around one million liters of waste water into parks and water ways [25]. According to a report by U.S. Industrial Control System Cyber Emergency Response Team (ICS-CERT), several attacks have occurred against water utilities in the USA [2]. The critical nature of water infrastructures makes them an attractive target for hackers and terrorist activities. Therefore, it is extremely important to ensure security of these systems.

The work in [5, 6] focuses on a water canal network. The authors characterize the effect of adversarial attacks on sensor readings. They conducted field tests to support their proposed approach. However, such methods cannot be directly applied to water distribution networks, because attackers have more access points to disrupt the system, due to complexity of the system. The work in [23] studies vulnerabilities in water distribution networks. They propose a game-theoretic approach to detect and minimize loss due to attacks.

In this manuscript, we propose a control-theoretic model-based approach for detection of sensor and actuator attacks of WDNs. We obtain a dynamical model of the system from sensor data and use statistical change detection techniques for attack detection. Most of the related work focuses on static detection procedures–e.g., chi-squared and bad-data detectors [18, 4, 13, 16]. These procedures detect attacks based on a single measurement at a time. However, in context of CPSs security, only a few papers have considered the use of dynamic detectors like the Cumulative Sum (CUSUM) procedure, which employs sensor measurement history [9, 10, 19]. Here, for both the Bad-Data and the CUSUM procedures, we study how features of the system (e.g., system matrices, estimator gains, and noise) relate to the performance of the attack detector (e.g., state estimate deviation and false alarm rate).

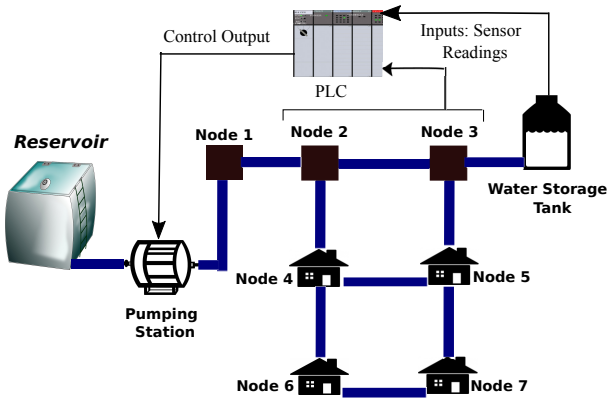


Figure 1: Case Study: Water Distribution Network

We provide a comprehensive study of a real world WDN designed and emulated in EPANET. We run EPANET for different demand patterns and collect the corresponding output data (simulated sensor measurements) e.g., water level in the water storage tank and flow/pressure at the junctions/nodes. This data is used to derive a state space Linear Time Invariant (LTI) model for the WDN using subspace identification techniques [21]. This model is employed to construct a Kalman filter to estimate the evolution of the system dynamics. Next, we construct residual variables by subtracting data coming from EPANET and the estimates of the Kalman filter. We use these residuals and the Bad-Data and the dynamic Cumulative Sum (CUSUM) change detection procedures for attack detection. Limitations of these statistical detectors are analyzed under a class of zero-alarm attacks. Simulation results compare the performance of the attack detectors under different attack scenarios.

The rest of the paper is organized as follows. In Section 2, the system description and the attack detection scheme are presented. In particular, the proposed water distribution network, the Kalman filter (and the residuals generation), and the CUSUM and Bad-Data procedures are introduced. In Section 3, the attacker model and the implemented attacks are presented. Performance limitations of the attack detectors and state estimate deviations under attacks are analyzed in Section 4. Section 5 discusses the simulation results that compare the theoretical analysis with experimental observations.

2. SYSTEM DESCRIPTION AND ATTACK DETECTION

In this section, we introduce the topology of the water distribution network considered here. A linear time invariant system model is obtained using subspace identification techniques. Then, we construct a Kalman filter which is used to construct attack detection schemes. A block diagram for the proposed method is shown in Figure 2.

2.1 Water Distribution Network

The proposed water distribution network is modeled in EPANET [1]. EPANET is a software tool used to model and simulate water systems. The schematic of the network is depicted in Figure 1. It consists of a water reservoir, a storage tank, a pump, and seven nodes/junctions. Nodes 4, 5, 6, and 7 represent four consumers. The consumer nodes have time-varying demand patterns based on their water requirements. The controller has the knowledge of the demand patterns, the water level in the tank, and the pressures at the junctions. This network is simulated in EPANET and data is collected for all measurable outputs and the input demands. These hydraulic simulations are carried out with a simulation time period of 10 days (240 hours) and a time step of 15 minutes (which we also use as sampling time for generating the state space model). The junctions and water level in the tank are used as outputs of the control system and the demands of the user nodes and the pump status are used as inputs to generate a state space model of the system. Using data collected under regular operation (no attacks) and subspace identification techniques [21], we approximate the input-output dynamical model of the WDN by a set of Linear Time Invariant (LTI) stochastic difference equations. In particular, we obtain a discrete time state space model of the form:

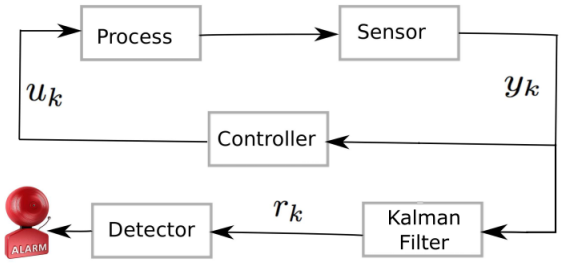


Figure 2: Block diagram of the estimation-based control and attack-detection.

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + v_k, \\ y_k = Cx_k + \eta_k. \end{cases} \quad (1)$$

where $k \in \mathbb{N}$ is the discrete time index, $x_k \in \mathbb{R}^n$ is the state of the approximated model, (its dimension depends on the order of the approximated model), $y \in \mathbb{R}^m$ are the measured outputs, and $u \in \mathbb{R}^p$ denote the demand patterns. The system identification problem is to determine the system matrices A, B, C from input-output data. The obtained model provides a 70% fit between measurements and simulated outputs (generated using the approximated model) with 10 states, i.e., $n = 10$ (the matrices are shown in appendix). We also identified a few higher and lower order models. Ultimately, the model with 10 states has a nice trade-off between prediction error and the dimension of the model. The quality of the identified model is validated by looking at the system evolution based on the identified state space matrices and initial state x_1. The closeness of the system evolution to the sensor measurements obtained from EPANET indicates that this model is a faithful representation of the water distribution network (see Figure 3). The top pane shows the sensor readings from EPANET as well as the modeled output for the water level sensor using system matrices. We can observe that modeled output is very close to sensor readings, resulting in small error(error is shown in middle pane while it's probability distribution is shown in bottom pane).

At the time-instants $k \in \mathbb{N}$, the output of the process y_k

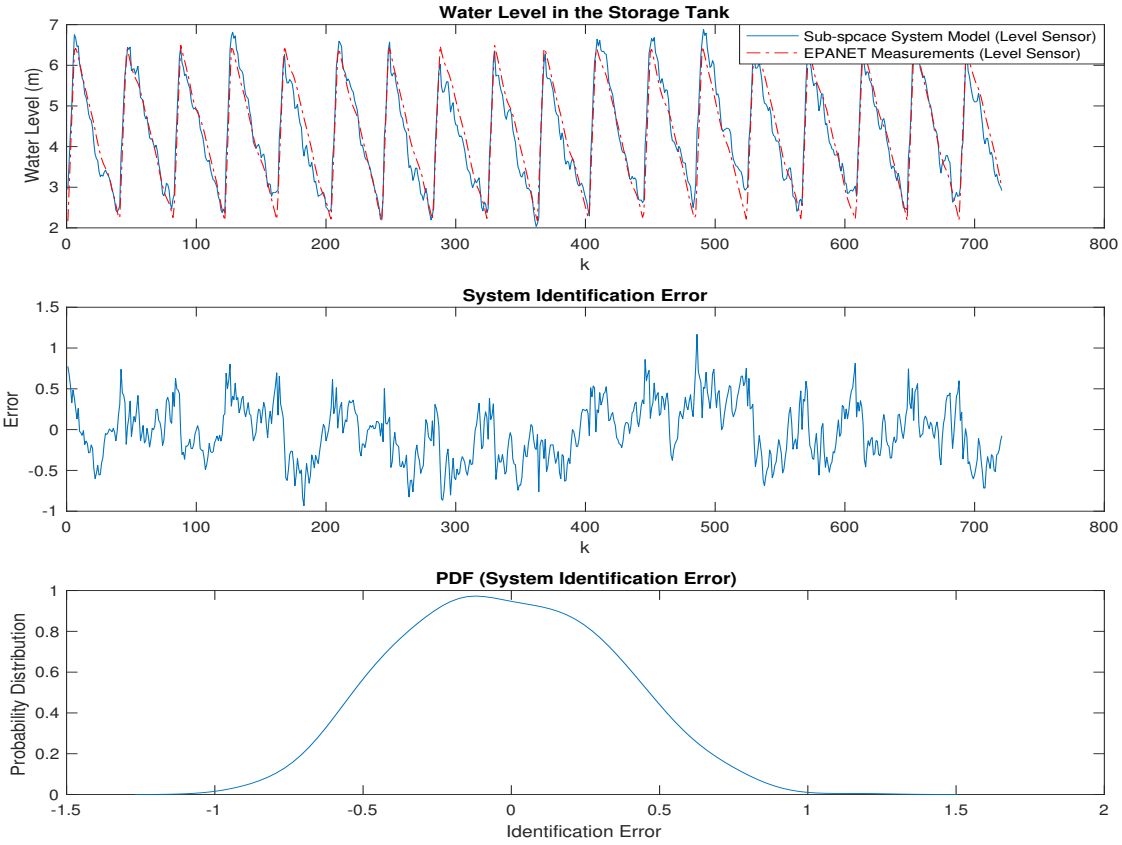


Figure 3: Validating the system model obtained using subspace identification method.

is sampled and transmitted over a communication channel (2). The received output $\bar{y}_k$ is used to compute control actions u_k which are sent back to the process. Throughout this control-loop, there are many potential points where an attacker can hack into the system. For instance, man-in-the-middle attacks at the communication channels and physical attacks directly on the infrastructure. In this manuscript, we focus on sensor and actuator attacks, which could be accomplished through a man-in-the-middle scheme and/or a replacement of onboard PLC software with malware. After each transmission and reception, the attacked output $\bar{y}_k$ takes the form:

$$\bar{y}_k := y_k + \delta_k = Cx_k + \eta_k + \delta_k, \tag{2}$$

where $\delta_k \in \mathbb{R}^m$ denotes sensor attacks. Throughout this manuscript, we reserve the variable k as the discrete-time index of various sequences; where clear, we omit reminding the reader that $k \in \mathbb{N}$.

