Security of Cyber-Physical Systems in the Presence of Transient Sensor Faults [1]

Junkil Park*, University of Pennsylvania
Radoslav Ivanov*, University of Pennsylvania
James Weimer, University of Pennsylvania
Miroslav Pajic, Duke University
Insup Lee, University of Pennsylvania
Sang Hyuk Son, Daegu Gyeongbuk Institute of Science and Technology

This paper is concerned with the security of modern Cyber-Physical Systems in the presence of transient sensor faults. We consider a system with multiple sensors measuring the same physical variable, where each sensor provides an interval with all possible values of the true state. We note that some sensors might output faulty readings and others may be controlled by a malicious attacker. Different from previous works, in this paper we aim to distinguish between faults and attacks and develop an attack detection algorithm for the latter only. To do this, we note that there are two kinds of faults – transient and permanent; the former are benign and short-lived whereas the latter may have dangerous consequences on system performance. We argue that sensors have an underlying transient fault model that quantifies the amount of time in which transient faults can occur. In addition, we provide a framework for developing such a model if it is not provided by manufacturers.

Attacks can manifest as either transient or permanent faults depending on the attacker's goal. We provide different techniques for handling each kind. For the former, we analyze the worst-case performance of sensor fusion over time given each sensor's transient fault model and develop a filtered fusion interval that is guaranteed to contain the true value and is bounded in size. To deal with attacks that do not comply with sensors' transient fault models, we propose a sound attack detection algorithm based on pairwise inconsistencies between sensor measurements. Finally, we provide a real-data case study on an unmanned ground vehicle to evaluate the various aspects of this paper.

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

Additional Key Words and Phrases: Cyber-Physical Systems security; sensor fusion; fault-tolerance; fault-tolerant algorithms

[1] Preliminary version of some of the results in this paper appeared in [Park et al. 2015].

Author's addresses: J. Park, R. Ivanov, J. Weimer, and I. Lee are with University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: {park11, rivanov, weimerj, lee}@seas.upenn.edu). M. Pajic is with Duke University, Durham, NC 27708 USA (e-mail: miroslav.pajic@duke.edu). S. H. Son is with Daegu Gyeongbuk Institute of Science and Technology, Daegu 711-873, Korea (e-mail: son@dgist.ac.kr).
* These authors contributed equally to this paper.

DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Many Cyber-Physical Systems (CPS) are vulnerable to security breaches and are increasingly being subjected to attacks. Recent developments have shown that it is possible for an attacker to hijack such a system by exploiting vulnerabilities in its on-board communication protocol [Checkoway et al. 2011; Koscher et al. 2010; Greenberg 2015] or through sensor spoofing [Rutkin 2014; Warner and Johnston 2002]. Weaknesses in Supervisory Control and Data Acquisition (SCADA) systems have also been used as a channel to disrupting critical infrastructure [Falliere et al. 2011]. Therefore, it is imperative that these systems are designed in a secure and resilient fashion so as to guarantee their proper functionality.

As a step towards the design of resilient CPS, in this work we focus on ways to improve their resiliency to sensor attacks. We assume that some of the system's sensors may be compromised by a malicious attacker; the attacker can then send any measurements on behalf of those sensors. As shown in previous work [Rutkin 2014; Warner and Johnston 2002], sensor attacks alone are sufficient to severely affect a system's operation, e.g., drastically disrupt its localization and lead it off the desired course. Thus, the goal of this work is to develop algorithms that guarantee proper system performance even in the presence of sensor attacks.

One way to address this problem is to utilize the increased diversity and reduced price of modern sensing technology, which have made it possible to equip CPS with multiple sensors. Not only can these systems measure variables that were not measured before (e.g., electric currents in batteries) but there also exist multiple sensors that can estimate the same physical variable (e.g., GPS, wheel encoders and IMU's can all provide velocity measurements). Fusing their measurements increases both the robustness to external disturbances (e.g., moving uphill) and the confidence in the obtained estimate [Luo et al. 2002]. This paper shows that such redundant information can also be used to cope with sensor attacks and proposes an attack-resilient sensor fusion framework.

The first aspect to be considered for such a framework is the underlying sensor model. There are two main sensor models used in the literature – *probabilistic* and *abstract*. In the former, each sensor measurement is corrupted by stochastic noise

(e.g., Gaussian [Kalman 1960]). In the latter, an interval is constructed around the measurement containing all possible values of the true state (e.g., set membership approaches [Milanese and Novara 2004]). While the probabilistic approach is well suited for the analysis of the system's expected operation, it requires knowledge of noise distribution, where wrong assumptions may introduce vulnerabilities that can be exploited by an attacker. The abstract model, on the other hand, makes no assumptions about the distribution of the noise and thus naturally lends itself to worst-case analysis.

