Physics-Based Methods for Distinguishing Attacks from Faults

Gregory Provan Riccardo Orizio

School of Computer Science
University College Cork
Cork, Ireland
Email: g.provan, r.orizio@cs.ucc.ie

***Abstract*—Cyber-physical systems (CPSs) are a key framework for analysing a range of systems, from power plants to automobiles. One recent trend has been using this framework for security analysis. This article uses physics-based methods for distinguishing attacks from faults. We frame a CPS as a discrete-time linear system that can switch between various modes. By encoding faults and attacks each as specific modes, we build CPS models that incorporate the impact of a range of types of fault and attack. We then use this CPS model to isolate (and distinguish between) a fault and an attack. We illustrate our approach on a hydraulic benchmark system.**

***Keywords*–model-based security; model-based diagnosis; state identification.**

I. Introduction

The study of Cyber-Physical Systems (CPSs) [1] is attracting great interest, due to the significance of the applications that a CPS can model. For example, CPSs can model nuclear power plants, air-traffic control systems, smart cities, etc.

Recently, researchers have been focusing on identifying and defending against attacks on a CPS, e.g., [2], [3], [4], [5]. A broad range of approaches have been used for attack modeling and detection, none of which is fully comprehensive in terms of the range of attacks that can be identified [2], [3].

This article focuses on using physics-based models to isolate attacks on a system. We assume that a CPS is an instance of a hybrid system, in that the system can operate in a variety of distinct behaviours, which we call modes. For example, an aircraft can be in take-off or cruise mode, or it can operate in one of several faulty modes. We use system mode identification approaches [6], together with appropriate attack models, to compute an attack on a system.

In our approach, we create a first-principles physics-based model of the CPS and its control system. We explicitly create modes depicting the impact of faults on the CPS. We assume that an attacker may inject data into the CPS to mimic faults that occur naturally. As a consequence, we also include physics-based attack models.

Our objective is to analyze which faults can be distinguished from attacks using limited sensors in the CPS (most real-world systems have limited sensors available). This analysis enables us to understand the strengths and limitations of physics-based CPS attack analysis.

Our contributions are as follows:

- We describe an observer-based framework for isolating faults and attacks, and a method for distinguishing between them;
- We show that physics-based methods can distinguish attacks on sensors from sensor faults, but that actuator attacks cannot be distinguished from actuator faults;
- We illustrate our approach on a well-known hydraulic benchmark.

We organize the paper as follows. We introduce a running example in Section II. Next, Sections III and IV present the formal framework for our work. We present our empirical studies in Section V, and summarize our results in Section VII.

II. Running Example

We illustrate our concepts using a three-tank system, as shown in figure 1.

A. Nominal Model

We denote the tanks as T_1, T_2, and T_3. They all have the same area $A_1 = A_2 = A_3 = 3$ [m^2]. We assume that $g = 10$ and the liquid is "pure" water with density $\rho = 1$.

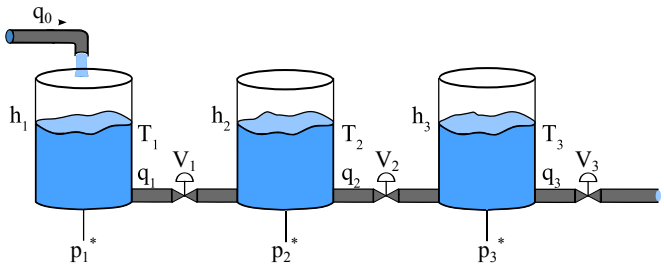

Figure 1. Diagram of the three-tank system.

Tank T_1 is filled from a pipe q_0 with a constant flow of 0.75 [m^3/s]. It drains into T_2 via a pipe q_1. The liquid level is denoted as h_1. There is a pressure sensor p_1 connected to T_1 that measures the pressure in Pascals [Pa]. The system has valves V_1, V_2, V_3 as shown in figure 1.

For this system we control the inflow q_0 and valve positions, i.e., our input vector $u = \{q_0, V_1, V_2, V_3\}$. We can measure the tank pressure values, i.e., the measurement

 ISBN: 978-1-61208-585-2

vector is $y = \{p_1, p_2, p_3\}$. Our control task is to maintain set-point heights in each of the tanks. The diagnostic task is to compute the true value of V_i, given p_i, for $i = 1, 2, 3$.