2.2 Attack Detection Framework

In this section, we explain the details of our attack detection scheme. First, we discuss the Kalman filter based state estimation and residual generation. Then, we present our residual-based attack detection procedures (namely the CUSUM and Bad-Data detectors).

2.2.1 *Kalman Filter*

To estimate the state of the system based on the available output y_k, we use a linear filter with the following structure:

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L_k\big(\bar{y}_k - C\hat{x}_k\big), \tag{3}$$

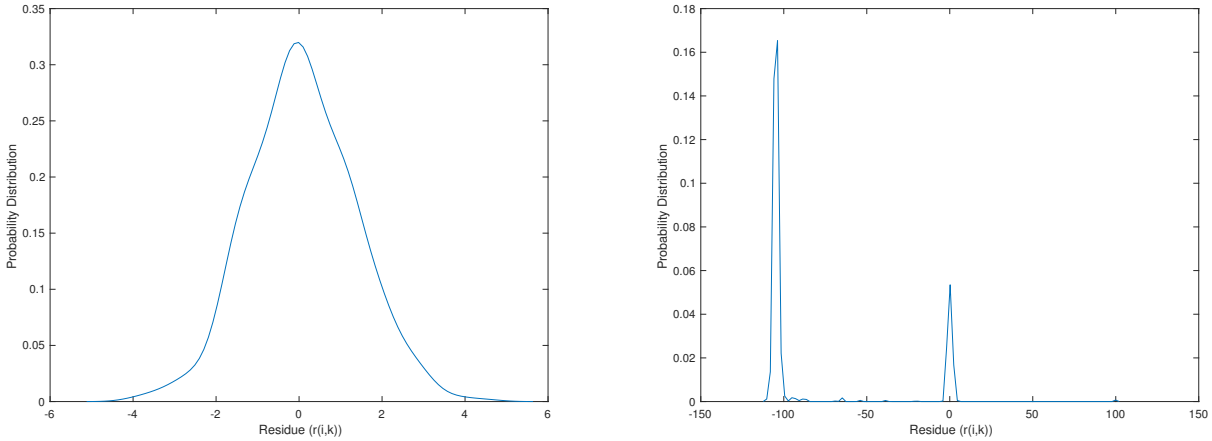
with estimated state $\hat{x}_k \in \mathbb{R}^n$, $\hat{x}_1 = E[x(t_1)]$, where $E[\cdot]$ denotes expectation, and gain matrix $L_k \in \mathbb{R}^{n\times m}$. Define the estimation error $e_k := x_k - \hat{x}_k$. In the Kalman filter, the matrix L_k is designed to minimize the covariance matrix $P_k := E[e_k e_k^T]$ (in the absence of attacks). Given the system model (1),(2) and the estimator (3), the estimation error is governed by the following difference equation

$$e_{k+1} = \big(A - L_k C\big)e_k - L_k\eta_k - L_k\delta_k + v_k. \tag{4}$$

If the pair (A, C) is detectable, the covariance matrix converges to steady state in the sense that $\lim_{k\to\infty} P_k = P$ exists [7]. We assume that the system has reached steady state before an attack occurs. Then, the estimation of the random sequence $x_k, k \in \mathbb{N}$ can be obtained by the estimator (3) with P_k and L_k in steady state. It can be verified that, if $R_2 + CPC^T$ is positive definite, the following estimator gain

$$L_k = L := \big(APC^T\big)\big(R_2 + CPC^T\big)^{-1}, \tag{5}$$

leads to the minimal steady state covariance matrix P, with



(a) Normal operation (no attack).

(b) System under attack (bias injection attack).

Figure 4: (a): Probability distribution of the residual for water level sensor measurements without attack. (b): Probability distribution of the residual for water level sensor measurements with bias injection attack.

P given by the solution of the algebraic Riccati equation:

$$APA^T - P + R_1 = APC^T(R_2 + CPC^T)^{-1}CPA^T. \quad (6)$$

The reconstruction method given by (3)-(6) is referred to as the steady state Kalman Filter, cf. [7].

2.2.2 Residuals and Hypothesis Testing

Although the notion of residuals and model-based detectors is now routine in the fault detection literature, the primary focus has been on detecting and isolating faults that occur with a specific structure (e.g., bias drifts). Now, in the context of an intelligent adversarial attacker, new challenges arise to understand the worst case effect that an intruder can have on the system. While fault detection techniques can be used to detect attacks; it is important to assess the performance of such methods against an intelligent adversary. In this work, by means of our simulation study, we assess the performance of two model-based fault detection procedures (the chi-squared and the CUSUM procedures) for a variety of attacks. These procedures rely on a state estimator (e.g., Kalman filter) to predict the evolution of the system. The estimated values are compared with sensor measurements $\bar{y}_k$ (which may have been attacked). The difference between the two should stay within a certain threshold under normal operation, otherwise an alarm is triggered to point a potential attack. Define the residual random sequence $r_k, k \in \mathbb{N}$ as

$$r_k := \bar{y}_k - C\hat{x}_k = Ce_k + \eta_k + \delta_k. \quad (7)$$

If there are no attacks, the mean of the residual is

$$E[r_{k+1}] = CE[e_{k+1}] + E[\eta_{k+1}] = \mathbf{0}_{m\times 1}. \quad (8)$$

where $\mathbf{0}_{m\times 1}$ denotes an $m\times 1$ matrix composed of only zeros, and the covariance is given by

$$\Sigma := E[r_{k+1}r_{k+1}^T] = CPC^T + R_2. \quad (9)$$

For this residual, we identify two hypothesis to be tested, $\mathcal{H}_0$ the *normal mode* (no attacks) and $\mathcal{H}_1$ the *faulty mode* (with attacks). For our particular case of study, the pressure at the nodes and the water level in the tank are the outputs of the system. Using this data along with the state estimates, we construct our residuals. Then, we have:

$$\mathcal{H}_0 : \begin{cases} E[r_k] = \mathbf{0}_{m\times 1}, \\ E[r_k r_k^T] = \Sigma, \end{cases} \text{ or } \quad \mathcal{H}_1 : \begin{cases} E[r_k] \neq \mathbf{0}_{m\times 1}, \\ E[r_k r_k^T] \neq \Sigma. \end{cases}$$

Figure 4 shows the approximated distributions of the residuals of the water level in the storage tank for both the attacked and the attack-free cases. In Figure 4(b), the residual under a bias injection attack (simple constant offset on the sensor measurements) is depicted. Our hypothesis can easily be verified by looking at the probability distribution of residuals. Our null hypothesis $\mathcal{H}_0$ which follows a zero mean normal distribution with variance Σ is also verified from plot in Figure 4(a). Similarly, for the attacked scenario $\mathcal{H}_1$, we do not have a zero mean normally distributed residual as it is shown in Figure 4(b). We can formulate the hypothesis testing in a more formal manner using existing change detection techniques based on the statistics of the residuals.

2.2.3 Cumulative Sum (CUSUM) Detector

The CUSUM procedure is driven by the residual sequences. In particular, the input to the CUSUM procedure is a *distance measure*, i.e., a measure of how deviated the estimator is from the actual system, and this measure is a function of the residuals. In this work, we assume there is a dedicated detector on each sensor (or on any sensor we want to include in the detection scheme). Throughout the rest of this paper we will reserve the index i to denote the sensor/detector, $i \in \mathcal{I} := \{1, 2, \ldots, m\}$. Thus, we can partition the attacked output vector as $\bar{y}_k = \text{col}(\bar{y}_{k,1}, \ldots, \bar{y}_{k,m})$ where $\bar{y}_{k,i} \in \mathbb{R}$ denotes the i-th entry of $\bar{y}_k \in \mathbb{R}^m$; then

$$\bar{y}_{k,i} = C_i x_k + \eta_{k,i} + \delta_{k,i}, \quad (10)$$

with C_i being the i-th row of C and $\eta_{k,i}$ and $\delta_{k,i}$ denoting the i-th entries of η_k and δ_k, respectively. Inspired by the empirical work in [9], we propose the absolute value of the entries of the residual sequence as distance measure, i.e.,

$$z_{k,i} := |r_{k,i}| = |C_i e_k + \eta_{k,i} + \delta_{k,i}|. \quad (11)$$

Note that, if there are no attacks, $r_{k,i} \sim \mathcal{N}(0, \sigma_i^2)$ (see Figure 4(a)), where σ_i^2 denotes the i-th entry of the diagonal of the covariance matrix Σ. Hence, $\delta_k = \mathbf{0}$ implies that $|r_{k,i}|$ follows a *half-normal distribution* [24] with

$$E\big[|r_{k,i}|\big] = \frac{\sqrt{2}}{\sqrt{\pi}}\sigma_i \text{ and } \text{var}\big[|r_{k,i}|\big] = \sigma_i^2\Big(1 - \frac{2}{\pi}\Big). \quad (12)$$

Next, having presented the notion of distance measure, we introduce the CUSUM procedure. For a given *distance measure* $z_{k,i} \in \mathbb{R}$, the CUSUM of Page [22] is written as follows.

