Sensor Attack Detection in the Presence of Transient Faults

Junkil Park
Dept. of Computer & Information Sc.
University of Pennsylvania
park11@seas.upenn.edu

Radoslav Ivanov
Dept. of Computer & Information Sc.
University of Pennsylvania
rivanov@seas.upenn.edu

James Weimer
Dept. of Computer & Information Sc.
University of Pennsylvania
weimerj@seas.upenn.edu

Miroslav Pajic
Dept. of Electrical & Systems Engineering
University of Pennsylvania
pajic@seas.upenn.edu

Insup Lee
Dept. of Computer & Information Sc.
University of Pennsylvania
lee@cis.upenn.edu

ABSTRACT

This paper addresses the problem of detection and identification of sensor attacks in the presence of transient faults. We consider a system with multiple sensors measuring the same physical variable, where some sensors might be under attack and provide malicious values. We consider a setup, in which each sensor provides the controller with an interval of possible values for the true value. While approaches exist for detecting malicious sensor attacks, they are conservative in that they treat attacks and faults in the same way, thus neglecting the fact that sensors may provide faulty measurements at times due to temporary disturbances (e.g., a tunnel for GPS). To address this problem, we propose a transient fault model for each sensor and an algorithm designed to detect and identify attacks in the presence of transient faults. The fault model consists of three aspects: the size of the sensor's interval (1) and an upper bound on the number of errors (2) allowed in a given window size (3). Given such a model for each sensor, the algorithm uses pairwise inconsistencies between sensors to detect and identify attacks. In addition to the algorithm, we provide a framework for selecting a fault model for each sensor based on training data. Finally, we validate the algorithm's performance on real measurement data obtained from an unmanned ground vehicle.

Categories and Subject Descriptors

K.6.5 [**Security and Protection**]: Unauthorized access (e.g., hacking, phreaking); C.3 [**Special-purpose and Application -based Systems**]: Process control systems, Real-time and embedded systems

ICCPS '15, April 14 - 16, 2015, Seattle, WA, USA

ACM 978-1-4503-3455-6/15/04$15.00
http://dx.doi.org/10.1145/2735960.2735984.

1. INTRODUCTION

As Cyber Physical Systems (CPS) become increasingly widespread in safety-critical domains, ensuring the safety of such systems is imperative. Malicious attacks exploiting security vulnerabilities can have a catastrophic impact upon CPS, thus undermining their safety. For instance, recent work has shown that it is possible for an attacker to hijack a modern vehicle through vulnerabilities in the vehicle's on-board communication protocol [4, 16] or sensor spoofing [2, 21]. Furthermore, the worm Stuxnet was able to disable critical infrastructure by exploiting weaknesses in Supervisory Control And Data Acquisition (SCADA) systems [6].

Since modern CPS are equipped with multiple sensors measuring the same physical variable (e.g., GPS, wheel encoder and IMU can provide velocity measurements), a system can use the redundant information to defend against malicious attacks. Fusing their measurements not only produces an estimate that is more precise than any single sensor's [15], but it also increases the system's robustness to external disturbances (e.g., rough terrain for automotive CPS).

Increased sensor diversity, however, leads to a greater vulnerability to attacks. While some sensors may be difficult to get access to (e.g., encoder), others may be easy to spoof (e.g., GPS [21]). Consequently, in this work we address the general problem of developing a resilient sensor fusion algorithm in the presence of attacks.

The first consideration when designing a resilient system is the sensor model. There are two main classes of sensor models: *probabilistic* and *abstract*. In the former, each sensor gives a numeric measurement that is corrupted by noise with a known distribution (e.g., Gaussian) [15]. In the latter, an interval is constructed around each sensor measurement, containing all points that may be the true value (e.g., set membership methods [18]), thus being well-suited for worst-case analysis. Note that the abstract model does not make any assumptions about the distribution of the sensor measurements or noise; however, the interval's size reflects the sensor's precision - a small interval implies high confidence in the obtained measurement. The abstract interval can be constructed based on manufacturer specifications about precision and accuracy of the sensor, as well as physical limitations such as sampling jitter and synchronization errors [19].

In this work, we address the problem of CPS security, which is usually concerned with worst-case system operation; hence, we adopt the abstract sensor model. There has been much work on sensor fault detection and isolation published over the last half century, with the vast majority focused on scenarios with probabilistic sensors (e.g., see [5, 7, 8, 10] and citations within). This work differs from the classical fault detection literature in the assumed prior information; classical approaches either assume a known sensor failure model [22] (e.g., jammed actuator) or a prior on the initial condition [11]. In the presence of attacks, such assumptions may lead to increased vulnerability [19], therefore we adopt a more general approach.

Related to the work herein, one of the first works employing the abstract sensor model [17] considers the case where some of the sensors might be faulty (i.e., outputting intervals that do not contain the true value); it provides worst-case analysis when the number of faulty intervals can be bounded. An extension relaxes the worst-case guarantees in favor of obtaining more precise fused measurements [3]. In addition, intervals can be assumed to have a predefined distribution on the true value so that again statistical analysis can be performed [23]. Finally, an attack-resilient version of [17] has been developed by introducing a sensor communication schedule that limits the attacker's power [12] and by incorporating measurement history into sensor fusion [13]. A primary shortcoming of existing fault/attack detection methods based on abstract sensors [12, 14, 17] is the unilateral treatment of faults as attacks. Thus, these approaches may lead systems to not trust sensors that were only transiently faulty (i.e., that malfunction and provide wrong measurements for a small period of time) and not under attack.

It is important to note that transient faults may occur during the system's normal operation and disappear shortly after. In fact, most sensors exhibit a transient fault model that bounds the amount of time in which they provide wrong measurements. For example, it is not uncommon for GPS to temporarily lose connection to its satellites (or receive noisy signals), especially in cities with high-rise buildings. Similarly, a sensor transmitting data using an over-utilized network (e.g., with the TCP/IP protocol with retransmissions) may fail to deliver its measurements on time, thus providing incorrect information when the messages do arrive. Due to their short duration, however, transient faults should not be considered as a security threat to the system.

In contrast, permanent faults are sensor defects that persist for a longer period of time and may seriously affect the system's operation. For instance, a sensor may suffer physical damage that introduces a permanent bias in its measurements. In such a scenario, unless the fault can be corrected for in the software, the system would benefit from discarding this sensor altogether.