We define our nominal model as follows. Starting from Newton's (and Bernoulli's) equations and manipulating them (the actual derivation is irrelevant in this paper) we derive the following Ordinary Differential Equation (ODE) that gives the level of the liquid in T_1:

$$\frac{dh_1}{dt} = q_0 - q_1 \;=\; \frac{q_0 - k_1 s(h_1, h_2)\sqrt{|h_1 - h_2|}}{A_1}, \qquad (1)$$

where $s(h_1, h_2)$ denotes $\text{sign}(h_1 - h_2)$. In eq. 1, the coefficient k_1 is given by $k_1 = \nu_1 \kappa_1$, which is the product of the valve V_1 setting, $\nu_1 \in [0, 1]$, where 0 denotes a closed valve and 1 an open valve, and the outflow parameters κ_1, which include the cross-sectional area of the tank A_1, the area of the drainage hole, $\sqrt{2g}$, and the friction/contraction factor of the hole. We emphasize the use of k_1 because, later, we will be "diagnosing" our system in term of changes in k_1. Consider a physical valve V_1 between T_1 and T_2 that constrains the flow between the two tanks. We can say that the valve changes proportionally to the cross-sectional drainage area of q_1 and hence k_1.

We define the water levels of T_2 and T_3, denoted as h_2 and h_3 respectively, as:

$$\frac{dh_i}{dt} = \frac{k_{i-1} s(h_{i-1}, h_i)\sqrt{|h_{i-1} - h_i|} - k_i\sqrt{h_i}}{A_i}, \qquad (2)$$

where i is the tank index ($i \in \{2, 3\}$).

We assume that $\kappa_1 = \kappa_2 = \kappa_3 = 0.75$.

Finally, we can compute from the water level a pressure given by

$$p_i = \frac{g\, h_i\, A}{A} = g\, h_i \qquad (3)$$

where i is the tank index ($i \in \{1, 2, 3\}$).

We assume that the initial water level in the three tanks is zero.

B. Fault Model

In the following we define valve (actuator) faults; other faults, e.g., leaks or sensor faults, can be defined analogously.

We assume an additive valve fault, where the actual valve position for valve i, given commanded position ν_i and fault Δ_{ν_i}, is

$$\nu_i = \begin{cases} \max\{0, \nu_i + \Delta_{\nu_i}\} & \text{if } \Delta_{\nu_i} \leq 0 \\ \min\{1, \nu_i + \Delta_{\nu_i}\} & \text{if } \Delta_{\nu_i} > 0 \end{cases} \qquad (4)$$

where $\Delta_{\nu_i} \in [-1, 1]$.

C. Attack Model

For our attack model, we assume that an attacker cannot monitor the system, but can inject false data.

We first consider injecting a fake sensor reading. Hence, for pressure sensor p_i ($i = 1, 2, 3$), which can output nominal values in the range $[0, p_i^{max}]$, an attacker can inject a fixed value of $p_i^a \in [0, p_i^{max}]$.

If an attacker injects a fake actuator value $\nu_i \in [0, 1]$ ($i = 1, 2, 3$), then valve i will be commanded to this "fake" position.

There is a difference in the physical behaviours of these two attacks. Whereas the actuator attack alters the system itself, the sensor attack has no impact on the physical behaviour unless the control system changes in response to the fake sensor value.

III. Cyber-Physical Systems with Faults and Attacks

This section provides the theoretical basis for our models and attack detection procedures. We first define a discrete-time state-space model for a Cyber-Physical System (CPS) that is subject to faults and attacks.

A. Cyber-Physical Systems

The nominal (or ideal) system model is given by

$$\begin{aligned} x_{k+1} &= A_\gamma x_k + B_\gamma u_k + w_k; \\ y_k &= C_\gamma x_k + v_k; \end{aligned} \qquad (5)$$

where $x_k \in \mathbb{R}^n$ is the state of the system, $x_0 \in \mathbb{R}^n$ the initial state of the system, $u_k \in \mathbb{R}^l$ the control input, and $y_k \in \mathbb{R}^p$ the measurement at time instance k. We assume that a system can operate in a mode $\gamma_i \in \Gamma$. Each mode determines the physical behaviours of CPS. We capture the mode using a matrix with subscript γ, e.g., A_γ. The unknown process and measurement noise are $w_k \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^p$, respectively. We define our matrices as follows: $A_\gamma \in \mathbb{R}^{n \times n}$ is the system matrix, $B_\gamma \in \mathbb{R}^{n \times l}$ is the control input matrix and $C_\gamma \in \mathbb{R}^{p \times n}$ the measurement matrix.[1]