CUSUM: $S_{0,i} = 0, \quad i \in \mathcal{I}$,

$$\begin{cases} S_{k,i} = \max(0, S_{k-1,i} + z_{k,i} - b_i), & \text{if } S_{k-1,i} \leq \tau_i, \\ S_{k,i} = 0 \text{ and } \tilde{k}_i = k-1, & \text{if } S_{k-1,i} > \tau_i. \end{cases} \quad (13)$$

Design parameters: bias $b_i > 0$ and threshold $\tau_i > 0$.
Output: alarm time(s) $\tilde{k}_i$.

From (13), it can be seen that $S_{k,i}$ accumulates the distance measure $z_{k,i}$ over time. When this accumulation becomes greater than a certain threshold τ_i an alarm is raised. The sequence $S_{k,i}$ is reset to zero each time it becomes negative or larger than τ_i. If $z_{k,i}$ is an independent non-negative sequence (which is our case) and b_i is not sufficiently large, the CUSUM sequence $S_{k,i}$ grows unbounded until the threshold τ_i is reached, no matter how large τ_i is set. In order to prevent these drifts, the bias b_i must be selected properly based on the statistical properties of the distance measure. Once the the bias is chosen, the threshold τ_i must be selected to fulfill a required false alarm rate $\mathcal{A}_i^*$. The occurrence of an alarm in the CUSUM when there are no attacks to the CPS is referred to as a false alarm, and $\mathcal{A}_i \in [0, 1]$ denotes the *false alarm rate* for the CUSUM procedure defined as the expected proportion of observations which are false alarms [3, 27].

2.2.4 Bad-Data Detector

We have also implemented the Bad-Data detector for this case of study because it is widely used in the CPS security literature [11, 17]. We also present a performance comparison between the CUSUM and the Bad-Data detectors. For the residual sequence $r_{k,i}$ given by (7), the Bad-Data detector is defined as follows.

Bad-Data Procedure:

$$\text{If } \; |r_{k,i}| > \alpha_i, \quad \tilde{k}_i = k, \quad i \in \mathcal{I}. \quad (14)$$

Design parameter: threshold $\alpha_i > 0$.
Output: alarm time(s) $\tilde{k}_i$.

Using the Bad-Data detector an alarm is triggered if distance measure $|r_{k,i}|$ exceeds the threshold α_i. Similar to the CUSUM procedure, the parameter α_i is selected to satisfy a required false alarm rate $\mathcal{A}_i^*$.

3. ATTACKER AND ATTACK MODELS

In this section, we introduce the types of attacks launched on our water distribution network. Essentially, the attacker model encompasses the attacker's intentions and it's capabilities. The attacker may choose its goals from a set of intentions [26], including performance degradation, disturbing a physical property of the system, or damaging a component. In our experiments, three classes of attacks are modeled and executed. It is assumed that the attacker has access to $y_{k,i} = C_i y_k + \eta_{k,i}$ (i.e., the opponent has access to sensor measurements). Also, the attacker knows the system dynamics, the state space matrices, the control inputs and outputs, and the implemented detection procedure. The adversary has perfect knowledge of the Kalman filter and can modify the sensor readings to an arbitrary value.

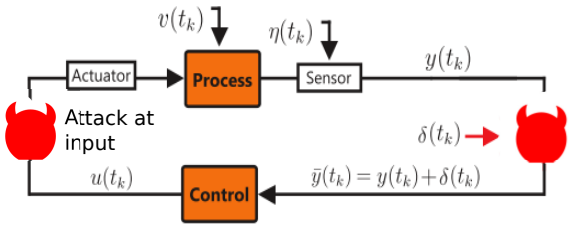


Figure 5: CPS under attack.

1. *Bias Injection Attack*: First, a failure-like attack is designed. The attacker's goal is to deceive the control system by sending incorrect sensor measurements. In this scenario, the level sensor measurements are increased while the actual tank level is invariant. This makes the controller think that the attacked values are true sensor readings; and hence, the water pump keeps working until the tank is empty and the pump is burned out. The attack vector can be defined as,

$$\bar{y}_k = y_k + \eta_k + \bar{\delta}, \quad (15)$$

where $\bar{\delta}$ is the bias injected by the attacker.

2. *Zero-Alarm Attack*: The second attack is more sophisticated and is carried out by carefully generating δ_k to drive the system to an undesired state. The objective of this attack is to maximize the damage without raising alarms. This attack is designed to deceive the detection schemes explained in Section 2.2.3 and Section 2.2.4. A detailed analysis on how to design such an attack is presented in Section 4.2.1. This attack does not cause alarms because the injected value and the previous steady state measurement differ only in a small amount, then the residual value would not be sufficiently large to raise an alarm. By knowing the parameters of the detection procedure, it is always possible to modify the sensor values by an amount such that the residuals never cross the detection thresholds.

3. *Attack on Control Inputs*: In the third type of attack, the attacker changes the inputs to the actuators. In our case of study, the user demand patterns are the control inputs. By changing the user demands, the attacker makes the controller think the demand has been modified, ultimately leading to over/under pumping of water. An schematic example of such an attack is shown in Figure 5. This attack is executed on the EPANET simulator and the details of the attack are not available to us. This is intentionally done to test our detection methods against completely unknown attacks. Although we do not explicitly model these input attacks, we observe that they lead to changes in

the residuals such that our residual-based detection methodology can be used as well.

4. PERFORMANCE LIMITATIONS OF ATTACK DETECTORS

As specified in the previous section, it is important to carefully select the parameters of the detectors. For the bad-data detector, we only have to take care of threshold α_i but for CUSUM, we have two parameters, the bias b_i and the threshold τ_i. For selecting the thresholds, it is intuitive to select them not too small or too large. Small thresholds result in increased false alarms while large ones may result in undetected attacks. For the CUSUM, too small values of bias b_i leads to unbounded growth of the CUSUM sequence while too large b_i hides the effect of the attacker. In [19, 20], the authors present tools for selecting b_i and τ_i based on the statistical properties of the distance measure $z_{k,i}$. In what follows, we briefly introduce these tools.

4.1 Boundedness and False Alarm Rate

Consider the closed-loop system (1),(3)-(6). Assume that sensors $y_{k,i}$ are monitored for attack detection. First, for $i \in \{1, ..., m\}$, let $\delta_{k,i} = 0$ and consider the CUSUM procedure (13) with distance measure $z_{k,i} = |r_{k,i}|$ and residual sequence (7). According to Theorem 1 in [19], the bias b_i must be selected larger than $\bar{b}_i = \sigma_i\sqrt{2/\pi}$ to ensure mean square boundedness of $S_{k,i}$ independent of the threshold τ_i. The standard deviation σ_i is given by the square root of the i-th entry of the residual covariance matrix Σ given in (9). In our analysis we set $b_i = 2\bar{b}_i$. Next, for the desired false alarm rate $\mathcal{A}_i^* = 0.01(1\%)$, we compute the corresponding thresholds $\tau_i = \tau_i^*$, using Theorem 2 and Remark 2 in [19].

For the bad-data detector, we can also find the thresholds α_i using the tools [19]. That is, if

$$\alpha_i = \alpha_i^* := \sqrt{2}\sigma_i \text{erf}^{-1}(1 - \mathcal{A}_i^*), \tag{16}$$

where $\text{erf}(\cdot)$ denotes the error function [15]. Then, $\mathcal{A}_i = \mathcal{A}_i^*$ for attack-free systems with $r_{k,i} \sim \mathcal{N}(0, \sigma_i^2)$, where $\mathcal{A}_i$ denotes the actual false alarm rate and $\mathcal{A}_i^*$ is the desired false alarm rate.

4.2 State Estimation Under Attacks

In this section, we assess the performance of the bad-data and the CUSUM procedures by quantifying the effect of the attack sequence δ_k on process dynamics when they are used to detect anomalies. In particular, we characterize for a class of zero-alarm attacks, the largest deviation on the estimation error due to the attack sequence. We derive upper bounds on the expectation of the estimation error given the system dynamics, the Kalman filter, the attack sequence, and the parameters of the detection procedure. For the same class of attacks, we quantify the largest deviation of the expectation for the estimation error when using the bad-data procedure and then compare it with the one obtained with the CUSUM.

4.2.1 Impact of Zero-Alarm Attacks

In this section we will evaluate the impact of a class of zero-alarm attacks on system state estimation. This can be termed as worst case analysis as the attacker is able to do damage and still not get detected. As stated in section 3, the attacker has complete knowledge of system dynamics and detection algorithms. Based on this information an attacker generates an attack sequence δ_k, such that detection algorithms would not generate an alarm. Although this attack goes undetected it can induce changes in the system dynamics and here we analyze such a disturbance in this section. First, consider the bad-data procedure and write the left-hand side of (14) in terms of the estimation error e_k:

$$|r_{k,i}| = |C_i e_k + \eta_{k,i} + \delta_{k,i}|, \quad i \in \mathcal{I}. \tag{17}$$

By assumption, the attacker has access to $y_{k,i} = C_i y_k + \eta_{k,i}$. Moreover, given its perfect knowledge of the Kalman filter, the opponent can compute the estimated output $C_i\hat{x}_k$ and then construct $C_i e_k + \eta_{k,i}$. It follows that

$$\delta_{k,i} = -C_i e_k - \eta_{k,i} + \alpha_i \rightarrow |r_{k,i}| = \alpha_i, \quad i \in \mathcal{I}, \tag{18}$$

is a feasible attack sequence given the capabilities of the attacker. These attacks maximize the damage to the CPS by immediately saturating and maintaining $|r_{k,i}|$ at the threshold α_i. Define $\alpha := \text{col}(\alpha_1, \ldots, \alpha_m)$. Then, the expectation of the estimation error under the attack (18) is given by

$$E[e_{k+1}] = AE[e_k] - L\alpha. \tag{19}$$

If $\rho[A] > 1$, then $||E[e_k]||$ diverges to infinity as $k \rightarrow \infty$ [7]. That is, the attack sequence (18) destabilizes the system if $\rho[A] > 1$. If $\rho[A] \leq 1$, then $|E[e_k]|$, may or may not diverge to infinity depending on algebraic and geometric multiplicities of the eigenvalues with unit modulus of A (a known fact from stability of LTI systems [7]).