Since in this paper we consider the problem of CPS security, usually addressed using worst-case reasoning, we adopt the abstract sensor model. Previous works employing the abstract model [Marzullo 1990; Jayasimha 1994] consider the case where the number of faulty sensors (i.e., providing intervals that do not contain the true value) can be bounded; they provide theoretical bounds on the output of sensor fusion in such a scenario. An attack-resilient extension of [Marzullo 1990] has been developed by introducing a sensor transmission schedule [Ivanov et al. 2014a] and by using measurement history [Ivanov et al. 2014b]. A primary limitation of existing fault and attack detection methods [Marzullo 1990; Jayasimha 1994; Ivanov et al. 2014a] is that they treat attacks and faults in the same way. However, it is possible for a sensor to experience a transient fault, i.e., provide wrong measurements for a short period of time and recover on its own, in which case the system should keep using it in the future.

Transient faults are a normal part of a sensor's operation. They are often due to temporary adverse conditions (e.g., a tunnel for GPS) but usually disappear quickly and are not considered a threat for the system's security. Thus, most sensors have a transient fault model that bounds the time in which they can provide wrong data. On the other hand, non-transient (or permanent) faults (e.g., a bias caused by a physically damaged sensor) occur for a longer period of time, and are thus more dangerous and can have catastrophic consequences. If systems cannot compensate for such faults in software, they would benefit from removing the sensor's measurements altogether.

Sensor attacks can manifest as either transient or non-transient (possibly Byzantine) faults, depending on the attacker's goals and capabilities. Masking a sensor's measurements as a transient fault may prevent the attacker from being discovered

but limits his capabilities. On the other hand, if the attacked measurements are consistently wrong and resemble a permanent fault, they may inflict more damage but may be detected quickly. In this paper, we analyze different kinds of attacks and their possible effect; we propose a detector for the more dangerous, but easier to detect, kind of attacks and a filtering algorithm whose output is robust to more stealthy attacks.

To distinguish between attacks and faults and to quantify the stealth of an attack, we make use of sensor transient fault models (TFMs) that are now being provided by some manufacturers [Frehse et al. 2014]. Such a model consists of three dimensions: (1) interval size, (2) window size, and (3) number of allowed faulty measurements per window. At the same time, such specifications are not always available, so the first contribution of this paper is a method for selecting the three parameters based on observed training data. We illustrate this with a real-data study using a ground vehicle called the LandShark [Black-I Robotics 2015].

Once such TFMs are available, we examine their effect on the performance of sensor fusion over time (as described in Section 3, we adopt a sensor fusion algorithm initially developed by [Marzullo 1990] and extended by [Ivanov et al. 2014a; 2014b]). We present results showing what is the worst-case number of rounds, in which sensor fusion cannot make any guarantees about its output. We then provide a filtered sensor fusion algorithm that is robust to this worst case and outputs a conservative, but bounded, interval that is guaranteed to contain the true value. This worst-case result also holds in the presence of attacks that appear as transient faults.

To deal with the more immediately disruptive class of attacks that manifest as permanent faults, we propose a detection and identification algorithm for sensors that do not comply with their TFM's. The algorithm uses pairwise relationships between sensors – if two sensors' measurements are too distant from each other, then one of them must be wrong. By accumulating this information over time, we develop a sound algorithm for attack detection and identification.

Finally, we illustrate the performance of the proposed solutions on a real-data case study using the LandShark. In particular, we show the performance of the attack detection/identification algorithm in the form of false alarm and detection rates and show

its advantage over the current sensor fusion technique. We also analyze the effect of the TFM on sensor fusion and show the benefit of the filtered fusion interval.

To summarize, the contributions of this work are as follows: (1) a sensor attack detection/identification algorithm in the presence of transient faults using the abstract sensor model; (2) a framework for selecting the TFM parameters based on training data; (3) an analysis of the effect of TFM's on the performance of sensor fusion and the introduction of the filtered fusion interval; and (4) a case-study evaluation of all the contributed solutions on a robotic platform.

2. RELATED WORK

This section describes related work on fault and attack detection with different sensor models, contrasting the attack detection methods to the traditional fault detection ones.

2.1. Sensor Model

For the detection of sensor faults and attacks, the first thing to consider is the underlying sensor model because different sensor models lead to different approaches to detection. Most sensor models used in the literature fall into two general categories: probabilistic sensor models and abstract sensor models. The probabilistic sensor models assume a probability distribution of the sensor noise (e.g., Gaussian [Kalman 1960]). The noise distribution puts more weight on the points which are more likely to be the true value. Thus, probabilistic sensor models are well suited for the analysis of the expected system performance [Kalman 1960; Xiao et al. 2005], but require knowledge of noise distribution, where detectors designed under wrong noise assumptions are well-known to have decreased (attack) detection accuracy [Willsky 1976]. On the other hand, in the abstract sensor model, a measurement is an interval which contains all points that may be true value (e.g., set membership approaches [Milanese and Novara 2004]). This type of sensor models assumes no knowledge of the noise distribution on the interval, but construct such intervals to contain the unknown true value even in the worst case. Since abstract measurements are the worst case bounds for the true

value, they are well-suited for worst case analysis of system operation under Byzantine faults and sensor attacks [Marzullo 1990; Ivanov et al. 2014a; 2014b].