For example, for the tank system our state vector is $x = \{h_1, h_2, h_3\}$, our input $u = \{q_0, V_1, V_2, V_3\}$, and $y = \{p_1, p_2, p_3\}$, our output. The output equation is given by

$$y_k \;=\; \begin{bmatrix} g & 0 & 0 \\ 0 & g & 0 \\ 0 & 0 & g \end{bmatrix} x_k \qquad (6)$$

We assume that we control the system using a state (Luenberger) observer based on a set of equations with observer matrix L. Using the observed system with observed state and measurement, $\hat{x}_k \in \mathbb{R}^n$ and $\hat{y}_k \in \mathbb{R}^p$, respectively:

$$\begin{aligned} \hat{x}_{k+1} &= A_\gamma \hat{x}_k + B_\gamma u_k + w_k; \\ \hat{y}_k &= C_\gamma \hat{x}_k + v_k; \end{aligned} \qquad (7)$$

we obtain the observer equations:

$$\begin{aligned} \hat{x}_{k+1} &= A_\gamma \hat{x}_k + B_\gamma u_k + L_\gamma(y_k - C_\gamma \hat{x}_k); \\ r_k &= y_k - C_\gamma \hat{x}_k; \\ u_k &= -K_\gamma \hat{x}_k, \end{aligned} \qquad (8)$$

where $r_k \in \mathbb{R}^p$ is the residual $y_k - \hat{y}_k$. We assume the control matrix $K_\gamma \in \mathbb{R}^{l \times p}$ and observer matrix $L_\gamma \in \mathbb{R}^{n \times p}$ are chosen so that the closed-loop system and error dynamics are stable.

[1] We assume that the initial conditions for all systems (e.g., x_0, $\hat{x}_0$ are known.

 ISBN: 978-1-61208-585-2

In the following, we assume that the actual input and measurement values, $\tilde{u}_k$ and $\tilde{y}_k$ respectively, can differ from the values of u_k and y_k due to data loss, noise in the network, faults, or due to a malicious attack $a_k \in \mathbb{R}^m$ on the system.

B. Fault Model

In this article we consider (a) sensor faults, where the sensor will either generate erroneous output or no output, and (b) plant/actuator faults. In the following we will specify additive fault models for these two fault classes. We define an additive fault vector $\boldsymbol{f}$, which we incorporate in a fault model as follows:

$$\begin{aligned} x^f_{k+1} &= A_\gamma x^f_k + B_\gamma u_k + B_f f_k + w_k; \\ y_k &= C_\gamma x^f_k + C_f f_k + v_k; \end{aligned} \tag{9}$$

where x^f_k is the faulty state vector at time k, B_f represents the influence the fault has on the state and C_f the influence of the fault on the measurement (sensor) data.

$$\begin{aligned} \tilde{x}_{k+1} &= A\tilde{x}_k + Bu_k + L(\tilde{y}_k - C\tilde{x}_k); \\ r_k &= \tilde{y}_k - C\tilde{x}_k; \\ u_k &= -K\tilde{x}_k, \end{aligned} \tag{10}$$

where we have: $\tilde{x}_k \in \mathbb{R}^n$ is the state of the observer, $\mathrm{u}_k \in \mathbb{R}^l$ the calculated control input, $\tilde{y}_k \in \mathbb{R}^p$ the measurements received over the network and $r_k \in \mathbb{R}^p$ is the residual. We assume the control matrix $K \in \mathbb{R}^{l\times p}$ and observer matrix $L \in \mathbb{R}^{n\times p}$ are chosen so that the closed-loop system and error dynamics are stable.

The values of $\tilde{u}_k$ and $\tilde{y}_k$ can differ from the values of u_k and y_k due to data loss, noise in the network, faults, or due to a malicious attack $a_k \in \mathbb{R}^m$ on the system.