Proposition 1. Consider the process (1), the Kalman filter (3)-(6), and the Bad-Data procedure (14). Let the sensors be attacked by the bad-data zero-alarm attack sequence (18). Then, if $\rho[A] < 1$, it is satisfied that $\lim_{k\rightarrow\infty} ||E[e_k]|| = \gamma_{\text{BD}}$, where $\gamma_{\text{BD}} := ||(I - A)^{-1}L\alpha||$.

Proof: By assumption $\rho[A] < 1$. This implies that $(I-A)$ is invertible; hence, system (19) has a unique equilibrium given by $E[e_k] = \bar{e} := (A - I)^{-1}L\alpha$. From (19), it is easy to verify that $E[e_k] - \bar{e}$ satisfies the following difference equation

$$E[e_{k+1}] - \bar{e} = A(E[e_k] - \bar{e}).$$

Therefore, $\rho[A] < 1$ imply that the equilibrium $\bar{e}$ is exponentially stable [7], i.e., $\lim_{k\rightarrow\infty} E[e_k] = \bar{e}$. The Euclidean norm on $\mathbb{R}^n$ is a continuous function from $\mathbb{R}^n$ to $\mathbb{R}_{\geq 0}$ [12]. It follows that $\lim_{k\rightarrow\infty} ||E[e_k]|| = ||\lim_{k\rightarrow\infty} E[e_k]|| = ||\bar{e}||$ and the assertion follows. ■

Next, consider the CUSUM procedure and write (13) in terms of the estimation error e_k:

$$S_{k,i} = \max(0, S_{k-1,i} + |C_i e_k + \eta_{k,i} + \delta_{k,i}| - b_i), \tag{20}$$

if $S_{k-1,i} \leq \tau_i$; and $S_{k,i} = 0$, if $S_{k-1,i} > \tau_i$. As with the bad-data, we look for attack sequences that immediately saturate and then maintain the CUSUM statistic at the threshold $S_{k,i} = \tau_i$. Assume that the attack starts at some $k = k^* \geq 2$ and $S_{k^*-1,i} \leq \tau_i$, i.e., the attack does not start immediately after a false alarm. Consider the attack:

$$\delta_{k,i} = \begin{cases} \tau_i + b_i - C_i e_k - \eta_{k,i} - S_{k-1,i}, & k = k^*, \\ b_i - C_i e_k - \eta_{k,i}, & k > k^*. \end{cases} \tag{21}$$

Launching a zero-alarm attack to deceive a CUSUM detector is not as simple as doing the same for the bad-data detector. Since CUSUM test depends on accumulated sum over

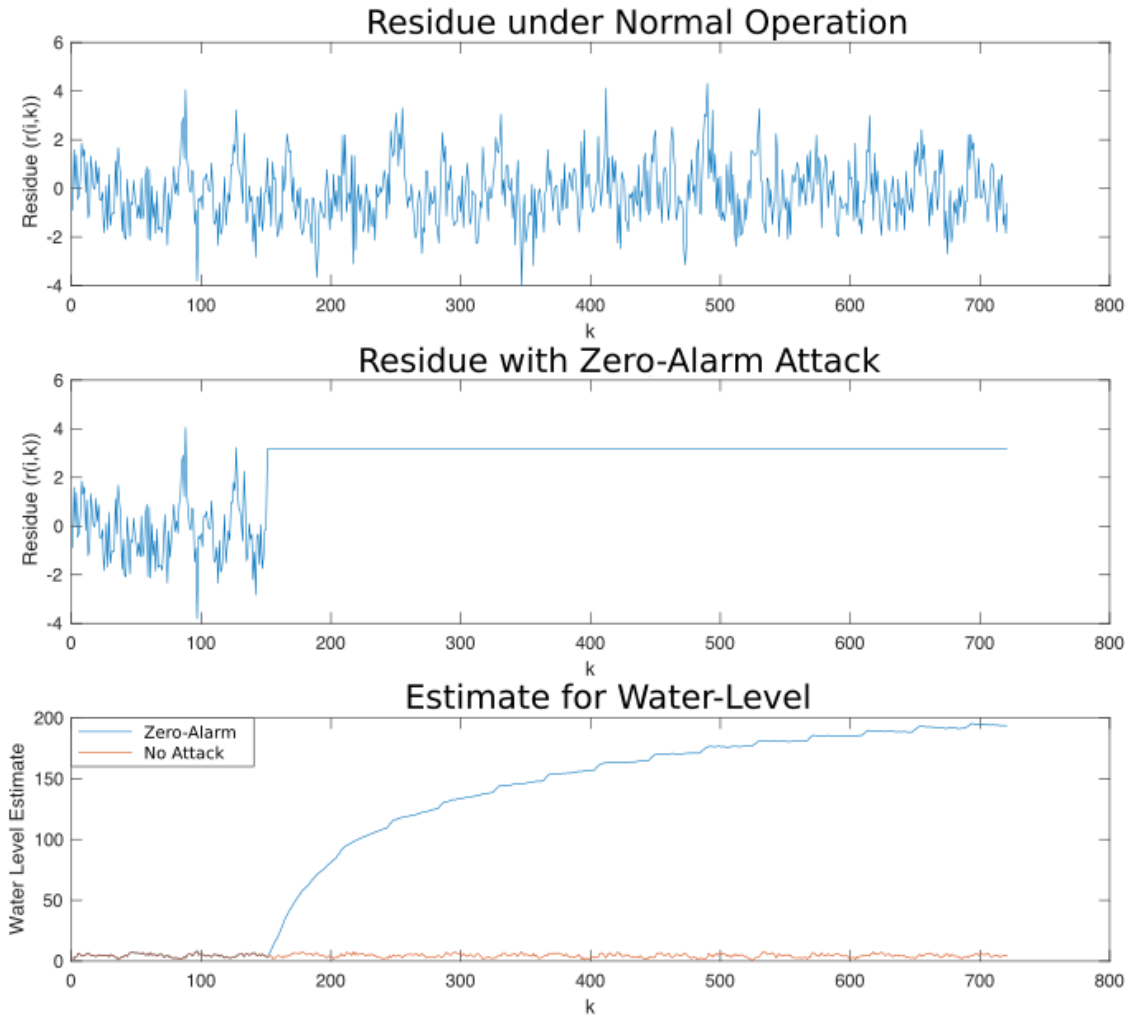


Figure 6: Bad-data detection method under a zero-alarm attack for scenario 1.

the time, to launch an attack as in (21), an attacker needs to know $S_{k^*-1,i}$, i.e., the value of the CUSUM sequence one step before the attack. This is a strong assumption since it represents a real-time quantity that is not communicated over the communication network. Even if the opponent has access to the parameters of the CUSUM, (b_i, τ_i), given the stochastic nature of the residuals, the attacker would need to know the complete history of observations (from when the CUSUM was started) to be able to reconstruct $S_{k^*-1,i}$ from data. This is an inherent security advantage in favor of the CUSUM over static detectors like the bad-data or chi-squared. Nevertheless, for evaluating the worst case scenario, we assume that the attacker has access to $S_{k^*-1,i}$. Define $b := \text{col}(b_1, \dots, b_m)$, $\tau := \text{col}(\tau_1, \dots, \tau_m)$, and, without loss of generality, assume $k^* = 2$. Then, by construction, $E[x_i] = E[e_i] = \mathbf{0}$, $i = 1, 2$, and the expectation of the closed-loop system under the attack sequence (21) is written as: $E[x_3] = \mathbf{0}$, $E[e_3] = -L\tau$, and, for $k > k^* = 2$,

$$E[e_{k+1}] = AE[e_k] - Lb. \tag{22}$$

Proposition 2. Consider the process (1), the Kalman filter (3)-(6), and the CUSUM procedure (13). Let the sensors be attacked by the CUSUM zero-alarm attack sequence (21). Then, if $\rho[A] < 1$, it is satisfied that $\lim_{k\to\infty} |E[e_k]| = \gamma_{\text{CS}}$, where $\gamma_{\text{CS}} := ||(I - A)^{-1}Lb||$.

The proof Proposition 2 follows the same lines as the proof of Proposition 1 and it is omitted here.