Depending on the attacker's goal, attacks on sensor measurements may manifest either as transient or permanent faults. Each one has benefits and drawbacks – making a sensor behave as if transiently faulty may prevent the attacker from being discovered but also limits his capabilities, whereas a prolonged attack that is similar to a permanent fault may be more powerful but could be detected quickly. Since this work is a first step towards ensuring CPS security, we address the detection and identification of attacks that appear as permanent faults – i.e., that do not comply with the transient fault model. We leave the detection of the more stealthy variety for future work.

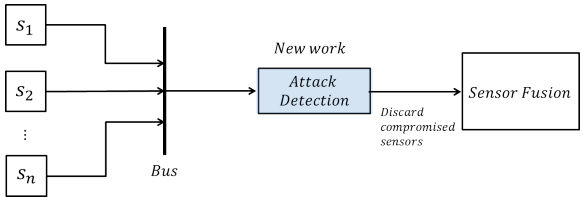


Figure 1: The proposed approach can be used as a front-end detector to the actual sensor fusion algorithm by removing compromised sensors that do not comply with the transient fault model.

Different from previous work, the main focus of this work is the detection of sensor attacks in the presence of transient faults under the abstract sensor model. We develop an algorithm that assumes a transient fault model (described below) for each sensor and raises an alarm if the observed sensor data does not match this model. In particular, we employ pairwise sensor relationships to develop sufficient conditions for attack detection and identification.

In addition to classical bounded errors, manufacturers now provide transient fault specifications for their sensors [9], consisting of three dimensions: (1) the interval size, (2) window size and (3) the number of allowed faulty measurements per window. When such a specification is not provided, we note that intuitively there exists a range of interval sizes that match the underlying sensor noise model and a range that matches the underlying fault model, i.e., sometimes sensors provide measurements that are significantly farther from the true value than noisy ones. Thus, we provide a framework for choosing the three parameters in order to capture the transient fault model but not include the noisy measurements. We illustrate this process using a real-date example.

Finally, we validate our approach on real data obtained from an unmanned ground vehicle, called the LandShark [1]. We provide the algorithm's detection and identification accuracy for three different classes of attacks and present the corresponding receiver operating characteristic (ROC) curves for different sensor models. In addition, we illustrate the algorithm's advantage over the current approach in [17].

Note that, as shown in Fig. 1, the proposed algorithm can be considered as a front-end procedure to the actual sensor fusion algorithms described in previous work [12, 14, 17]. As such, it can be used to eliminate sensors that are attacked or permanently faulty before sensor fusion is performed [17]; our approach is not complete but is sound, i.e., given correct fault models, it will not raise false alarms. On the other hand, it provides no guarantees about the output of sensor fusion in any particular round; bounding the performance of sensor fusion given that the remaining sensors satisfy their transient fault models is a topic for future work.

In summary, the contributions of this work are: (1) a sensor attack detection algorithm in the presence of faults based on the abstract sensor model; (2) a method, with a real-data analysis, for fault model parameter selection; (3) an evaluation of the fault detection scheme on a robotic platform in comparison to the approach introduced in [17]. The remainder of this paper is organized as follows. The subsequent section provides preliminaries and formulates the problem considered. Sections 3 presents our attack detection approach, and Section 4 illustrates it with an example. Section 5 presents transient fault modeling. A robotic case

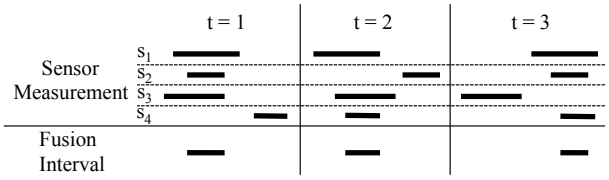


Figure 2: Motivating example: where transient faults are treated as attacks.

study evaluation is provided in Section 6, and Section 7 concludes this work.

2. PROBLEM FORMULATION AND PRELIMINARIES

This section describes the problems considered in this paper. It provides the system model used in this work as well as the current attack detection approach. We explain its shortcomings and propose a way of eliminating them through introducing a transient fault model for each sensor.

2.1 System Model and Current Approach

We consider a system with n sensors measuring the same physical variable. As mentioned above, we assume *abstract* sensors; therefore, each sensor provides the controller with an interval of all possible values. We assume the system queries all the sensors periodically such that a centralized estimator receives measurements from all sensors, and then performs attack detection/identification and *sensor fusion* (SF). We now explain the current approach to attack detection, referred to herein as a SF-based detector, before providing the improved version addressed in this paper.

As described in [17], the abstract sensor model assumes an error bound ϵ_i for each sensor s_i, i.e., s_i's interval size is $2\epsilon_i$. If s_i's measurement at time t, denoted by $y_i^{(t)}$, deviates from the true value, $\theta^{(t)}$, by more than ϵ_i, $y_i^{(t)}$ is said to provide a *faulty measurement*. Formally, this statement is captured by the predicate $F(i,t)$ of sensor index i and time t.

Definition 1 (Faulty Measurement).

$$F(i,t) \equiv |y_i^{(t)} - \theta^{(t)}| > \epsilon_i.$$

Although $\theta^{(t)}$ is unknown in general, one can derive an interval that is minimal in size and guaranteed to contain $\theta^{(t)}$ by assuming that of the measurements of the n sensors, no more than f are faulty at each time [17]. This interval is referred to as the *fusion interval*. Since any point contained in $(n-f)$ intervals may be the true value, the fusion interval is the smallest interval containing all such points.[1] Any sensor measurement that does not intersect the fusion interval must be faulty since the fusion interval is guaranteed to contain the true value [17]. Thus, the SF-based detection algorithm built on the abstract sensor model identifies sensors not intersecting the fusion interval as compromised, i.e., attacked or permanently faulty.

2.2 Motivation and Example

Figure 2 illustrates the sensor fusion algorithm for three rounds assuming a maximum of 1 compromised sensor per round ($f = 1$). Applying the SF-based detection algorithm, sensor 4 is declared compromised at time $t = 1$ since it does not overlap the fusion interval; similarly, s_2 and s_3 are declared compromised at $t = 2$ and $t = 3$, respectively. Suppose, however, that each sensor is known to occasionally provide faulty measurements (e.g., GPS in a tunnel), such that it may provide at most 1 faulty measurement in any window of size 3. Each of s_2, s_3 and s_4 satisfy this criterion in Figure 2. In the presence of these transient faults, the SF-based detector declares all faults as malicious attacks. Motivated by recent manufacturer trends to provide *faulty-measurements-per-window* specification for sensors, we aim to incorporate a transient fault model for each sensor to improve the overall system performance.