2.2. Fault Detection

There exists a large body of literature in the sensor fault detection and isolation domain with probabilistic sensors, including multiple well-written and exhaustive survey papers, e.g., [Chen and Patton 2012; Frank 1990; Frank and Ding 1997; Hwang et al. 2010] and references within. The common theme among fault detection techniques is that they either assume a prior on the initial condition [Isermann 1984] or a certain model in which faults could occur [Joshi et al. 2011; Jiang et al. 2006]. The former type is referred to as a "change detector" in which large deviations from expected behavior are flagged as faults [Basseville et al. 1993]. In the latter, the assumed failure models include jump systems [Davis 1975], probability of occurrence of faults [Willsky 1976] (or probability of missing measurements altogether [Sinopoli et al. 2004]), and known directions in the state space where faults may occur [Willsky 1976]. Approaches also exist to distinguish between transient and persistent faults, e.g., [Serafini et al. 2007; Lee and Choi 2008; Serafini et al. 2011], and to provide trust assessment of sensors [De Kerchove and Van Dooren 2010; Rezvani et al. 2015]. Finally, a powerful technique is the utilization of sensor redundancy [Kim et al. 2010].

With the abstract sensor model, fault detection is usually performed by using sensor redundancy and a voting system on the provided intervals [Marzullo 1990; Jayasimha 1994]. While some algorithms are more conservative and provide guarantees that the output of sensor fusion contain the true value (hence, their fault detection performance may suffer), others relax these worst-case guarantees in favor of better performance [Brooks and Iyengar 1996]. Other works also suggest imposing a distribution on the provided interval so that a hybrid abstract-probabilistic analysis may be performed [Zhu and Li 2006]. Finally, in addition to the one-dimensional intervals that are assumed by the above, sensors can be also assumed to provide multidimensional hyper-rectangles [Chew and Marzullo 1991], polyhedra [Ivanov et al. 2014b], or more general sets [Milanese and Novara 2004; 2011].

2.3. Attack Detection

In contrast to traditional fault detection and isolation, attack detection works in general involve worst-case analysis [Fawzi et al. 2011] because wrong model assumptions can introduce vulnerabilities that can be exploited [Pajic et al. 2014]. Attack detection papers try to minimize the prior assumptions in an attempt to address a wider variety of possible attacks; thus, the abstract sensor model (or more generally, the set membership model) is almost exclusively the model of choice, except when the initial condition and system dynamics are known [Rezvani et al. 2015; Teixeira et al. 2012]. In the abstract sensor domain, an attack-resilient extension to [Marzullo 1990] has been developed [Ivanov et al. 2014a; 2014b]. [Ivanov et al. 2014a] introduces the use of sensor transmission schedules in order to limit the attacker's available information, thus reducing the attacker's capabilities. [Ivanov et al. 2014b] improves the accuracy of sensor fusion by incorporating historical measurements, thus providing a better attack detection. A major shortcoming of the attack detection works [Marzullo 1990; Jayasimha 1994; Ivanov et al. 2014a] is that they conservatively treat mere faults as attacks. In our preliminary work [Park et al. 2015], we presented an attack detection method differentiating transient faults from attacks based on the transient fault model which will be defined later herein. This paper extends our initial study [Park et al. 2015] and our previous work on incorporating measurement history [Ivanov et al. 2014b] by adding a new sensor fusion algorithm considering the effect of the transient fault model, thus providing an overall framework for attack detection in the presence of transient faults.

3. PROBLEM FORMULATION

This section presents the problems considered in this work. It begins by explaining the system and sensor models, including the transient fault model. It then introduces attacks and analyzes the possible means and goals of an attacker. Finally, the problem statements are presented.

3.1. System Model

The considered system consists of n sensors that can be used to estimate the same physical variable (e.g., velocity). The system is run in a periodic fashion during T time rounds (the total time of the system's operation) – at each round sensors transmit their measurements to a centralized estimator; it then performs sensor fusion and attack detection/identification.

As described above, we adopt the abstract sensor model, in which each sensor provides an interval of possible values [Marzullo 1990]. For each sensor s_i, the interval is constructed symmetrically around its measurement at time t, denoted by $y_i^{(t)}$; the interval's length is twice the size of s_i's error bound, ϵ_i, which can be obtained from manufacturer guarantees and physical limitations such as sampling jitter [Frehse et al. 2014]. However, due to external disturbances (e.g., rough terrain) and other reasons, sensors sometimes provide *faulty measurements* that are outside of their predefined bounds, hence their intervals may not contain the true value. We formally define the predicate $F(i,t)$ of sensor index i and time t as follows, to denote that sensor i provides a faulty measurement at time t:

Definition 3.1 (*Faulty Measurement*). For any sensor index $1 \leq i \leq n$ and time $1 \leq t \leq T$,

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

where $\theta^{(t)}$ is the true value. We say that sensor i provides a faulty measurement at time t iff $F(i,t)$ holds.