C. Attack Model

We propose an attack model that specifies two types of attack: attacks on the system's actuators (or state), a^x_k, and attacks on the system output, a^y_k. Introducing the attack vector $a_k = [(a^x_k)^T (a^y_k)^T]^T$ to the plant and observer leads to

$$\begin{aligned} x_{k+1} &= A_\gamma x_k + B_\gamma u_k + B_a a_k + w_k; \\ y_k &= C_\gamma x_k + D_a a_k + v_k; \end{aligned} \tag{11}$$

where B_a represents the influence the attack has on the state by either a physical or an actuator attack and D_a the influence of the attack on the measurements by falsifying sensor data.

$$\begin{aligned} x_{k+1} &= A_\gamma x_k + Bu_k + B_a a_k + w_k; \\ y_k &= C_\gamma x_k + v_k; \\ \tilde{x}_{k+1} &= A_\gamma \tilde{x}_k + B_\gamma u_k + L_\gamma(y_k + D_a a_k - C_\gamma \tilde{x}_k); \\ r_k &= y_k + D_a a_k - C_\gamma \tilde{x}_k; \\ u_k &= -K_\gamma \tilde{x}_k; \end{aligned} \tag{12}$$

Due to the separation of the attacks into attacks on the states and the measurements, the attack matrices often take the structure

$$B_a = [B^x_a, \mathbf{0}] \text{ and } D_a = [\mathbf{0}, D^y_a]; \tag{13}$$

where $\mathbf{0}$ is the zero matrix with dimensions appropriate to the attack vector.

D. Extended System Model

We combine the plant and the observer to get an extended system. We define $\mathcal{X}_k = [x^T_k \tilde{x}^T_k]^T$ as the extended system state, the attack a_k as the input and the residual r_k as the system output

$$m_{k+1} = A_e m_k + B_e a_k + \begin{bmatrix} w_k \\ Lv_k \end{bmatrix}; \tag{14}$$

$$r_k = C_e m_k + D_e a_k + v_k \tag{15}$$

with

$$A_e = \begin{bmatrix} A & -BK \\ LC & A - BK - LC \end{bmatrix}, \quad B_e = \begin{bmatrix} B_a \\ LD_a \end{bmatrix} \tag{16}$$

$$C_e = [C \quad -C] \text{ and } D_e = D_a. \tag{17}$$

The initial state is given by $\mathcal{X}_0 = [x^T_0 \tilde{x}^T_0]^T$. Since K and L stabilize the plant and the error dynamics, A_e is stable as well. The residual r_k is used to determine how much the real system state deviates from the estimated state given by the observer, so we can use r_k to detect faults or attacks on the system.

IV. Distinguishing Faults from Security Breaches

This section focuses on methods for distinguishing faults from security breaches. We assume that a stealthy attacker will attempt to mask attacks as natural events, e.g., faults. In that case, we use the physics of the fault evolution and/or onset to isolate true faults.

A. Model-Based Isolation

We address this problem using a model-based framework. We assume that our system can be in one of q possible modes, where a mode characterizes a system behaviour. We can define modes corresponding to nominal, fault, and attack conditions.

We assume that we can specify the behaviour of each mode using a physical model of that mode. We denote model i using ψ_i. Our family of models $\Psi = \{\psi_1, ..., \psi_q\}$ consists of subsets of models denoting nominal, fault, and attack modes, $\{\Psi^N, \Psi^f, \Psi^a\}$ respectively. Model i generates a behaviour $\boldsymbol{\xi}_i$ (with measurement $\hat{\boldsymbol{y}}_i$) given initial conditions x_0. A behaviour over interval $[0, ..., T]$ is a state sequence $\{x_0, ..., x_T\}$.

Definition 1 (Mode Estimation): Our mode estimation task, given an anomalous observation $\tilde{y}$, is to compute the model whose behaviour most closely matches the observation $\tilde{y}$, i.e.,

$$\psi^* = \arg\min_{\psi_i \in \Psi} \parallel \tilde{y}_i - \hat{y}_i \parallel, \tag{18}$$

where $\parallel \tilde{y}_i - \hat{y}_i \parallel$ is a difference norm at instant i

 ISBN: 978-1-61208-585-2

We assume that we compute a residual vector $\boldsymbol{r} = \{r_1, ..., r_q\}$, with residual i associated with mode i. Residual r_i is "activated", i.e., $r_i > \delta_i$ for some tunable threshold δ_i, iff the system is in mode λ_i.