5. RESULTS AND DISCUSSION

In this section, we present the obtained simulation result. The CUSUM and the bad-data procedures are based on the statistical properties of the residuals. In Figure 4(b), we show the probability distributions of the water level residual under bias attacks and without attacks. It is evident that

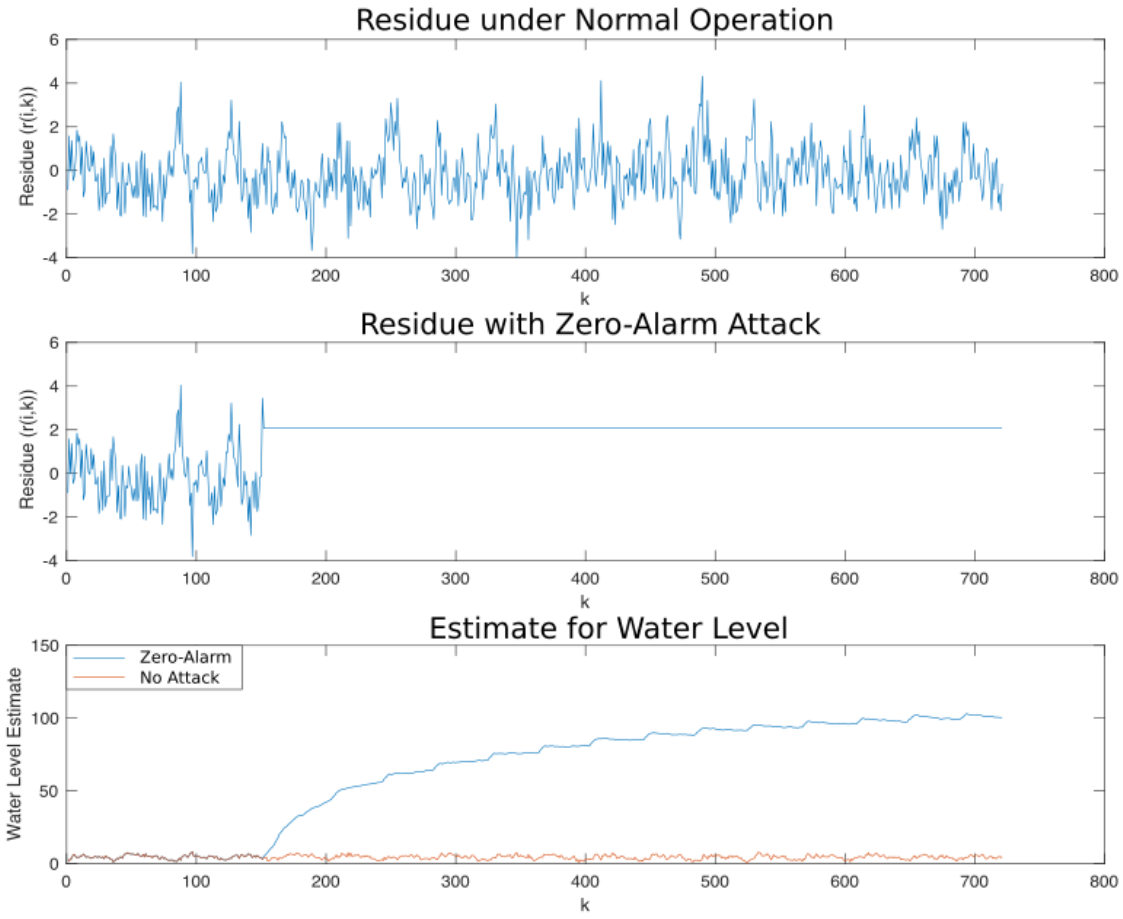


Figure 7: CUSUM detection method under a zero-alarm attack for scenario 1.

the statistics of the residual may change under some types of attacks (e.g., bias injection attacks). However, by construction, the zero-alarm attacks introduced in the last section are not detected by the bad-data and CUSUM procedures. We carry out simulations for two sets of data (labeled as scenario 1 and scenario 2) generated by EPANET for the water distribution network depicted in Figure 1. These scenarios are described as follows.

- Scenario 1: In this case, we collect data from EPANET under *normal operating conditions* (no attacks). Based on this data, the matrices (A, B, C) of model (1) are obtained using subspace identification techniques (presented in the appendix). In this scenario, the matrix A satisfies $\rho(A) < 1$. To study the effect of attacks, we manipulate the sensor measurement data coming from EPANET. In particular, we add a *constant bias* to the sensor measurements. We also induce the zero-alarm attacks introduced in the previous section.
- Scenario 2: Here, data is collected from EPANET not only under normal operating conditions but also under control input attacks, i.e., we get a set of "healthy" data and another under attacks. We use the healthy data to generate new system matrices (A, B, C). In this case, the obtained matrix A satisfies $\rho(A) > 1$, i.e., the system model is open-loop unstable. We remark that this case considers input attacks. We have no information about the starting time and the structure of these control input attacks. It makes this scenario interesting for testing the effectiveness of our detection schemes.

Figure 6 shows the evolution of the residuals under the zero-alarm attack for scenario 1. The top plot depicts the residual for the level sensor reading of the storage tank when the system is running under normal conditions. Each value of k represents sampling time from the simulations in EPANET (sampling interval in this case study is 15 minutes). The zero-alarm attack for the bad-data detector is designed as in (18). The attack is launched at $k = 150$, so that the first 150 data samples are attack free. This helps us to understand the evolution of the residuals before and during the attack. After the attack is launched, the residual approaches the threshold $\alpha_{k,i}$ (as seen in Figure 6); and stays there for the rest of the simulation (because attack is not removed). This zero-alarm attack leads to deviations in the state estimates and sensor measurements (as shown in

	Alarm Rate Output 1	Alarm Rate Output 2	Alarm Rate Output 3	Alarm Rate Output 4	Alarm Rate Output 5
Bad-Data: no attack	0.0194	0.0153	0.0125	0.0139	0.0125
Bad-Data: bias attack	0.7947	0.7920	0.7933	0.7933	0.7947
CUSUM: no attack	0.0208	0.0180	0.0153	0.0166	0.0153
CUSUM: bias attack	0.7947	0.7947	0.7947	0.7933	0.7947
Bad-Data: zero-alarm attack	0.0042	0.0042	0.0014	0.0014	0.0014
CUSUM: zero-alarm attack	0.0083	0.0055	0.0069	0.0055	0.0069

Table 1: Alarm Rates for given conditions for Scenario 1.

	Alarm Rate Output 1	Alarm Rate Output 2	Alarm Rate Output 3	Alarm Rate Output 4	Alarm Rate Output 5
Bad-Data: no attack	0.0447	0.0385	0.0520	0.0510	0.0229
Bad-Data: attack at input	0.1686	0.1374	0.1613	0.1301	0.1811
CUSUM: no attack	0.0527	0.0430	0.0527	0.0583	0.0222
CUSUM: attack at input	0.2094	0.1650	0.2025	0.1442	0.1650
Bad-Data: zero-alarm attack	0.0021	0.0031	0.0052	0.0031	0
CUSUM: zero-alarm attack	0.0042	0.0111	0.0111	0.0014	0.0139

Table 2: Alarm Rates for given conditions for Scenario 2.

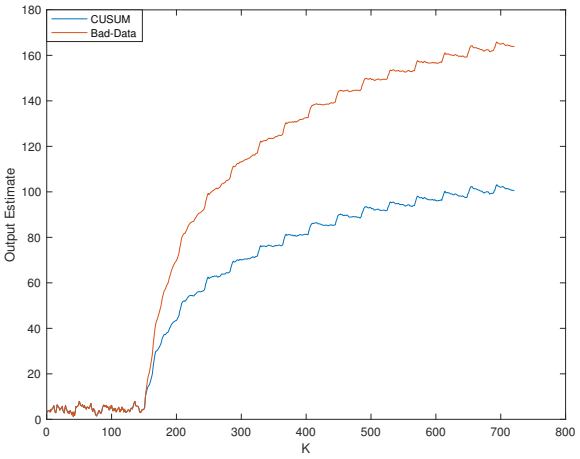


Figure 8: Degradation of $\hat{y}(k)$ due to a zero-alarm attack for scenario 1.

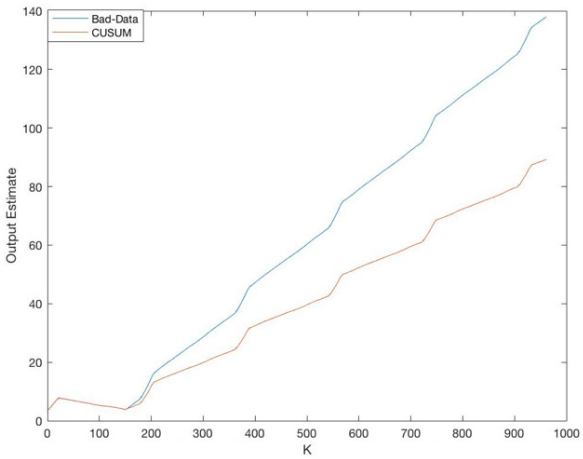


Figure 9: Degradation of $\hat{y}(k)$ due to a zero-alarm attack for scenario 2.

Figure 6). Figure 7 depicts similar results for the CUSUM procedure under the corresponding zero-alarm attack. Here, we note that the residual under the zero-alarm attack converges to b_i after the attack starts. For both procedures, under zero-alarm attacks, note that (Figures 6 and 7) the state estimate converges to steady state, i.e., it remains bounded. This is expected in scenario 1 because $\rho(A) < 1$; and hence, by Proposition 1 and Proposition 2, the estimation error is bounded which implies boundedness of the state estimate because the state of the model is bounded itself. Figure 8 compares the output estimate degradation between the bad-data and CUSUM procedures under zero-alarm attacks. It is evident from this figure that the CUSUM performs better than bad-data procedure in terms of estimation deviation– which is also in accordance with our analytic results in section 4.2.1.

Figure 10 shows the performance of the bad-data procedure under zero-alarm attacks for scenario 2. In this case, we have similar results as in scenario 1. However, we note that the output estimate diverges under zero-alarm attacks. This is because the matrix A satisfies $\rho(A) > 1$ which implies that $||E[e_k]||$ diverges. Similar results are obtained for the CUSUM (see Figure 11). Figure 9 compares the output estimate degradation between the bad-data and CUSUM procedures under zero-alarm attacks. Note that although both estimates diverge, the CUSUM leads to slower divergence rate than the bad-data procedure.