3.2. Transient Fault Model

By their nature, faulty measurements occur infrequently and usually do not indicate a permanent problem with the sensor. To reflect this feature and motivated by recent manufacturer trends to provide *faulty-measurements-per-window* specifications [Frehse et al. 2014], we introduce the notion of a sensor's transient fault model (TFM). A TFM for a 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 output at most e_i faulty measurements in any window of w_i measurements. If s_i complies with its TFM, then any faulty

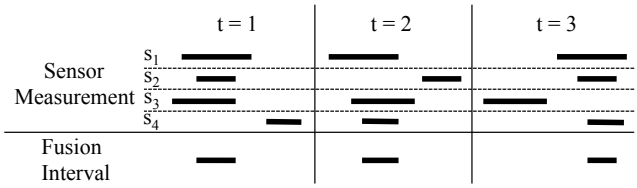


Fig. 1: An example of fusion intervals for three different rounds.

measurements are deemed transient faults. Otherwise, it is *non-transiently faulty*, denoted by the predicate $NTF(i,t)$ of sensor index i and time t.

Definition 3.2 (*Non-Transiently Faulty Sensor*). For any sensor index $1 \leq i \leq n$ and time $1 \leq t \leq T$,

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

where $F_1(i,t) = 1$ if $F(i,t)$, and $F_1(i,t) = 0$ if $\neg F(i,t)$. We say that sensor i is non-transiently faulty at time t iff $NTF(i,t)$ holds.

3.3. Sensor Fusion

Since the true value is not known in practice, one can use the redundant sensor information to get an estimate of where it might be. One of the first works to develop sensor fusion with abstract sensors [Marzullo 1990] assumes an upper bound on the number of faulty measurements at any round, f, and outputs a *fusion interval* that is guaranteed to contain the true value. The fusion interval is computed at each round as the smallest interval that contains all points that lie in at least $n - f$ intervals. Intuitively, since there are at least $n - f$ correct intervals, the true value may lie in any such group, thus all of them are included. Fig. 1 presents example fusion intervals in a system with four sensors and $f = 1$. At each round, the fusion interval is the smallest interval containing all points that lie in at least three intervals.

The fusion interval is used for worst-case analysis. For instance, the system is considered safe if the fusion interval does not contain any undesired states (since it is guaranteed to contain the true value). Two results that will be used throughout this paper are as follows: if $f < \lceil n/2 \rceil$ then the fusion interval is bounded by the size of

some sensor's interval; otherwise, the fusion interval can be arbitrarily large [Marzullo 1990].

3.4. Attack Model

In this work we assume that sensors may not only provide faulty measurements but may also be compromised by attacks. A malicious attacker may gain control of a sensor and send any measurement on its behalf. This subsection describes how and why a sensor attack might be performed and what its effect on system perfomance may be.

3.4.1. Attack Goals and Means. CPS may be subject to sensor attacks for a variety of reasons. The advancement of robotics research has made it possible to use unmanned vehicles to perform critical missions on enemy territory; as shown in the case of the RQ-170 Sentinel drone captured in Iran [Peterson and Faramarzi 2011; Shepard et al. 2012], it is possible to compromise these systems' sensors and disrupt their operation. Additionally, as described in [Checkoway et al. 2011; Koscher et al. 2010], an attacker (e.g., a former employee or a market competitor) could greatly disrupt the performance of a modern automobile by corrupting a single electronic control unit. In both cases, sensor attacks could lead to a complete takeover or even destruction of the respective system.

There are (at least) two ways for an attacker to compromise a given sensor. The first is through a physical attack, e.g., unplugging a sensor and replacing its software or using other physical means [Shoukry et al. 2013]. Additionally, modern sensors have complex software modules that may be vulnerable to cyber attacks through weaknesses exposed in code (e.g., buffer overflow). Thus, as discussed in [Checkoway et al. 2011; Koscher et al. 2010], an attacker could gain access to a sensor over the network without even requiring physical proximity.

At the same time, we argue that it may not be possible to compromise all sensors on a given system. Some sensors may be more difficult to compromise than others. In particular, certain sensors are attached to other platforms (e.g., wheel encoder) and cannot be tampered without affecting critical components (e.g., the entire wheel); however, such actions could be detected and reported by an on-board diagnostics system. Similarly, cyber attacks require significant efforts and knowledge of a sensor's spe-

cific implementation. Therefore, in this work we assume that some sensors might be malicously attacked while others are not attacked but may be sometimes transiently faulty.