Definition 2 (Mode Identifiability): Given a model Ψ with a set of modes $\Lambda = \{\lambda_1, ..., \lambda_q\}$, mode i is identifiable (i.e., can be distinguished from mode j, for $i \neq j$) if (a) λ_i generates a behaviour ξ_i that is distinguishable from ξ_j for all $i \neq j$, and (b) there exists a residual r such that $r_i > \delta_i$ iff the system is in mode λ_i.

This notion of mode identifiability enables us to detect attacks, since an identifiable system guarantees that attacks can always be isolated. The ability to distinguish fault- and attack-modes depends on the fidelity of the models and the availability of appropriate sensor data.

B. Example: Sensor/Actuator Attack Detection

We assume a system in which we have the correct measurement y, the simulated measurement $\hat{y}$, and an attacker who injects a false measurement $\tilde{y}$ for a subset of the sensors. We can compute residuals for the "true" system as $r_i = \parallel y_i - \hat{y}_i \parallel$, and the system under attack as $r_i^a = \parallel \tilde{y}_i - \hat{y}_i \parallel$.

We compare r with r_a to distinguish faults from attacks. If $r_i = r_i^a, \quad \forall i > 0$ then the fault and attack are indistinguishable via physics-based analysis. Distinguishing faults from attacks also depends on the models assumed for faults from attacks. In this article we restrict our attention to attacks that fix the sensor/actuator at an anomalous value at some $k > 0$ and remains at that value.

Sensor Attack: We assume that, given a physical fault (e.g., stuck actuator or tank leak) a sensor will report the physical deviations from nominal conditions. For example, a tank leak in tank T_2 will lead to lower-than-expected tank height for T_2, such that the deviation will increase over time. In this article we look at residuals, but also rates of change of outputs y_k and residuals r_k, i.e., $\dot{y}_k$, $\dot{r}_k$, respectively.

Actuator Attack: If we restrict our fault model to "stuck" actuators, e.g., a valve that gets stuck open, then our attack model can exactly mimic a "stuck" actuator, and hence this class of attack cannot be distinguished from a "stuck" actuator fault.

V. Experimental Results

In the following we will show some tests and results achieved on the three-tank system, starting from the data simulation of the system itself in various conditions through the analysis and fault detection of these data.

A. Simulations

We based our experiments on our own simulated data of the three-tank system. In order to correctly simulate faults and attacks we set three different simulation modes for the system: normal, faulty and attack.

The faulty simulation included a random delta value for each valve, either positive or negative, in order to reproduce a positioning problem differing from the normal value. Each delta value is independent from each other, plus the final valve position will still respect the [0,1] interval constraint.

The attack simulation influences either the valve settings, the sensor measurements or both. The principle of each attack is the same: the attacker sets a fixed value to one or more of the system's components, overriding the correct value. The difference between the two attacks relies on the fact that a valve attack immediately influences the system behaviour, forcing more (or less) fluid to go through the system. On the other hand, attacking a sensor could not be as effective in the case of a non-feedback system.

We run our simulations on a 50 and 500 seconds period, extracting data from our sensors every 2 seconds.

B. Analysis procedure

In order to detect faults and attacks on our system, we used residuals and first derivative studies of the sensor data. Relative errors and deeper derivative studies were performed, but we were not able to extract good results from them.

We were able to identify incongruences in the data when the residuals were over a predefined tolerance. On the other hand, the gradients were able to give an idea of how the data would evolve in time, allowing us to identify absolute tendencies of data.

This approach has been proven to be a good way to find injected sensor data. When sensor data are attacked, we obtain the relation

$$\dot{y}_k = -\dot{r}_k, \tag{19}$$

where $\dot{y}_k$ and $\dot{r}_k$ denote the first derivative of the sensor output and the residual, respectively.

The fact that we are limited of having only the sensor data allows us to detect when a fault or attack occurred but they are not enough to identify which valve had a problem and if the problem was an attack or a random fault of the system.

C. Experiment I - Attacks on sensors

The objective of our first experiment was to identify whether a sensor has been attacked or not.