In Tables 1 and 2, the alarm rates for scenario 1 and scenario 2 are presented respectively. In both tables, each column shows the rate of alarms produced by sensor measurement for both detectors. Outputs 1-4, represent the pressure sensors at the four consumer nodes (Node 4-7). Out-

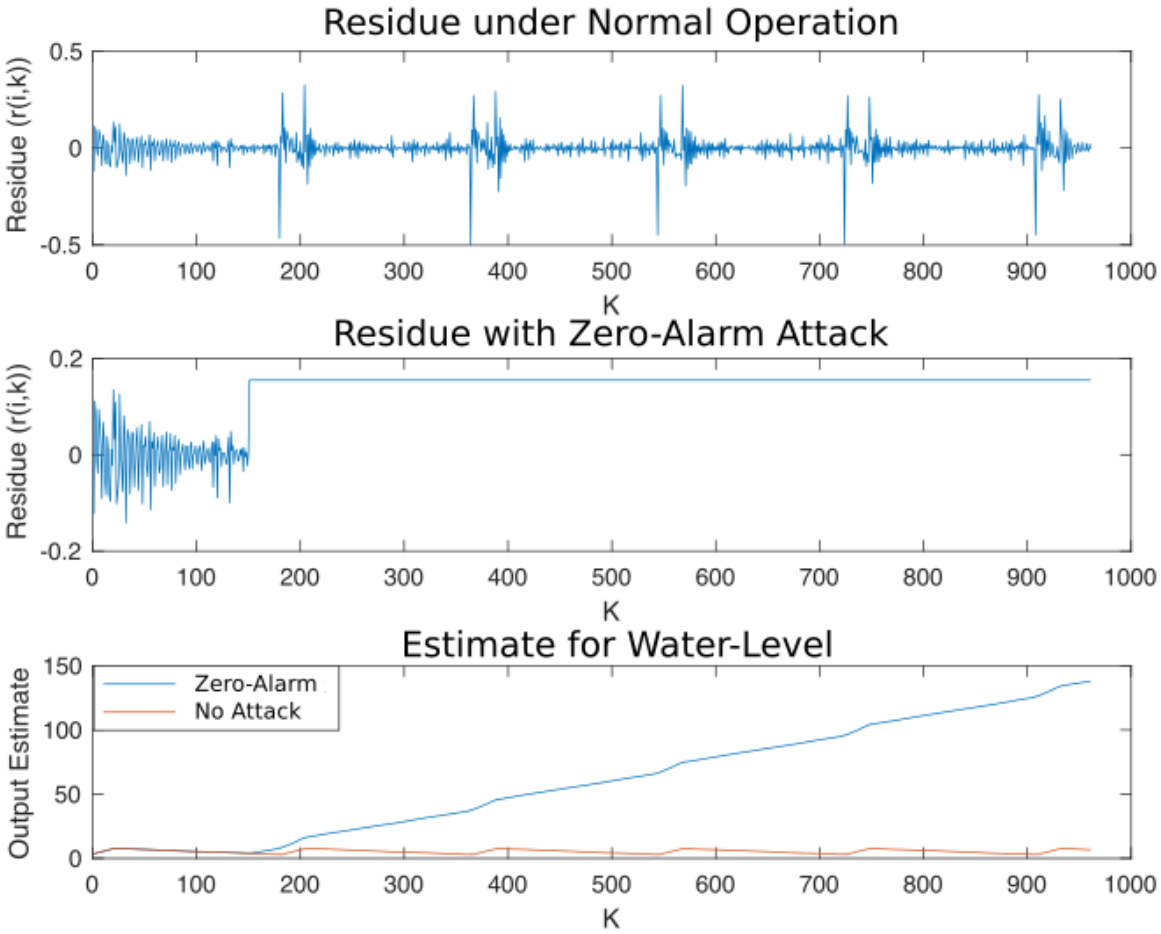


Figure 10: Bad-data detection method under a zero-alarm attack for scenario 2.

put 5 is the level sensor in the storage tank. Each row label in Table 1, presents a key1:key2 pair where key1 is detection method and key2 is attack scenario, e.g., bad-data:bias attack, presents results for bad-data detector under a bias attack. In scenario 1 (Table 1), the attack is referred to addition of a bias value in the output measurement from the level sensor. This attack starts at time slot $k = 150$. In the no attack scenario we see the performance of the bad-data and CUSUM detectors as expected, i.e., alarm rate (for attack) is equal to the false alarm rate of 1%. In the bias attack case of scenario 1, the alarm rate goes up to 79% which implies that this attack is easily detected based on detector alarms. Considering that the first 20% of the measurements correspond to normal operation (for visualizing the effects of the attack), the rest of the attacked readings result in an alarm showing that such a bias attack is easily found by these detection schemes. For the zero-alarm attack, results show very small alarm rate as these attacks are designed not to raise alarms. For zero-alarm attacks, if we start attack from beginning of the measurements, we get 0% alarm rate. For scenario 2 (Table 2), we observe that when the system is under attack the alarm rate is higher than compared to normal operation. Since, the attack in this case is on control inputs and we do not know when the attack starts and finishes. Thresholds are calculated for 1% false alarm rate but for attacks on input, the alarm rate reached 20%, which is much higher than the normal false alarm rate. So, we can point out that the system is under attack. This also points to the fact that if the system is being attacked at inputs, such attacks can be detected by using output measurements of the system. The zero-alarm attacks are created for sensor attacks, not actuator attacks, so it is not surprising that we are able to detect when attacks are initiated on the inputs.

6. CONCLUSION

In this manuscript, for the model of a water distribution network, we have explained step by step how to construct model-based attack detectors for identifying compromised sensors and actuators. In particular, a Kalman filter has been proposed to estimate the state of the physical process; then, these estimates have been used to construct residual variables (difference between sensor measurements and estimations) which drive the CUSUM procedure. For a class of zero-alarm attacks, we have characterized the performance

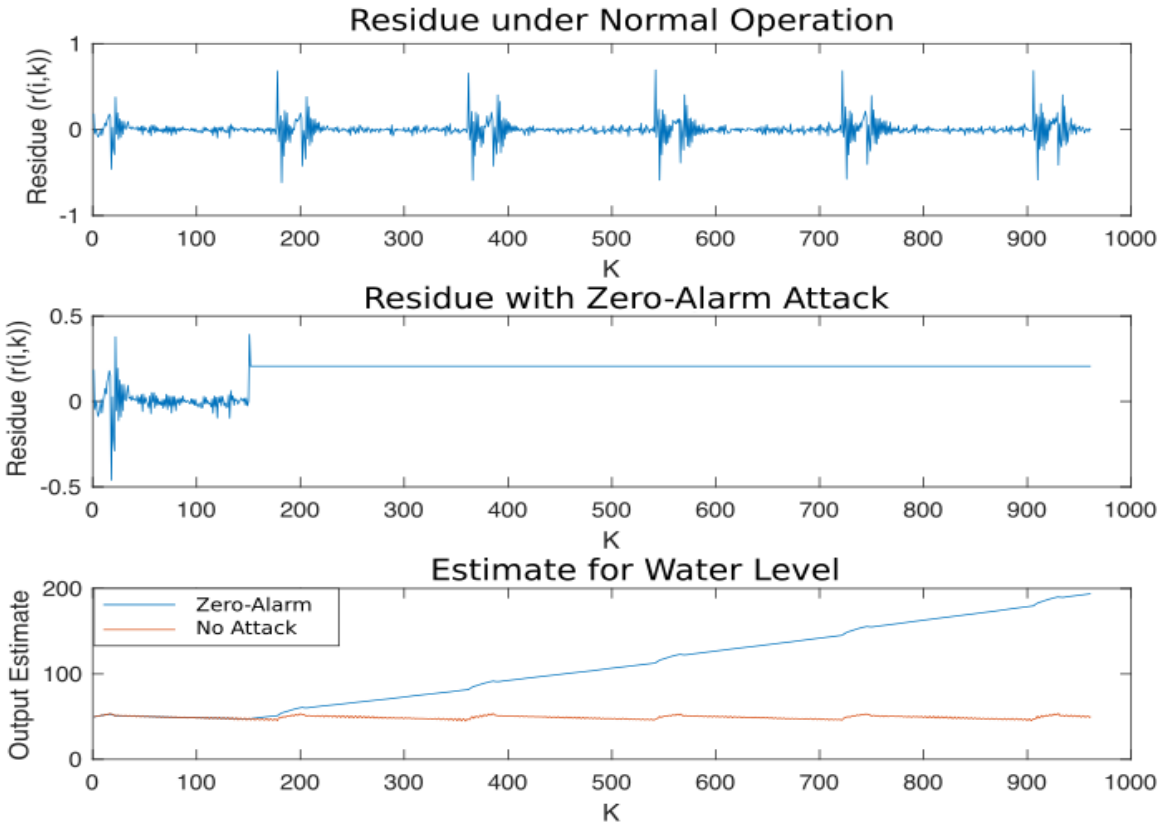


Figure 11: CUSUM detection method under a zero-alarm attack for scenario 2.

of the proposed detection procedures in terms of the effect that the attack sequence can induce on the system dynamics, namely in the output estimate. Then, we have compared performance of CUSUM and Bad-Data method against each other. We have shown how bias attacks (and most probably any output-injection attack as well) are easily detected using fault-detection techniques as long as the statistics of the residuals (in the attack-free case) are well characterized. Moreover, input-injection attacks are also detected easily using the proposed methods. Numerical simulations show the effectiveness of the proposed methods against different classes of attacks.

7. ACKNOWLEDGMENTS

This work was supported by the National Research Foundation (NRF), Prime Minister's Office, Singapore, under its National Cyber Security R&D Programme (Award No. NRF2014NCR-NCR001-40) and administered by the National Cybersecurity R&D Directorate.