3.4.2. Attack Detection and Consequences. Whether and when an attack is detected depends first and foremost on the detection algorithm used by the system. As argued above and in Section 2, in this work we use the abstract sensor model and the sensor fusion algorithm presented in Section 3.3.

The way attacks are detected in this algorithm is by checking if an interval intersects the fusion interval. Any interval not intersecting the fusion interval cannot contain the true value (since the true value lies in the fusion interval). Thus, in Fig. 1 sensors s_4, s_2 and s_3 would be detected as attacked in rounds 1, 2 and 3, respectively. The limitation of this algorithm is that it treats attacks and faults in the same manner. While it may be true that s_4 is attacked in Fig. 1, it may also be the case that it experienced a transient fault but recovered shortly afterwards.

This poses the question of how to formalize attacks and distinguish them from faults. Of course, for any definition of a fault, it is possible for an attacker to mask his measurements as a fault in order to avoid detection; it is even possible for the attacker to just relay the actual sensor measurements. Therefore, in this paper we focus on the detection of attacks that manifest as the most disruptive kind of faults, namely non-transient faults.

Attacks that manifest as transient faults also pose a threat to the system, hence they are addressed as well. However, rather than detecting such attacks, which would also cause unnencessary detecions of transient faults, we use the fact that they are actually bounded by the definition of transient faults, i.e., at most e_i faulty (or attacked) measurements can be provided in any window of size w_i. By utilizing this information, we can develop both an algorithm for the detection of attacks that manifest as non-transient faults and an algorithm for deriving a fusion interval that is robust to the stealthy attacks that appear as transient faults. To avoid confusion, in the remainder of the paper an attacked sensor is equivalent to a non-transient fault only. Thus we formally the predicate $A(i)$ of sensor index i as follows, to denote that sensor i is attacked:

Definition 3.3 (*Attacked Sensor*). For any sensor index $1 \leq i \leq n$,

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

We say that sensor i is attacked iff $A(i)$ holds.

3.5. Problem Statements

There are three problems addressed in this work. The first one arises from the fact that TFM's are not widely available for current sensors and are not straightforward to obtain.

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.*

We note that once a TFM is introduced, the analysis of sensor fusion changes as well. In particular, the assumption that at most f sensors provide faulty measurements in a given round cannot be justified since it is possible that all sensors provide faulty measurements in one round and are all correct in the next without violating their respective TFM's. Note that in this problem attacks are not yet considered.

PROBLEM 2. *Given a system with n sensors and a transient fault model (ϵ_i, e_i, w_i) for each sensor, analyze the performance of sensor fusion over time.*

Finally, we introduce sensor attacks and develop an algorithm for attack detection and identification.

PROBLEM 3. *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 and possibly identify which sensor is under attack.*

4. TFM PARAMETER SELECTION

This section describes a framework to choose the TFM parameters. As mentioned earlier, manufacturers are transitioning towards providing transient fault specifications for their sensors to allow for more realistic analysis [Frehse et al. 2014]. However, when the TFM of a sensor is not provided, it is necessary to identify the TFM param-

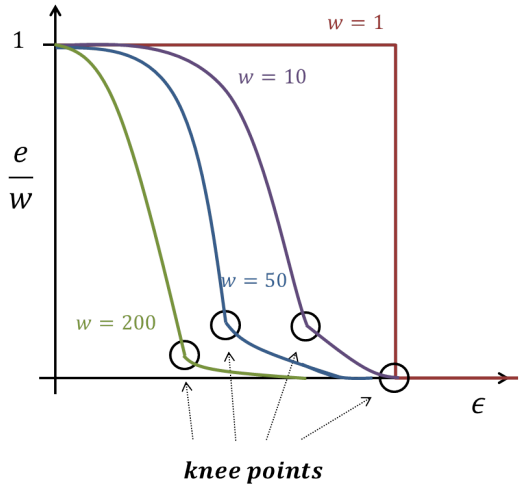


Fig. 2: Sample plots of the proportion of faults in a window (e/w) against the error bound (ϵ).

eters from empirical data. Unlike probabilistic sensor models, abstract sensor models are required to contain the true value in the interval except in the case of a faulty measurement. Thus, statistical approaches to parameter selection (e.g., the best-fit Poisson process) are unsuitable because they estimate parameters to maximally explain the data, thus not providing worst-case bounds. Therefore, we provide a new method for selecting the TFM parameters from empirical data.

To empirically identify the TFM parameters, we apply the following procedure. First, we gather sensor measurements with known true value $\theta^{(t)}$ as training data (e.g., by applying a constant input to an automotive CPS and adjusting for the bias in the input-output speed relation). Next, we examine the data and identify the set of feasible parameters (ϵ, e, w) by sliding a window of size w and finding the worst-case number of faulty measurements e in a window for different values of ϵ.