Thanks to the first derivative analysis of the normal behaviour and the residuals we were able to identify in which cases the data were crafted by an attacker.

Figure 2 shows the data generated by an attacked sensor. We can clearly see how that equation 19 holds, i.e., the residual function is a y-mirrored version of the normal behaviour.

Each sensor attack is correctly detected by our procedure, either alone or in conjunction with other attacks.

D. Experiment II - Attacks on Actuators (Valves)

This experiment addresses detection and isolation of actuator attacks, i.e., valves whose control setting are set to be incorrect.

We started analysing only one attack per simulation. In each simulation we were able to detect that an attack has occured, but we could not precisely locate on which valve. Besides we found that our procedure observed attacks on

 ISBN: 978-1-61208-585-2

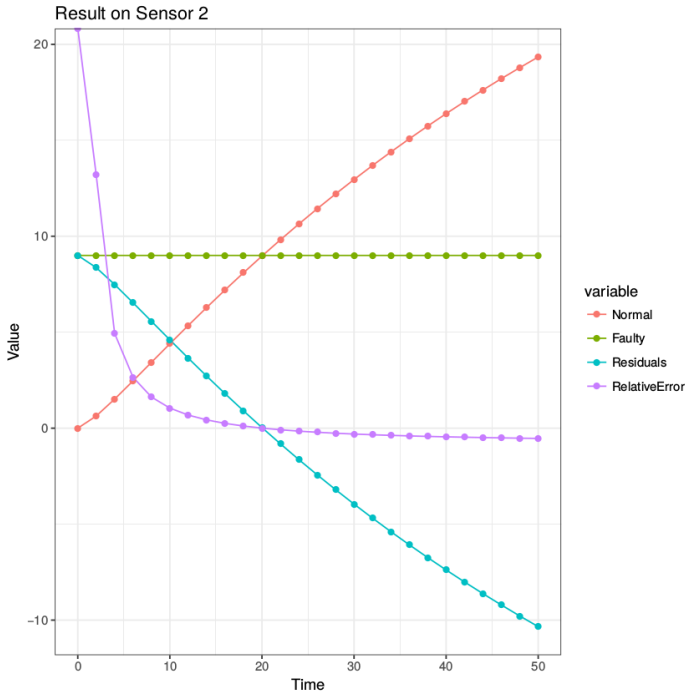

Figure 2. Injected data on the second sensor of our system. The graph shows the normal and faulty behaviour and their related residuals and relative errors.

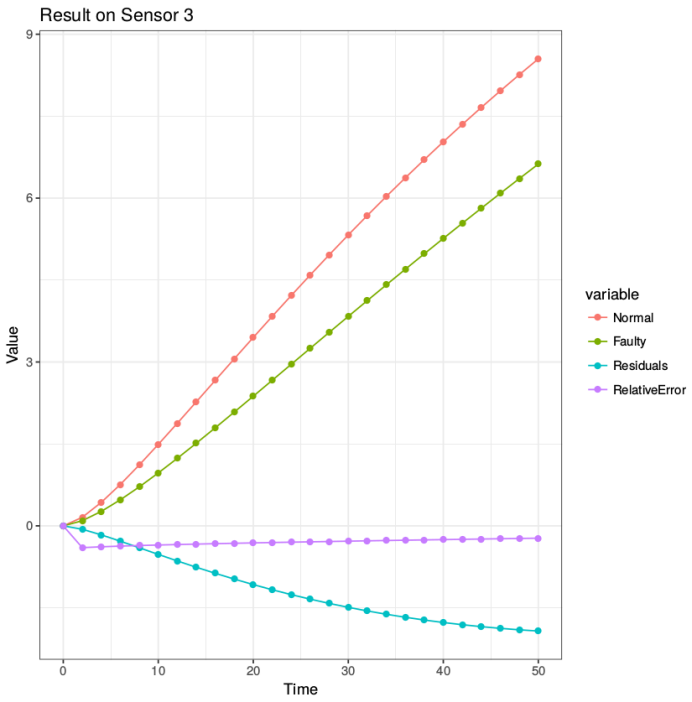

Figure 3. Data of the third sensor related to test 535.

different valves even if the attack was performed only on one: this is due to the complexity and synergies of the system itself which we were not be able to capture with only data from the pressure sensors.