8. REFERENCES

[1] EPANET: software that models the hydraulic and water quality behavior of water distribution piping systems. https://www.epa.gov/water-research/epanet. Accessed: 2016-03-29.

[2] I. C. 2014. Ics-mm201408: May-august 2014. *Report no., U.S. Department of Homeland Security-Industrial Control Systems-Cyber Emergency Response Team, Washington, D.C. Available online at https://ics-cert.us-cert.gov.*, 2014.

[3] B. Adams, W. Woodall, and C. Lowry. The use (and misuse) of false alarm probabilities in control chart design. *Frontiers in Statistical Quality Control 4*, pages 155–168, 1992.

[4] C. M. Ahmed, A.Sridhar, and M. Aditya. Limitations of state estimation based cyber attack detection schemes in industrial control systems. In *IEEE Smart City Security and Privacy Workshop, CPSWeek*, 2016.

[5] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen. Cyber security of water scada systems-part i: analysis and experimentation of stealthy deception attacks. *IEEE Transactions on Systems Technology*, pages 1963–1970, 2013a.

[6] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen. Cyber security of water scada systems-part ii: Attack detection using enhanced hydrodynamic models. *IEEE Transactions on Systems Technology*, pages 1679–1693, 2013b.

[7] K. J. Aström and B. Wittenmark. *Computer-controlled Systems (3rd Ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1997.

[8] A. Bobat, T. Gezgin, and H. Aslan. The scada system applications in management of yuvacik dam and reservoir. *Desalination and Water Treatment*, 2015.

[9] A. Cardenas, S. Amin, Z. Lin, Y. Huang, C. Huang, and S. Sastry. Attacks against process control systems: Risk assessment, detection, and response. In *6th ACM Symposium on Information, Computer and Communications Security*, pages 355–366, 2011.

[10] J. Giraldo, A. Cardenas, and N. Quijano. Integrity attacks on realtime pricing in smart grids: Impact and countermeasures. *IEEE Transactions on Smart Grid*, 2016.

[11] Y. Gu, T. Liu, D. Wang, X. Guan, and Z. Xu. Bad data detection method for smart grids based on distributed estimation. In *IEEE ICC*, 2013.

[12] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, New York, NY, USA, 2nd edition, 2012.

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

[14] E. A. Lee. Cyber physical systems: Design challenges. In *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2008-8*. http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-8.html, Jan. 2008.

[15] N. Lehtinen. Error functions. *Stanford University,Webpage:http://nlpc.stanford.edu/nleht/Science/reference/errorfun.pdf*, April 2010.

[16] F. Miao, Q. Zhu, M. Pajic, and G. J. Pappas. Coding sensor outputs for injection attacks detection. In *IEEE conference in Decision and Control (CDC)*, pages 5776–5781, 2014.

[17] L. Mili, T. Cutsen, and M.R.-Pavella. Bad data identification methods in power system state estimation - a comparative study. *IEEE Trans. on Power Apparatus and Systems*, 1985.

[18] Y. Mo, E. Garone, A. Casavola, and B. Sinopoli. False data injection attacks against state estimation in wireless sensor networks. In *IEEE conference in Decision and Control (CDC)*, pages 5967–5972, 2010.

[19] C. Murguia and J. Ruths. Characterization of a cusum model-based sensor attack detector. In *55th IEEE Conference on Decision and Control Conference (CDC)*, 2016.

[20] C. Murguia and J. Ruths. Cusum and chi-squared attack detection of compromised sensors. In *2016 IEEE Conference on Control Applications (CCA)*, pages 474–480, Sept 2016.

[21] P. V. Overschee and B. D. Moor. Subspace identification for linear systems: theory, implementation, applications. *Boston: Kluwer Academic Publications*, 1996.

[22] E. Page. Continuous inspection schemes. *Biometrika*, 41:100–115, 1954.

[23] L. Perelman and S. Amin. A network interdiction model for analyzing the vulnerability of water distribution systems. In *Proceedings of the 3rd international conference on High confidence networked systems, ACM.*, pages 135–144, 2014.

[24] M. Ross. *Introduction to Probability Models, Ninth Edition*. Academic Press, Inc., Orlando, FL, USA, 2006.

[25] J. Slay and M. Miller. Lessons learned from the maroochy water breach. *Springer 620 US, Boston, MA*, pages 73–82, 2008.

[26] A. Sridhar and M. Aditya. Generalized attacker and attack models for cyber physical systems. In *40th IEEE COMPSAC*, 2016.

[27] C. van Dobben de Bruyn. *Cumulative sum tests : theory and practice*. London : Griffin, 1968.

APPENDIX

A. STATE SPACE MATRICES FOR SCENARIO-2 AND SCENARIO-1

In what follows, we present the state space matrices (A, B, C), obtained using sub-space system identification. For scenario-2, more than 70% accuracy is achieved by a 10th order model and for scenario-1 by a 20th order model. Therefore, we have system matrix A_2 as a 10 x 10 and A_1 as a 20 x 20. For a 6 inputs (4 user demands, flow at the pumping station, ON/OFF status of the pumping station), matrix dimensions for B_2 are 10 x 6 and for B_1 are 20 x 6. For 5 outputs, the dimensions for matrix C_2 are, 5 x 10 and for C_1 are, 5 x 20. Using these state space matrices and system model of (1), one can find the dynamics of the system evolution.

$$\mathbf{A_2} = \begin{pmatrix} 0.9774 & -0.0190 & -0.0747 & 0.0246 & 0.0154 & -0.0717 & -0.1313 & -0.0721 & 0.0649 & 0.1173 \\ -0.0054 & 0.9184 & -0.3095 & 0.0309 & -0.2480 & -0.0283 & -0.0595 & -0.0388 & 0.0565 & 0.0813 \\ -0.0196 & -0.2107 & -0.6614 & -0.6767 & -0.0330 & 0.0107 & 0.0137 & 0.0321 & 0.1525 & 0.0392 \\ -0.0262 & 0.2753 & 0.0299 & -0.0964 & 0.8360 & -0.1407 & -0.1406 & -0.0304 & 0.1403 & -0.0136 \\ 0.0095 & 0.1986 & 0.7830 & -0.7442 & -0.0968 & -0.0499 & 0.0277 & -0.0121 & 0.0126 & 0.0426 \\ 0.0716 & -0.0264 & 0.1303 & 0.1686 & 0.0778 & 0.1560 & -0.1839 & -0.1363 & 0.1427 & 0.7906 \\ 0.1279 & 0.0876 & 0.0159 & 0.0632 & 0.0633 & -0.2296 & 0.9815 & 0.0160 & 0.2045 & -0.1912 \\ 0.0095 & -0.0425 & -0.0630 & 0.0937 & 0.0746 & -0.2460 & -0.0816 & 0.9258 & -0.0314 & 0.3342 \\ -0.1307 & -0.2171 & -0.0718 & 0.1451 & -0.0927 & -0.6864 & -0.4954 & -0.2661 & 0.8537 & 0.1556 \\ -0.0174 & -0.0135 & -0.0614 & 0.0132 & -0.0248 & -0.3059 & -0.3387 & 0.0797 & 0.5341 & -0.8659 \end{pmatrix}$$

$$\mathbf{B_2} = \begin{pmatrix} 0.007572 & -0.05175 & 0.108 & 0.02162 & -0.2539 & 160 \\ 0.01116 & -0.039 & 0.07715 & 0.01197 & -0.2052 & 129.1 \\ 0.2299 & 0.144 & 0.2696 & 0.2064 & -0.2294 & 136.5 \\ 0.1663 & 0.1449 & 0.1711 & 0.1732 & 0.02358 & -20.95 \\ -0.06869 & -0.09574 & -0.08144 & -0.0957 & -0.1105 & 72.67 \\ 0.03138 & -0.4238 & 0.8555 & 0.1249 & -2.411 & 1538 \\ -0.01166 & 0.04002 & -0.08159 & -0.01465 & 0.2355 & -145.4 \\ 0.02767 & -0.1526 & 0.3593 & 0.06446 & -0.99 & 630 \\ 0.07585 & -0.1316 & 0.5653 & 0.1126 & -1.748 & 1129 \\ -0.0259 & 0.6811 & -1.134 & -0.1565 & 3.043 & -1904 \end{pmatrix}$$

$$\mathbf{C_2} = \begin{pmatrix} 114.6361 & -161.3788 & 31.0325 & -30.3654 & 13.6921 & -3.1363 & 17.3276 & -11.2433 & 4.0045 & -3.4214 \\ 114.5909 & -161.3954 & 31.0365 & -30.3679 & 13.6918 & -3.1178 & 17.2831 & -11.2558 & 4.0017 & -3.4165 \\ 164.6299 & -232.2755 & 44.4661 & -43.7358 & 20.5170 & -6.4195 & 22.5551 & -2.8888 & 3.3832 & -3.3145 \\ 144.6166 & -203.9462 & 39.0985 & -38.3926 & 17.7888 & -5.0984 & 20.4400 & -6.2367 & 3.6315 & -3.3543 \\ 14.3219 & -19.5879 & 4.1716 & -3.6204 & 0.0279 & 3.5410 & 6.5744 & -28.0264 & 5.2328 & -3.5966 \end{pmatrix}$$