For a fixed window size w, intuitively, there exists a relation between ϵ and e. Suppose that we plot the proportion of the number of faulty measurements in a window (e/w) against ϵ (Fig. 2 shows possible examples of such curves for different window sizes). Then, there can be observed a few interesting patterns. To begin with, there is a large enough ϵ such that no faulty measurements can ever be observed (i.e., $e = 0$). As ϵ is decreased from that point, the number of faulty measurements should slowly in-

crease. The increase rate should be relatively moderate while ϵ is in the range of underlying true TFM. In other words, e increases in a relatively constant rate as ϵ decreases, because ϵ gradually excludes more faulty measurements that occur transiently. Once ϵ passes a certain threshold, it enters the range of the underlying noise model where most of the sensor measurements lie. Thus, as ϵ decreases from this threshold, the number of measurements that are deemed faulty increases rapidly. We refer to the threshold as a "knee point".

We argue that the knee points should be selected as the values for the TFM. On the one hand, they are outside of the sensor's underlying noise model, thus not making noisy measurements be flagged as faulty. On the other, they are smaller than the sensor's underlying TFM, thus forcing most faulty measurements to be declared as such. Consequently, the knee points govern the choice of ϵ and e for any window size w. Note that the right window size depends on the purpose for which it is used; a larger window size will better capture the true TFM; as will be apparent, however, sometimes it may increase the time necessary to detect an abnormality. Section 7 provides a real-data illustration of this process.

5. EFFECT OF TFM ON SENSOR FUSION PERFORMANCE

Having provided a framework for selecting the TFM parameters, in this section we analyze their effect on the worst-case performance of sensor fusion. In addition, we illustrate how to compute a filtered fusion interval that is robust to this worst-case scenario. We only consider stealthy attacks in this analysis – the theory on attack detection/identification is presented in the next section. Thus, in this section we assume that all sensors comply with their respective TFM's.

5.1. Precision vs. Accuracy of the Fusion Interval

We begin by noting that the assumption of at most f faulty measurements per round that is required in the original sensor fusion algorithm no longer holds. This is due to the fact that each TFM only quantifies one sensor's output in isolation from the others. Thus, it is possible that all sensors[2] provide faulty measurements in a single round or

[2]Only possible if all sensors have $e > 0$.

that all are correct in a single round. Therefore, f can now be considered as an input parameter to the fusion algorithm as opposed to a preliminary assumption. Note that if f is smaller than the actual number of faulty measurements per round, the resulting fusion interval may not contain the true value.

The chosen value of f introduces a trade-off between accuracy and precision of the fusion interval. In particular, decreasing f will result in a smaller (i.e., more precise) fusion interval in any given round. On the other hand, it may increase the proportion of rounds where the fusion interval does not contain the true value (i.e., less accurate), in which case a more conservative value of f would be required. Therefore, in this section we provide a way of quantifying the effect of the value of f on the performance of sensor fusion.

To formalize these statements, suppose that we are given a TFM for each sensor. Since we consider a periodic system in which sensors are sampled at the same rate, in this section we assume that window sizes are the same for all sensors, i.e., the TFM's have the form (ϵ_i, e_i, w). Define a *global fault* as a round in which there are more than f faulty measurements. Recall that in such a case the fusion interval is not guaranteed to contain the true value.

Definition 5.1 $(\textit{Global Fault})$. Given the maximum number of faulty measurements f, for any time t,

$$GF(t) \equiv \left(\sum_{i=1}^{n} F_1(i,t) \right) > f.$$

The goal is to find a global fault model $(E, W)_f$ for the entire system in which there are at most E rounds with a global fault in any window of W rounds. The fault model will determine how robust (and consequently, conservative) any filtering algorithm has to be in order to produce a meaningful output. Note that the value of $(E, W)_f$ depends on the sensors' TFM but not on the actual sensor measurements, even if they are faulty; hence, this result holds even in the presence of stealthy attacks that comply with the sensors' TFM.

Obtaining a closed-form solution for the values of E and W is made difficult by the combinatorial nature of the problem. Therefore, we have derived an algorithm that,

Algorithm 1 Computing the Global Fault Model of Sensor Fusion

Input: n transient fault models of the form (ϵ_i, e_i, w) and sensor fusion parameter f

1: $W_R \leftarrow w$
2: $E_S \leftarrow order\_descending(\bigcup e_i)$
3: $E \leftarrow 0$
4: **while** $W_R > 0$ **and** $E_S(f+1) > 0$ **do**
5: **for** $\{i \leftarrow 1;\ i \leq f+1; i \leftarrow i+1\}$ **do**
6: $E_S(i) \leftarrow E_S(i) - 1$
7: **end for**
8: $E_S \leftarrow order\_descending(E_S)$
9: $W_R \leftarrow W_R - 1$
10: $E \leftarrow E + 1$
11: **end while**
12: $W \leftarrow w$
13: **return** (E, W)

given the TFM's and f as input, outputs E and W. As formalized in Algorithm 1, it computes the largest possible number of rounds in which at least $f+1$ faulty measurements can occur; this is the largest number of rounds in which the fusion interval is not guaranteed to contain the true value. Intuitively, at each round the algorithm "schedules" faulty measurements for the sensors that have the largest number of "allowed" faulty measurements until the end of the window.