Figure 3 shows the data of an attacked valve, while Table I shows which faults were detected for each experiment:

Test	Valve 1	Valve 2	Valve 3
155	✓	X	X
355	✓	X	X
755	✓	X	X
955	✓	X	X
515	X	✓	X
535	X	✓	X
575	X	✓	X
595	X	✓	X
551			✓
553			✓
557			✓
559			✓

TABLE I. Results of our procedure. The test number shows the valve settings for each valve (i.e. 155: v1 = 0.1, v2 = 0.5, v3 = 0.5). The nominal setting is 555. A valve is marked when our procedure identifies a problem with it. ✓ denotes a correct diagnosis, and X denotes an incorrect diagnosis.

We also tested combination of attacks: attacks are still detected, but is even more difficult to identify on which valves the attack was done. The results of our tests are shown in Table II.

Test	Valve 1	Valve 2	Valve 3
544	X	✓	✓
158	✓	X	✓
658	✓	X	✓
958	✓	X	✓
745	✓	✓	X
247	✓	✓	✓
432	✓	✓	✓
632	✓	✓	✓
638	✓	✓	✓
678	✓	✓	✓

TABLE II. Results of our procedure on multi valve attacks.

E. Experiment III - Attacks on both Sensors and Actuators (Valves)

The goal of this experiment was to combine the previous experiments and see how simultaneous attacks impact the system and if we were still able to identify which parts of the system have been attacked. We presume to be able to correctly detect sensor problems and the presence of valve errors, but cannot identify the faulty valves, as happened also in experiment II.

As expected and shown in Table III we are able to identify the attacks on the sensors but not on the valves.

VI. Related Work

This article extends the work of [7], who describe a framework for detecting security breaches in networked control systems. [7] make the simplifying assumption that anomalies due to security breaches and to other sources are *a priori* separable, so the task of identifying security breaches becomes trivial. In real situations, this assump-

 ISBN: 978-1-61208-585-2

Test	Valve 1	Valve 2	Valve 3	Sensor
s1_325		✓	X	1
s2_553	X		✓	2
s3_148	✓	✓		3
s12_558			✓	1-2
s23_647	✓			2-3
s31_348		✓		1-3
s123_666				1-2-3

TABLE III. Results of our procedure on multi valve attacks. The test number, other than showing the valve settings, shows also which sensors are attacked.

tion does not hold, and we focus on methods for distinguishing faults from security breaches.

VII. Conclusion

This article has proposed a physics-based approach for modeling a CPS and using this model to distinguish faults from attacks. We have shown on a hydraulic system the capabilities of this approach. We have also shown that not all attacks can be identified via this physics-based approach. To extend this approach, deeper studies on sensors data synergies are needed in order to extract some more information about possible valve faults/attacks.

Acknowledgment

The authors would like to thank SFI for funding this work under grants SFI grants 12/RC/2289 and 13/RC/2094.

References

[1] E. A. Lee and S. A. Seshia, Introduction to embedded systems: A cyber-physical systems approach. MIT Press, 2016.

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

[3] D. I. Urbina, J. A. Giraldo, A. A. Cardenas, N. O. Tippenhauer, J. Valente, M. Faisal, J. Ruths, R. Candell, and H. Sandberg, "Limiting the impact of stealthy attacks on industrial control systems," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016, pp. 1092–1105.

[4] S. Lakshminarayana, T. Z. Teng, D. K. Yau, and R. Tan, "Optimal attack against cyber-physical control systems with reactive attack mitigation," in Proceedings of the Eighth International Conference on Future Energy Systems. ACM, 2017, pp. 179–190.

[5] T. Darure, J.-J. Yamé, and F. Hamelin, "Model-based fault-tolerant control of vav damper lock-in place failure in a multizone building," in Control, Automation, Robotics and Vision (ICARCV), 2016 14th International Conference on. IEEE, 2016, pp. 1–6.

[6] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal, "Identification of hybrid systems: a tutorial," European journal of control, vol. 13, no. 2-3, 2007, pp. 242–260.

[7] D. Umsonst, H. Sandberg, and A. A. Cárdenas, "Security analysis of control system anomaly detectors," in American Control Conference (ACC), 2017. IEEE, 2017, pp. 5500–5506.

 ISBN: 978-1-61208-585-2