2.3 Transient Fault Model

A transient fault model for sensor s_i is a triple (ϵ_i, e_i, w_i) where ϵ_i is the error bound, and (e_i, w_i) is a transient threshold specifying that s_i can provide at most e_i faulty measurements in any window of size w_i. The transient threshold is used to define the boundary between transient faults and non-transient faults. If s_i violates its transient threshold, it is said to be *non-transiently faulty*, denoted by predicate $NTF(i,t)$.

Definition 2 (Non-Transiently Faulty Sensor).

$$NTF(i,t) \equiv \left(\sum_{t'=t-w_i+1}^{t} F_1(i,t') \right) > e_i,$$

where $F_1(i,t) = 1$ *if* $F(i,t)$, *and* $F_1(i,t) = 0$ *if* $\neg F(i,t)$.

Note that sensors may be non-transiently faulty due to either permanent faults or sensor attacks; the above definition captures both of these scenarios. Of course, there might also be attacks that comply with the transient fault model. As described in the introduction, in this work we only consider attacks that manifest as non-transient faults and defer the analysis of other attacks for future work.

Consequently, permanent faults and attacks are both formalized as non-transient faults, and hence we treat them in the same way. Therefore, if sensor s_i is non-transiently faulty at least one time during the whole system operation, we say that s_i is attacked. Predicate $A(i)$ formally defines attacked sensors.

Definition 3 (Attacked Sensor).

$$A(i) \equiv \exists t \leq T, NTF(i,t),$$

where T *is the total time of the system's operation.*

2.4 Problem Statements

There are three problems addressed in this work. For the first we note that while transient fault models may exist for certain systems, in general they are unknown, and it is not straightforward to develop them. Thus the first problem is the following:

Problem 1. *Given a system with n sensors and a set of training measurement data, develop a transient fault model for each sensor* s_i.

Once such models have been derived, we develop an algorithm that uses them in order to detect and identify sensor attacks while the system is operating.

[1] Note that the fusion interval can be bounded when $f < \lceil n/2 \rceil$ [17].

PROBLEM 2. *Given a system with n sensors and a transient fault model (ϵ_i, e_i, w_i) for each sensor, develop an algortihm to* detect *the existence of an attacked sensor.*

PROBLEM 3. *Given a system with n sensors and a transient fault model (ϵ_i, e_i, w_i) for each sensor, develop an algorithm to* identify *which sensor is attacked.*

For Problem 3, we assume that there are at most a attacked sensors, but do not limit the number of faults at a round unlike the current abstract sensor fusion approach. Naturally, Problem 3 (i.e., identification) is harder than Problem 2 (i.e., detection) since obtaining a positive solution for identification also provides a solution for detection, but not vice versa. Yet, even if they cannot isolate the attacked sensor, systems could still benefit from the knowledge that an attack is present; therefore, both the detection and identification problems are addressed in this work.

3. A SOUND ALGORITHM FOR ATTACK DETECTION AND IDENTIFICATION

This section describes our Pairwise Inconsistency (PI) approach to the sensor attack detection and identification problems. Note that in this section we assume that a transient fault model has been provided for each sensor and discuss how to derive such a model in Section 5. Motivated by the example in Figure 2, we aim to develop an attack detection and identification approach that can differentiate between those sensors which are attacked and those which are merely transiently faulty.

The key concepts in the following approach are two types of pairwise inconsistencies between sensors: *weak inconsistency* and *strong inconsistency*. We accumulate the information of strong inconsistency over time and use it for attack detection and identification. The following subsections first define weak and strong inconsistencies, then formally presents our attack detection and attack identification methods.

3.1 Weak and Strong Inconsistency

This section is built on the premise that the true value is not known in general, hence it is not known which sensors have provided correct measurements. What is known, however, is how sensor measurements relate to each other, and it is this mutual information that is used in this work.

The first relationship between two sensors, s_i and s_j, is weak inconsistency, denoted by the predicate $WI(i, j, t)$. Two sensors are weakly inconsistent in a given round if and only if one of them provides a faulty measurement.

DEFINITION 4 (WEAK INCONSISTENCY).

$$WI(i, j, t) \equiv F(i, t) \vee F(j, t).$$

Note that this condition is not possible to verify in general since the true value is not known. Yet, there exists a useful sufficient condition for weak inconsistency, which is the essence of this work. In particular, if two sensors' intervals do not intersect with each other, then one of them must have provided a faulty measurement, since the true value cannot lie in both. This is formally stated in the following lemma:

LEMMA 1. *Given i, j and t,*

$$|y_i^{(t)} - y_j^{(t)}| > \epsilon_i + \epsilon_j \implies WI(i, j, t)$$

PROOF. Assume for a contradiction that both s_i and s_j provide non-faulty measurements at time t, i.e., there exists $\theta^{(t)}$ satisfying $|y_i^{(t)} - \theta^{(t)}| \leq \epsilon_i$ and $|y_j^{(t)} - \theta^{(t)}| \leq \epsilon_j$. This implies that

$$|y_i^{(t)} - y_j^{(t)}| = |(y_i^{(t)} - \theta^{(t)}) - (y_j^{(t)} - \theta^{(t)})| \leq$$
$$|y_i^{(t)} - \theta^{(t)}| + |y_j^{(t)} - \theta^{(t)}| \leq \epsilon_i + \epsilon_j$$

which contradicts the premise of the Lemma statement. □

As discussed in Section 2.2, transient faults and attacks may both manifest as weak inconsistencies in a single round. Therefore, we introduce a *strong inconsistency* to disambiguate between the two. A strong inconsistency exists between two sensors if and only if one is non-transiently faulty (i.e., one sensor does not match its transient fault model). Formally,

DEFINITION 5 (STRONG INCONSISTENCY).

$$SI(i, j, t) \equiv NTF(i, t) \vee NTF(j, t)$$

Similar to WI, strong inconsistencies cannot be verified in general. However, again a sufficient condition exists. In particular, if two sensors are weakly inconsistent too frequently in a given window, they become strongly inconsistent.

LEMMA 2. *Given i, j, t,*

$$\sum_{t'=t-\min(w_i,w_j)+1}^{t'=t} WI_1(i, j, t') > e_i + e_j \implies SI(i, j, t)$$

PROOF. Note that a weak inconsistency at time t' implies at least one sensor provides a faulty measurement at t', hence the premise implies that the number of faulty measurements in both sensors combined is also greater than $e_i + e_j$. This means that, in a window of size $\min(w_i, w_j)$, either s_i has at least e_i faulty measurements or s_j has at least e_j faulty measurements. In turn, this implies that one of them must be non-transiently faulty. □

The pairwise notions on inconsistency discussed in this subsection provide a basis for the attack detection and attack identification approaches presented in the following subsections.