THEOREM 5.2. *The output, E, of Algorithm 1 is the largest number of global faults possible in a window of size W.*

PROOF. The proof of optimality mirrors the proof of optimality of the Earliest Deadline First (EDF) scheduling algorithm. Suppose there exists a schedule s that is better than the proposed here. Then s contains a round t in which a sensor s_i produces a faulty measurement and sensor s_j does not, even though s_j has more "unused" faulty measurements.

Suppose s_j's next scheduled faulty measurement according to s is at time $k > t$. Without loss of generality, we can assume s_i does not have a faulty measurement at k.[3] Then by swapping s_j and s_i's faulty measurements, i.e. making s_i's measurement faulty at time k and s_j's faulty at time t, we do not affect the magnitude of E (since the number of faulty measurements in each round remains the same). By replacing all

[3]Since s_j has more remaining faulty measurements, there exists a time k when s_j provides a faulty measurement and s_i does not. If no such time exists, then we can remove the "scheduled" faulty measurement by s_i at time t and replace it with a faulty measurement by s_j (still within its TFM).

such pairs we eventually transform s into a new schedule s' that is exactly the schedule suggested by the proposed algorithm here. Therefore, Algorithm 1 is optimal. □

Note that Algorithm 1 is polynomial in the number of sensors, n, and is pseudo-polynomial in the window size, w. At the same time, we note that it is executed offline, at design stage, hence the execution time will not be prohibitive even for very large window sizes. To inspect which choice of f is best suited for a given system, designers need to take into account Algorithm 1 and its output. Comparing different pairs $(E, W)_f$ may not always be possible in a quantitative way but an analysis similar to that of Fig. 2 may be performed so that the best combination of accuracy vs. precision is chosen.

5.2. Filtered Fusion Interval

In this subsection we describe how, given a pair $(E, W)_f$ and W rounds with a fusion interval computed in each, one can derive a *filtered fusion interval* that is guaranteed to contain the true value and is bounded in size. The filtered fusion interval can be thought of as the system's conservative, but correct, guess of its current state – since it does not trust its last fusion interval, it examines the historical fusion intervals to improve this estimate. To do this, we assume that the system has a known dynamical model, up to additive noise, of the form:

$$x_{t+1} = g(x_t) + w_t, \tag{1}$$

where $x \in \mathbb{R}$ denotes the system's state (e.g., velocity), $g(\cdot)$ is the transition function and w is bounded process noise, i.e., $\|w\| \leq M$ for some positive M. It is assumed that each y_i is a direct (possibly faulty) measurement of x.

Given this model, each fusion interval can be mapped from time t to $t+1$ [Ivanov et al. 2014b]. For instance, let $I = [a, b]$; then the mapping of I to the next round is

$$m(I) = \{p \mid p = g(q) + n, \forall q \in [a, b], |n| \leq M\}.^4$$

[4]This definition is implicitly assuming g is continuous on the region $[a, b]$. If that is not true, the convex hull of the mapping needs to be taken as well.

Algorithm 2 Filtered Fusion Interval

Input: transition function g, an array FI containing W fusion intervals (in chronological order) and a bound E on the number of global faults

1: $FI_C \leftarrow \emptyset$
2: **for** $\{i \leftarrow 1;\ i \leq W-1; i \leftarrow i+1\}$ **do**
3: $\quad mapped\_I \leftarrow m(m(\cdots m(FI(i))))$ // map i times
4: $\quad FI_C.add(mapped\_I)$
5: **end for**
6: $FI_C.add(FI(W))$
7: **return** $sensor\_fusion(FI_C, E)$

It is now possible to design an algorithm to compute the filtered fusion interval at time t using the last W fusion intervals.

The proposed algorithm is formalized in Algorithm 2. Essentially, all fusion intervals are mapped, using g, to the current time t, thus obtaining W intervals at t. Then we apply the original sensor fusion algorithm – since at most E mapped intervals are faulty, we output the smallest interval that contains all points that lie in at least $W-E$ mapped intervals. Thus, a filtered fusion interval is computed that is a conservative, but bounded, estimate of the system's current state.

We note that Algorithm 2 does not always produce the smallest possible interval that is guaranteed to contain the true value. On other hand, it is efficient and can be implemented in real time whereas it is difficult to obtain an algorithm that outputs such an interval and is not exponential in the number of sensors and rounds. Finally, Algorithm 2's output is guaranteed to contain the true value and is bounded (provided $E < \lceil W/2 \rceil$), so it is still in the spirit of worst-case analysis.