$$\mathbf{A_1} = \begin{pmatrix} 0.19 & 0.09 & -0.01 & -0.08 & -0.05 & -0.01 & -0.18 & 0.13 & 0.27 & -0.20 & 0.29 & -0.27 & 0.08 & 0.12 & -0.30 & -0.04 & 0.22 & -0.21 & 0.43 & -0.03 \\ -0.04 & 0.84 & 0.01 & -0.13 & -0.37 & -0.14 & -0.10 & -0.08 & 0.04 & -0.36 & 0.07 & 0.42 & 0.10 & 0.81 & -0.34 & -0.36 & -0.92 & 0.31 & 0.30 & -0.71 \\ -0.01 & -0.15 & 0.95 & -0.04 & -0.08 & -0.00 & 0.21 & -0.01 & 0.01 & -0.02 & -0.15 & 0.12 & -0.17 & 0.41 & -0.01 & 0.08 & -0.10 & -0.11 & -0.04 & 0.13 \\ -0.57 & 0.27 & -0.03 & 0.67 & -0.00 & 0.11 & -0.07 & -0.47 & -0.37 & 0.57 & -0.07 & 0.19 & -0.04 & -0.36 & 0.10 & 0.18 & 0.84 & 0.48 & -0.51 & 0.16 \\ 0.08 & -0.01 & 0.08 & -0.19 & -0.01 & -0.17 & 0.26 & -0.05 & -0.06 & -0.31 & -0.12 & 0.07 & 0.08 & 0.69 & 0.03 & -0.07 & -0.93 & -0.44 & 0.03 & -0.38 \\ 0.30 & 0.40 & 0.03 & 0.00 & 0.11 & 0.96 & -0.43 & 0.19 & -0.27 & 0.08 & 0.40 & 0.01 & 0.33 & -0.84 & -0.15 & -0.64 & 0.19 & 0.83 & -0.03 & -0.77 \\ 0.14 & 0.14 & -0.00 & -0.03 & -0.25 & -0.00 & 0.91 & -0.27 & -0.06 & -0.28 & 0.04 & -0.02 & -0.06 & 0.10 & -0.14 & -0.32 & -0.28 & 0.03 & 0.19 & -0.07 \\ 0.13 & -0.04 & 0.01 & 0.09 & 0.01 & -0.07 & 0.30 & 0.82 & -0.18 & -0.09 & 0.11 & 0.09 & 0.06 & 0.01 & 0.12 & -0.00 & -0.00 & 0.10 & -0.11 & -0.12 \\ 0.11 & 0.23 & -0.00 & -0.02 & -0.28 & 0.07 & 0.08 & 0.24 & 0.49 & -0.27 & -0.16 & -0.20 & 0.05 & -0.45 & -0.05 & -0.26 & -0.16 & 0.13 & 0.06 & 0.14 \\ 0.20 & 0.47 & 0.12 & -0.11 & -0.53 & -0.24 & 0.11 & 0.25 & 0.73 & 0.39 & 0.17 & -0.05 & 0.09 & 0.12 & 0.07 & -0.12 & -0.17 & -0.04 & 0.13 & 0.19 \\ -0.37 & -0.12 & 0.04 & -0.15 & 0.24 & -0.09 & -0.18 & -0.31 & 0.68 & -0.02 & 0.79 & 0.68 & 0.05 & 0.39 & -0.06 & 0.52 & 0.24 & -0.20 & 0.18 & 0.15 \\ 0.10 & -0.10 & -0.02 & 0.07 & 0.34 & 0.03 & -0.10 & -0.19 & -0.16 & 0.14 & -0.31 & 0.61 & 0.53 & 0.19 & 0.07 & 0.27 & 0.42 & 0.01 & 0.08 & 0.00 \\ -0.24 & -0.20 & -0.06 & 0.03 & 0.13 & 0.07 & 0.04 & -0.10 & -0.08 & 0.01 & -0.14 & -0.35 & 0.40 & 0.23 & -0.21 & 0.65 & 0.15 & 0.12 & -0.01 & 0.23 \\ -0.19 & 0.03 & -0.01 & 0.02 & 0.01 & 0.04 & -0.04 & -0.01 & 0.19 & -0.04 & 0.09 & -0.06 & -0.38 & 0.59 & 0.42 & 0.21 & 0.14 & 0.07 & 0.20 & -0.12 \\ 0.19 & -0.06 & 0.11 & 0.15 & 0.03 & -0.01 & 0.11 & -0.09 & 0.18 & -0.05 & 0.16 & -0.02 & -0.05 & -0.40 & -0.03 & 0.63 & -0.29 & -0.03 & 0.01 & 0.02 \\ -0.38 & -0.06 & -0.05 & 0.00 & 0.15 & 0.08 & 0.01 & 0.13 & -0.04 & 0.23 & -0.03 & 0.27 & -0.18 & -0.04 & -0.52 & -0.00 & 0.06 & -0.07 & 0.28 & -0.36 \\ 0.18 & -0.08 & -0.08 & 0.11 & 0.36 & 0.18 & -0.23 & -0.10 & 0.16 & 0.01 & 0.29 & -0.25 & 0.14 & -0.01 & -0.31 & -0.17 & 0.36 & 0.46 & -0.13 & 0.33 \\ 0.00 & -0.21 & -0.02 & 0.18 & 0.17 & 0.12 & 0.00 & -0.03 & -0.07 & 0.13 & -0.08 & -0.01 & -0.05 & -0.15 & 0.19 & -0.08 & -0.18 & 0.05 & 0.56 & 0.55 \\ -0.16 & -0.06 & -0.02 & 0.11 & -0.03 & 0.07 & -0.10 & -0.01 & 0.11 & 0.05 & 0.10 & -0.03 & 0.14 & 0.01 & -0.04 & -0.14 & 0.10 & -0.64 & -0.27 & 0.52 \\ 0.29 & -0.27 & 0.04 & 0.09 & -0.03 & -0.15 & -0.04 & -0.07 & 0.14 & -0.03 & -0.01 & -0.04 & -0.08 & 0.14 & -0.08 & 0.11 & 0.07 & -0.35 & -0.08 & -0.34 \end{pmatrix}$$

$$\mathbf{B_1} = \begin{pmatrix} -0.0049 & -0.0026 & -0.0070 & 0.0009 & 0.0268 & -19.9005 \\ 0.0047 & -0.0040 & -0.0008 & 0.0243 & 0.0923 & -67.4258 \\ 0.0077 & 0.0094 & 0.0126 & 0.0110 & -0.0043 & 3.3105 \\ -0.0215 & -0.0351 & -0.0131 & -0.0217 & -0.0362 & 26.1488 \\ 0.0086 & 0.0105 & 0.0231 & 0.0300 & 0.0690 & -50.4232 \\ -0.0128 & -0.0306 & -0.0363 & -0.0232 & 0.0032 & -3.1391 \\ 0.0062 & 0.0021 & -0.0015 & 0.0088 & 0.0052 & -3.5669 \\ 0.0103 & 0.0052 & -0.0043 & 0.0128 & -0.0036 & 3.3107 \\ 0.0175 & 0.0111 & -0.0058 & 0.0237 & -0.0075 & 5.9806 \\ -0.0122 & 0.0118 & 0.0038 & -0.0060 & 0.0278 & -20.7621 \\ -0.0044 & -0.0018 & 0.0019 & -0.0079 & -0.0002 & -0.2009 \\ -0.0094 & -0.0212 & -0.0079 & -0.0123 & -0.0060 & 3.4808 \\ -0.0018 & 0.0109 & 0.0133 & 0.0045 & -0.0113 & 8.8180 \\ 0.0081 & 0.0198 & 0.0154 & 0.0146 & 0.0157 & -10.5528 \\ 0.0036 & -0.0196 & -0.0142 & -0.0146 & -0.0384 & 27.2594 \\ -0.0089 & -0.0274 & -0.0225 & -0.0240 & 0.0185 & -14.5213 \\ -0.0047 & -0.0035 & -0.0015 & -0.0050 & -0.0109 & 8.2116 \\ 0.0217 & 0.0182 & 0.0118 & 0.0072 & -0.0132 & 9.5218 \\ -0.0034 & -0.0012 & -0.0025 & -0.0095 & -0.0045 & 3.0176 \\ 0.0045 & 0.0073 & 0.0070 & -0.0036 & -0.0169 & 12.7814 \end{pmatrix}$$

$$\mathbf{C_1} = \begin{pmatrix} 0.202 & 2.873 & -2.1917 & 2.3322 & -1.9137 & -2.1566 & -2.2838 & 0.330 & 1.272 & 2.0943 & -1.4317 & 0.8167 & -1.0703 & 0.574 & 0.001 & -0.258 & 1.1656 & 1.6621 & 0.0410 & 1.8834 \\ 0.148 & 2.855 & -2.1916 & 2.3308 & -1.9153 & -2.1490 & -2.2641 & 0.315 & 1.278 & 2.0990 & -1.4257 & 0.8023 & -1.0441 & 0.5535 & 0.002 & -0.251 & 1.1502 & 1.6658 & 0.0366 & 1.8753 \\ -0.003 & 4.231 & -2.0163 & 3.2396 & -2.8943 & -2.6020 & -2.9784 & 0.143 & 1.614 & 3.0032 & -2.1411 & 1.0252 & -1.1644 & 0.6702 & 0.169 & -0.467 & 1.1799 & 2.5161 & 0.1678 & 2.4226 \\ 0.069 & 3.679 & -2.0885 & 2.8690 & -2.5015 & -2.4194 & -2.6947 & 0.215 & 1.479 & 2.6415 & -1.8544 & 0.9364 & -1.1212 & 0.6277 & 0.101 & -0.383 & 1.1741 & 2.1719 & 0.1169 & 2.2032 \\ 0.288 & 0.081 & -2.5399 & 0.5308 & 0.0734 & -1.2121 & -0.7473 & 0.600 & 0.627 & 0.2954 & 0.0208 & 0.3193 & -0.7330 & 0.2522 & -0.332 & 0.220 & 1.0221 & -0.0277 & -0.2465 & 0.7559 \end{pmatrix}$$