3.2 Attack Detection

In this subsection, we present our approach to attack detection by employing the inconsistency notions above. The attack detection approach developed herein only concerns detecting the *existence* of a sensor attack and does not consider the (harder) problem of *which* sensors are attacked, which is addressed in the following subsection. In words, our detection algorithm declares that there is an attacked sensor if a strong inconsistency between any two sensors has ever occurred. The remainder of this subsection formalizes the attack detector.

In order to monitor whether a strong inconsistency has ever occurred, we employ a sequential detection approach [20] and accumulate the information of strong inconsistency over time. We use the predicate $SI^*(i, j)$ to denote that there exists a time $t \leq T$ in which sensors s_i and s_j are strongly inconsistent. Formally,

DEFINITION 6 (ACCUMULATED SI).

$$SI^*(i, j) \equiv \exists t \leq T, SI(i, j, t),$$

where T is the total time of the system's operation.

Employing SI^*, we can formally describe attacks, as shown below.

LEMMA 3. *Given s_i, s_j*

$$SI^*(i,j) \implies A(i) \vee A(j)$$

PROOF. From the definition,
$SI^*(i,j) \equiv \exists t, (NTF(i,t) \vee NTF(j,t))$. This implies
$(\exists t, NTF(i,t)) \vee (\exists t, NTF(j,t)) \implies A(i) \vee A(j)$. □

To simplify the problem analysis and for better illustration, we observe that the pairwise relationship between sensors can be represented in a graph, where each sensor corresponds to a node, and an edge is added between two nodes if there exists an inconsistency between them. Consequently, the attack detection problem reduces to a graph problem over an *inconsistency graph*.

DEFINITION 7 (INCONSISTENCY GRAPH). *The inconsistency graph $G(V,E)$ is an undirected graph where*

- *V = the set of all sensor indices $\{1,...,n\}$*
- *$E = \{(i,j) \mid SI^*(i,j)\}$*

Thus, a sensor attack is detected whenever an edge exists in the inconsistency graph, i.e., there exists a strongly inconsistent pair of sensors.

THEOREM 1. $E \neq \emptyset \implies \exists i : A(i)$.

PROOF. Since $E \neq \emptyset$, there exists a pair $(i,j) \in E$. This implies $SI^*(i,j)$ by Lemma 3. □

Having described the properties of the inconsistency graph, we utilize it in the following section to formalize the attack identification approach.

3.3 Attack Identification

We now consider the attack identification problem. To perform identification, we require the additional assumption that at most $a < n-1$ sensors are attacked during system execution. Note that a can be as large as $n-2$ but it cannot be either $n-1$ or n since in those cases even if the inconsistency graph is a clique, one would be not able to identify which sensors are attacked.

Assuming $a < n-1$, one may derive a sufficient condition for a sensor attack as follows. Suppose sensor s_i is strongly inconsistent with more than a other sensors. In this case, we can identify s_i as attacked; for if not, then all sensors which are strongly inconsistent with s_i must be attacked. However, the number of attacked sensors in this case would be greater than a, thus contradicting the assumption of at most a attacked sensors.

THEOREM 2. *Let* $\mathbf{deg}(i)$ *denote the degree of a vertex i in graph G. Given i,*

$$\mathbf{deg}(i) > a \implies A(i)$$

PROOF. Let sensor s_i be the sensor which is connected to $d > a$ other sensors in the inconsistency graph. Suppose that s_i is not attacked. It follows that the d sensors which are connected to sensor i must be attacked. This is a contradiction because there are at most a attacks. □

The attack detection and identification approach presented in this section provides a sufficient condition for the existence and isolation of sensor attacks. The following section illustrates various features through the use of pedagogic examples.

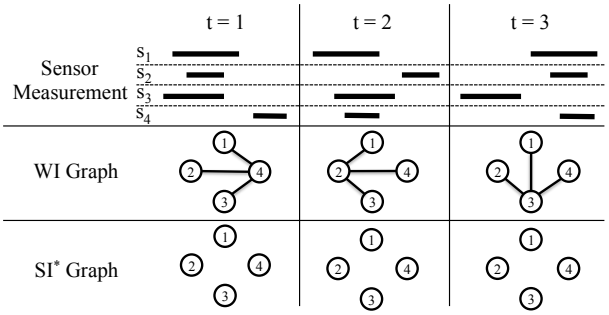


Figure 3: Motivating example revisited: where transient faults are not treated as attacks.

4. ATTACK DETECTION EXAMPLES

To illustrate the PI-based attack detection and identification approach, this section provides examples highlighting various features. In the following, we first revisit the motivating example from Section 2, then present a more involved example illustrating the nuances of the attack detection and identification approach.

4.1 Transient Fault Example: Revisited

We begin by reconsidering the motivating example in Figure 2 with respect to the PI-based attack detection and identification strategy presented in this work. The results are shown in Figure 3, where we recall that the transient fault model for each sensor s_i is assumed to be $e_i = 1$ and $w_i = 3$. In Figure 3, we visualize weak (and strong, respectively) inconsistencies as solid edges in the WI (SI^*) graphs. We observe that one sensor is weakly inconsistent with the other three sensors at each time step: s_4 at $t = 1$; s_2 at $t = 2$; and s_3 at $t = 3$. While the SF-based approach detects an attack at $t = 1$ based on the weak inconsistency, the PI-based approach does not alarm since all faults are transient. In particular, the sum of weak inconsistencies between any pair of nodes is at most 2, hence the attack detection (and identification) approach presented in this section does not detect an attack, i.e., there are no edges in the SI^* graphs.

This example illustrates that the PI-based approach is robust to transient faults and is conservative in the sense that an alarm is raised only if the measurements cannot be explained as transient faults. However, the lowered false alarm rate does not come at the cost of a decreased attack detection (and identification) rate, as the following example describes.

4.2 Attack with Transient Faults Example

To illustrate the detection and identification scheme used in this paper we utilize the fabricated example shown in Fig. 4. We consider a system with 5 sensors over 6 time steps. Suppose that sensors 3 and 4 are under attack; the transient fault model for each sensor is given as shown in Table 1. Additionally, we assume that there is a maximum of 2 attacked sensors. In Fig. 4, the vertical dotted lines indicate the true values (unknown to the attack detection system).

At time 1, the intervals of sensors s_1 and s_3 are disjoint implying that at least one of the two sensors must have provided a faulty measurement (weak inconsistency). Similarly, we detect 5 weak inconsistencies in total at time 1 such as (1, 2), (1, 3), (2, 3), (2, 4), (2, 5), as shown in the WI graph.

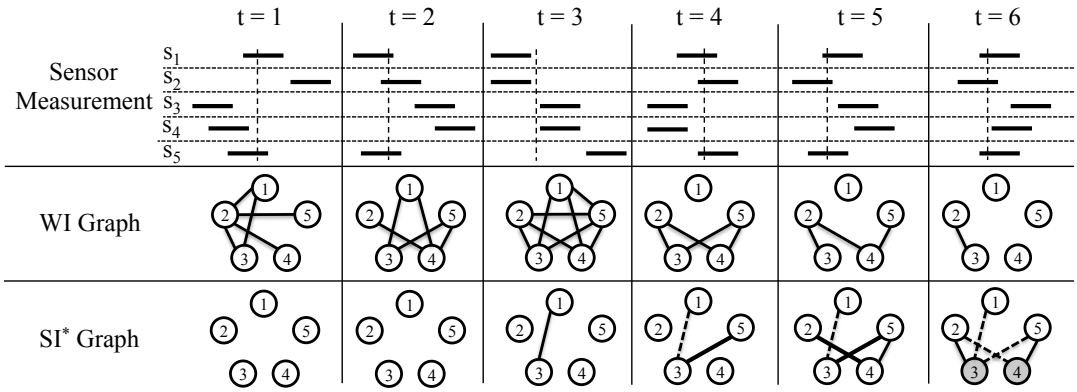


Figure 4: An example showing the detection and identification capabilities of the algorithm proposed in this paper. Dashed edges in the SI^* graphs indicated edges that were drawn in previous rounds.

Table 1: Transient fault models for the sensors used in the larger example in Section 4.

Sensor	(ϵ_i, e_i, w_i)
s_1	(1,1,6)
s_2	(1,2,5)
s_3	(1,1,4)
s_4	(1,2,6)
s_5	(1,1,5)

More weak inconsistencies are detected at times 2 and 3. At $t = 3$, however, we conclude s_1 and s_3 are strongly inconsistent because they have been weakly inconsistent three times in a window of size three, which satisfies the sufficient condition described in Lemma 2. Thus, an edge is drawn in the SI^* graph (note that dashed lines indicate edges that were drawn in previous rounds). Further strong inconsitencies, namely (3,5) and (2,4) are identified at times 4 and 5, using the same reasoning.

At $t = 6$, we add the strong inconsistency between s_2 and s_3, which allows us to *identify* s_3 as attacked. To do so, we utilize Theorem 2; node 3 now has degree 3, which is more than the number of assumed attacked sensors, i.e., 2, hence s_3 must be attacked. Furthermore, by removing s_3 from the graph and updating the assumption a from 2 to 1, we can also identify s_4 as attacked, due to its 2 edges.

This example describes a toy scenario for attack detection and identification. It illustrates that the PI-based detector not only reduces the number of false alarms but is also able to detect and identify attacks. Further evaluation is performed in Section 6.

5. TRANSIENT FAULT MODELING

The PI-based attack detection and identification algorithm requires accurate transient fault models. As argued in Section 2, modern manufacturers are transitioning towards providing transient fault specifications for their sensors as that allows them to perform more sophisticated analysis [9]. However, if such a model is not provided or the sensor is operated in unfavorable environments (e.g., using a GPS while surrounded by tall buildings), it may be necessary to develop transient fault models based on empirical data. Thus, this section presents parameter constraints governing the feasibility of the attack detection and identification approach presented in Section 3, as well as introduces a parameter selection procedure to specify sensor-specific transient fault models.

5.1 Transient Fault Parameter Constraints

In this subsection, we provide theoretical bounds on the transient fault model parameters constraining the feasibility and performance of the detection and identification approach described above. For sequential detectors, an important measure of performance is the *time-to-detection*, which is the time that elapses from the onset of an event (attack) until it is detected [20]. The time-to-detection of the attack detector presented in this section depends on the number of faults allowed for the different sensors in a window. We let $\mathcal{E}$ denote $(e_1, e_2, ..., e_n)$ and find a lower bound on the time-to-detection based on $\mathcal{E}$.

LEMMA 4. *Let e_{p_1} and e_{p_2} be the two smallest numbers in $\mathcal{E}$ (with corresponding sensors s_{p_1} and s_{p_2}). If $e_{p_1}+e_{p_2} \geq T$, then no attack can be detected by the proposed algorithm (T is the system's total operation time).*

PROOF. Note that the premise implies that no strong inconsistencies can be found. This is true because even if s_{p_1} and s_{p_2} are weakly inconsistent in each round, it is possible that the measurements of s_{p_1} were faulty in the first e_{p_1} rounds and correct in the remaining ones, while the measurements of s_{p_2} were correct initially and faulty in the last e_{p_2} rounds. In this way both sensors would be within their transient fault models, and one cannot conclude that a strong inconsistency (hence, attack) exists.

Since e_{p_1} and e_{p_2} are the smallest bounds on the number of faulty measurements, by using the same argument one can show that no strong inconsistency can exist between any pair of sensors. □

The previous lemma provides a global result regarding the minimum time to detection. While this result relates the number of faults to time, the premise can always be satisfied by allowing a longer operation time. A similar bound exists relating the transient fault model window and the number of faults allowed, which provides a sufficient condition for the impossibility of a strong inconsistency being detected.

LEMMA 5. *For any distinct i and j, if $e_i+e_j \geq \min(w_i, w_j)$, no attack can be detected.*

PROOF. The proof is similar to that of Lemma 4. □

The theoretical bounds developed in this subsection constrain our fault parameter selection approach, which is presented in the following subsection.

5.2 Transient Fault Parameter Selection

When transient fault models are unavailable or unreliable, it is necessary to identify such a model for use in the attack detection approach. Different from algorithms using probabilistic sensors, algorithms over abstract sensor models require that the true value be always contained in the interval, except in the case of a faulty measurement. Thus, statistical approaches to fault parameter selection (e.g., the best-fit Poisson process) are unsuitable since they estimate parameters to be the ones that maximally explain the data, and not provide worst-case bounds. Therefore, in this section, we present a new method for selecting the transient fault model parameters for the purposes of abstract sensor fusion and attack detection.

To empirically identify the transient fault model parameters for a sensor, namely (ϵ, e, w), we proceed as follows. We begin by gathering training measurement data using known values of $\theta^{(t)}$ (e.g., by applying a constant input to an automotive CPS and adjusting for the bias in the input-output speed relation). By examining the data and sliding a window of fixed size w, one can identify the worst-case number of faulty measurements e in a window for different values of ϵ.

Note that intuitively there is a relation between e and ϵ. In particular, if we plot the proportion of the number of faulty measurements in a window (e/w) against ϵ, one should see a few patterns (a few such sample plots for different window sizes are shown in Fig. 5). First of all, there exists a large enough ϵ such that no faulty measurements can ever be observed. As it is decreased, the number of faulty measurements should gradually increase. The rate of increase should be relatively constant while ϵ is in the range of the sensor's transient fault model, i.e., the interval sizes that capture large faulty measurements. As soon as ϵ is decreased past a certain threshold, however, it enters the range of the sensor's noise model, i.e., the range where most of the sensor's measurements lie. At this point (informally referred to as a "knee point"), the number of faulty measurements increases faster as ϵ decreases.

Therefore, it appears that the knee points should be selected as the values for ϵ and e. They are outside of the sensor's noise model so it will not be flagged as compromised when it is simply noisy; on the other hand, they are also smaller than the sensor's fault model, thus most faulty measurements should in fact be declared as permanent faults/attacks. Therefore, for a fixed window size w, the choice of e and ϵ is governed by the knee points. The selection of the window size itself will be discussed and evaluated in Section 6.

Having described expected trends and patterns in this section, we use real data from an unmanned ground vehicle to illustrate the existence of these trends in reality.

6. CASE STUDY

In this section, we evaluate the PI-based attack detector through a case study on the LandShark robotic platform [1], shown in Fig. 6. The LandShark is an electric unmanned ground vehicle, containing many sensors including two encoders for the left and right wheels, and a GPS unit. Each of these sensors can be filtered to provide a vehicle velocity measurement at a minimum rate of 10 Hz. Thus, we evaluate the detector by considering the detection of attacks on the velocity measurements in the presence of real-world transient faults (e.g., tire slip). Specifically, using the LandShark, this section discusses the selection of the transient fault model parameters and evaluates the attack detection and identification performance, respectively.

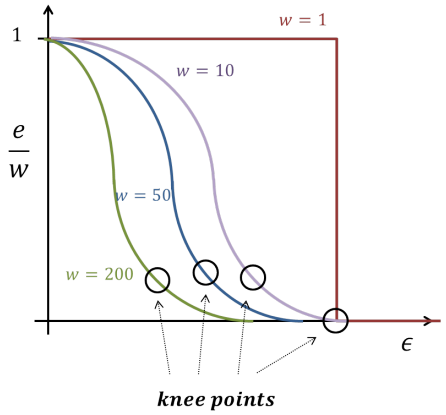


Figure 5: Sample plots of the proportion of faults in a window (e/w) against the error bound (ϵ).

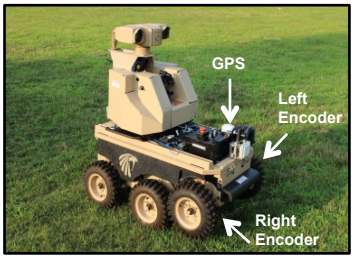
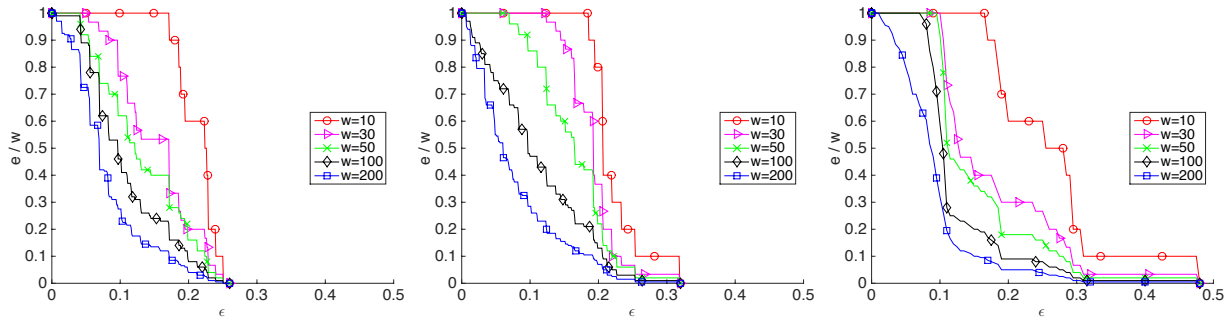


Figure 6: The LandShark robot.

6.1 Fault Model Parameter Selection

For the purposes of evaluation, we select the fault model parameters according to the approach in Section 5. Fault model parameter identification is performed using 4 minutes of training data corresponding to 2400 measurements by each sensor at 10 Hz. The training data is gathered by driving the LandShark in straight lines at a constant speed of 1 m/s on surfaces including grass, asphalt, and snow, where the environment provides potential for transient faults. For example, the LandShark tires occasionally slip in the snow causing a temporary bias in the encoder-inferred velocity until traction is restored. Using this training data, we obtain the real-data equivalents of Fig. 5, as shown in Fig. 7.

Table 2 summarizes the chosen parameters for multiple window sizes. Setup PI_w uses the window size w, which was varied between 10, 30, 50, 100 and 200. Thus, using the plots in Fig. 7 and following the approach outlined in the previous section, the fault model parameters are obtained for the different setups. For example, in GPS (Fig. 7c) for a window of size 50, the knee appears about the point $\epsilon = 0.19$ and $e/w = 0.18$, which corresponds to $e = 9$. Note



(a) Sensor 1: Left Wheel Encoder (b) Sensor 2: Right Wheel Encoder (c) Sensor 3: GPS

Figure 7: Empirical plots of the proportion of faults in a window (e/w) against the error bound (ϵ).

Table 2: Fault models for the sensors on LandShark.

Detector	L. Encoder		R. Encoder		GPS	
	ϵ	e	ϵ	e	ϵ	e
SF	0.26	n.a.	0.32	n.a.	0.48	n.a.
PI_{10}	0.229	2	0.234	2	0.295	2
PI_{30}	0.195	6	0.207	6	0.19	9
PI_{50}	0.195	11	0.199	11	0.19	9
PI_{100}	0.131	26	0.168	22	0.19	9
PI_{200}	0.117	36	0.126	37	0.19	10

Table 3: False Alarm Rate

Detector	SF	PI_{10}	PI_{50}	PI_{200}
False Alarm Rate(%)	0.06	0.64	0.00	0.00

that as the window size increases, the lines become smoother and the knees are more pronounced – this means that in general using longer periods of training data should yield more robust parameters.

To compare the performance of the PI-based detector with the existing SF-based ones, we note that interval sizes are chosen conservatively in the regular SF model because it is designed for the worst case. Thus, in Fig. 7 the smallest ϵ is picked for which no faulty measurements can occur (e.g., 0.26 for the left encoder); thus, it is equivalent to PI_1. This is another reason to introduce a PI-detector – it allows for smaller interval sizes, resulting in more precise estimates.

6.2 Detection Performance

In this subsection, we evaluate the detectors' performance for the fault model parameters selected above using the LandShark robot in the presence of multiple attack scenarios. To evaluate the performance of the detectors, we gathered sensor data from 17 runs of the LandShark where all three sensors were sampled at 10 Hz for 50 seconds on average, corresponding to 500 measurements from each sensor. Various attacks were then added to each sensor's measurements as described in this section.

The **false alarm** rate of each attack detector is evaluated using the data for all 17 runs under no attacks, where the false alarm rate corresponds to the number of incorrect alarms (i.e., since no attacks are present in the false alarm rate test, all of the alarms raised are considered to be incorrect) divided by the total number of tests performed.

Table 4: Detection Rate

Detector	SF	PI_{10}	PI_{50}	PI_{200}
Biased Attack	62.74	99.74	100	100
Random Attack	4.91	36.10	93.30	100
Greedy Attack	0	0.4817	0	0

Note that the first test is performed as soon as w measurements are available; a new test, with a sliding window, is performed whenever a new measurement is received from each sensor. The results for the fault model parameters in Table 2 are shown in Table 3 (PI_{30} and PI_{100} are excluded for the remainder of the paper to avoid clutter). These results indicate that the false alarm rate is zero when $w = 50$ and $w = 200$ but is non-zero when $w = 10$ and $w = 1$, i.e., the SF-based approach. The improvement in the false alarm rate for larger windows is most likely due to the fact that these are indeed transient faults, so they do not occur too often in larger windows. The SF-based approach, on the other hand, has a low false alarm rate due to the conservative way it selects the ϵ's; at the same time, it is non-zero since the largest observed faulty measurement in the training data was smaller than the largest one in the test data.

To evaluate the attack detection rate, we assume the attacker can compromise only one (unknown to us) of the three abstract velocity sensors and consider three types of attacks: (1) bias attack; (2) random attack; (3) greedy attack. The bias attack adds a constant of 0.8 m/s to the attacked sensor, while the random attack adds a uniformly distributed random noise between 0 and 0.8 m/s.[2] The greedy attack replaces the abstract measurement with a new measurement designed to maximize the size of the fusion interval, as discussed in Section 2.1.[3] As formalized in [12], this is a stealthy attack that aims to maximize the uncertainty in the system. Note that the compromised sensor's measurements are attacked in every round, hence every raised alarm in these scenarios is a true alarm.

Employing the same data used to evaluate the false alarm rate, augmented by the attack scenarios described above, the detection rates are calculated for various fault models as shown in Table 4. The detection rate generally improves

[2]The magnitude of the attacks was chosen to be roughly as large as the size of the largest sensor interval, i.e., GPS.

[3]We assume the greedy attack knows the other abstract measurements, as possible if sensor communication occurs on a shared medium, e.g., CAN bus.

with the window size, except for the greedy attack where almost zero detection is recorded; this shows that given enough knowledge and computational power, the attacker can mask his sensors as correct. Note that the SF-based approach detected much less than the PI-based ones due to the conservative way of choosing the sensors' interval sizes.

While the false alarm rate improves with the transient fault window size, for the same reason attack detectors with large-windowed fault models may be slow to detect attacks. Thus, it is natural to evaluate the detection rate (number of alarms divided by the number of tests) vs. the elapsed time since the attack onset. The detection rate vs. elapsed time for the various fault models is shown in Fig. 8, where the steady-state detection rates correspond to the values in Table 4. We observe that for biased or random attacks, the steady-state detection rate improves with the window size and experiences only a marginal increase in the time needed to reach this level of detection accuracy.

To provide a more thorough comparison of the various attack detectors and to examine their robustness to fault modeling errors, we varied the error bounds of the fault model parameters chosen in Section 6.1. In particular, the ϵ's of the three sensors were varied from 50% to 150% of their magnitudes in Section 6.1; we then calculated the false alarm rate and steady-state detection rate for each setup. By studying the robustness of the attack detector with respect to the transient fault parameters, we can qualitatively evaluate the importance of accurate parameter selection. Over these new model parameters we construct the receiver operator characteristic (ROC) curve, which is a classical measure of detector performance, for each window size in Fig. 9.[4]

In Fig. 9, we note that data points which trend towards the upper left corner denote a better detector, i.e., a detector with a larger detection rate and a smaller false alarm rate [20]. One detector is more robust than another (qualitatively) if varying its parameters results in ROC data points which cluster closer to the upper left corner [20]. Thus, the robustness of the PI-based approach generally improves with window size; we note that the performance of PI_{10} is marginally better than that of the SF-based detector, but for larger window sizes the benefits are clear. Finally, the ROC curves in the presence of a greedy attack essentially coincide with the 45° line, meaning that in the presence of the most powerful attacker their performance is not better than a coin flip.

The attack detection results suggest that the detection rate, false alarm rate, and robustness of the PI-based attack detection improve with window size, at a cost in time to detection. Moreover, as the window size increases, the PI-based detector has an increasingly higher detection rate and lower false alarm rate with respect to the SF-based detector.

Finally, we note that the identification performance of the algorithms was almost identical to the detection one. The identification rate generally increases with the window size at the slight cost in time to identification. An avenue of future work is to explore under what scenarios detection could occur without identification such that the attacker may remain stealthy despite the system recognizing that there is a compromised sensor.

[4]Only 13 data points are used to show the general trend and avoid overcrowding on the plot.

7. CONCLUSION

In this paper, we considered the problem of detection and identification of sensor attacks in the presence of transient faults when multiple sensors measure the same physical variable. A novel approach to attack detection was presented based upon transient fault models. When no model is provided by the manufacturer, we demonstrated how to obtain such a worst-case model. The algorithm was evaluated over a robotic application with real-sensor data that indicates that there is a trade-off between correct identification and elapsed time to detection.

Based on the evaluations herein, future work includes the exploration of the qualitative relationships between various parameters and the overall detection and identification performance in hopes of establishing quantitative design metrics. Additionally, we plan to incorporate the abstract sensor detection scheme as a front-end to other resilient estimation schemes in hopes of supplementing their performance and improving their computational complexity.

Acknowledgements

This material is based on research sponsored by DARPA under agreement number FA8750-12-2-0247. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government. This research was supported in part by Global Research Laboratory Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2013K1A1A2A02078326) with DGIST.

8. REFERENCES

[1] Black-I Robotics LandShark UGV. http://blackirobotics.com/LandShark_UGV_UC0M.html.

[2] 'Spoofers' Use Fake GPS Signals to Knock a Yacht Off Course. MIT Technology Review, August 2014.

[3] R. R. Brooks and S. S. Iyengar. Robust distributed computing and sensing algorithm. *Computer*, 29(6):53–60, June 1996.

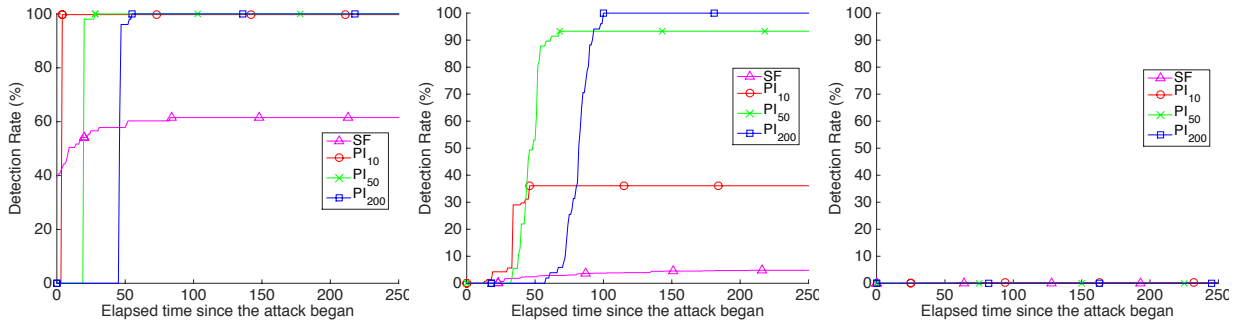
[4] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno. Comprehensive experimental analyses of automotive attack surfaces. In *SEC'11: Proc. 20th USENIX conference on Security*, pages 6–6, 2011.

[5] J. Chen and R. J. Patton. *Robust model-based fault diagnosis for dynamic systems*. Springer Publishing Company, Incorporated, 2012.

[6] N. Falliere, L. O. Murchu, and E. Chien. W32. stuxnet dossier. *White paper, Symantec Corp., Security Response*, 2011.

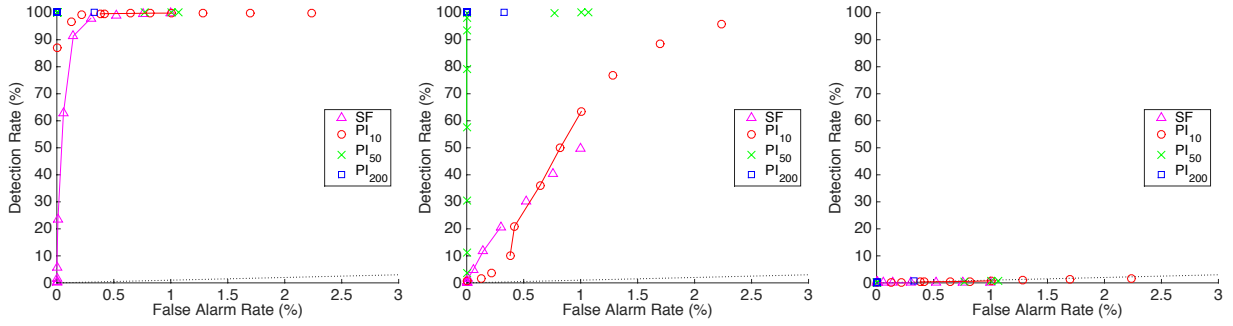
[7] P. M. Frank. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results. *Automatica*, 26(3):459–474, 1990.

[8] P. M. Frank and X. Ding. Survey of robust residual generation and evaluation methods in observer-based



(a) Constant attack. (b) Random attack. (c) Greedy attack.

Figure 8: Time to detection plots under the three classes of attacks.



(a) Constant attack. (b) Random attack. (c) Greedy attack.

Figure 9: Detection Rate vs. False Alarm Rate under the three classes of attacks. Dotted black lines denote 45° lines. Solid lines connect points for a clearer presentation.

fault detection systems. *Journal of process control*, 7(6):403–424, 1997.

[9] G. Frehse, A. Hamann, S. Quinton, and M. Woehrle. Formal analysis of timing effects on closed-loop properties of control software. In *IEEE Real-Time Systems Symposium*, 2014.

[10] I. Hwang, S. Kim, Y. Kim, and C. Seah. A survey of fault detection, isolation, and reconfiguration methods. *Control Systems Technology, IEEE Transactions on*, 18(3):636–653, May 2010.

[11] R. Isermann. Process fault detection based on modeling and estimation methodsâĂŤa survey. *Automatica*, 20(4):387–404, 1984.

[12] R. Ivanov, M. Pajic, and I. Lee. Attack-resilient sensor fusion. In *DATE'14: Design, Automation and Test in Europe*, 2014.

[13] R. Ivanov, M. Pajic, and I. Lee. Resilient multidimensional sensor fusion using measurement history. In *HiCoNS'14: High Confidence Networked Systems*, 2014.

[14] D. N. Jayasimha. Fault tolerance in a multisensor environment. In *SRDS'94: Proc. 13th Symposium on Reliable Distributed Systems*, pages 2–11, 1994.

[15] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[16] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage. Experimental security analysis of a modern automobile. In *SP'10: IEEE Symposium on Security and Privacy*, pages 447–462, 2010.

[17] K. Marzullo. Tolerating failures of continuous-valued sensors. *ACM Trans. Comput. Syst.*, 8(4):284–304, Nov. 1990.

[18] M. Milanese and C. Novara. Set membership identification of nonlinear systems. *Automatica*, 40(6):957–975, 2004.

[19] M. Pajic, J. Weimer, N. Bezzo, P. Tabuada, O. Sokolsky, I. Lee, and G. Pappas. Robustness of attack-resilient state estimators. In *Cyber-Physical Systems (ICCPS), 2014 ACM/IEEE International Conference on*, pages 163–174, 2014.

[20] A. Wald. *Sequential analysis*. Courier Corporation, 1973.

[21] J. S. Warner and R. G. Johnston. A simple demonstration that the global positioning system (gps) is vulnerable to spoofing. *Journal of Security Administration*, 25(2):19–27, 2002.

[22] A. S. Willsky. A survey of design methods for failure detection in dynamic systems. *Automatica*, 12(6):601–611, 1976.

[23] Y. Zhu and B. Li. Optimal interval estimation fusion based on sensor interval estimates with confidence degrees. *Automatica*, 42(1):101–108, 2006.