6. A SOUND ALGORITHM FOR ATTACK DETECTION AND IDENTIFICATION

In this section we introduce attacks and describe our approach to their detection and identification, which aims to differentiate sensor attacks from mere transient faults given each sensor's TFM. It is based on Pairwise Inconsistencies (PI's) between two sensors. Two types of PI's are the key concepts of our approach: *weak inconsistency* and *strong inconsistency*. We accumulate the information of strong inconsistencies over time in order to utilize it for attack detection and identification. In the following subsections, we first define each type of inconsistency and then present the attack

detection/identification method. We conclude with a discussion on the conditions on the TFM parameters under which our approach can operate.

6.1. Weak and Strong Inconsistency

This section is built on the premise that the true value $\theta^{(t)}$ is unknown in general. Thus, it is not always known which sensors have provided correct measurements. However, we know how correct sensor measurements should relate to each other, and mainly use this mutual information in our approach. The first relation 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 6.1 (*Weak Inconsistency*). For any sensor indices i and j, and any time t,

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

We say that sensors i and j are weakly inconsistent at time t iff $WI(i, j, t)$ holds.

Since weak inconsistency is defined upon the unknown true value $\theta^{(t)}$, it is impossible to decide weak inconsistency in general. However, there exists a useful sufficient condition. If the intervals of two sensors do not overlap each other, one of them must have provided a faulty measurement because the true value cannot lie in both the intervals. This condition is formally stated in the following lemma:

LEMMA 6.2. *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. □

Note that both transient faults and attacks can cause weak inconsistency in a round. Thus, to disambiguate between transient faults and attacks, we introduce another relation between two sensors, namely strong inconsistency, denoted by the predicate $SI(i, j, t)$. Two sensors are strongly inconsistent if and only if one of them is non-transiently faulty (i.e., it does not comply with its transient fault model).

Definition 6.3 (*Strong Inconsistency*). For any sensor indices i and j, and any time t,

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

We say that sensors i and j are strongly inconsistent at time t iff $SI(i, j, t)$ holds.

Similar to weak inconsistency, strong inconsistency cannot be decided in general. However, there exists a sufficient condition again. If two sensors are weakly inconsistent more times than a certain threshold in a window, they become strongly inconsistent.

LEMMA 6.4. *Given i, j, t,*

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

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 notions of pairwise inconsistency in this subsection form a basis for the attack detection and identification techniques to be explained in the following subsection.

6.2. Attack Detection and Identification

In this subsection, we describe our approach to attack detection/identification using the notions of weak and strong inconsistency. An attack is *detected* when there exist two sensors which are strongly inconsistent because one of them must be non-transiently faulty. An attacked sensor is *identified* if it is strongly inconsistent with multiple sensors. These statements are formalized in the remainder of this subsection.

To propagate the strong inconsistencies over time, we use a sequential detection approach (motivated by sequential detection theory [Wald 1973]) and accumulate the information over time. We use the predicate $SI^*(i,j)$ to denote that there exists a time $t \leq T$ when sensors s_i and s_j are strongly inconsistent.

Definition 6.5 (*Accumulated SI*). For any sensor indices i and j,

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

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

Note that accumulated strong inconsistency between two sensors implies that one of the two sensors is attacked.

LEMMA 6.6. *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
$(\exists t, NTF(i,t)) \vee (\exists t, NTF(j,t)) \implies A(i) \vee A(j)$. □

We now formalize attack detection using accumulated strong inconsistency; there exists a sensor attack if any pair of sensors have ever been strongly inconsistent.

THEOREM 6.7. $(\exists i, \exists j, SI^*(i,j)) \implies \exists i : A(i)$.

PROOF. Let s_i and s_j be two sensors that satisfy $SI^*(i,j)$. By Lemma 6.6, this means $A(i) \vee A(j)$, and the Theorem statement follows. □

We now formalize the attack identification approach. Note that it is necessary to assume that at most a sensors are attacked such that $a < n-1$. To explain the need for the assumption, suppose that sensor s_i is strongly inconsistent with all other sensors. Without the assumption on a, it is impossible to declare that s_i is attacked because s_i could be correct and all other sensors could be attacked. Note that a should be less than $n-1$ because otherwise no sensor could be identified as attacked even if every pair of sensors are strongly inconsistent. When $a < n-1$, there is a sufficient condition to identify attacked sensors.

THEOREM 6.8. *Let $d(i)$ denote the size of set $\{j \mid SI^*(i,j)\}$, i.e., the number of sensors that have been strongly inconsistent with s_i during the system's operation. Then, assuming $a < n-1$,*

$$d(i) > a \implies A(i).$$

PROOF. Suppose for a contradiction that s_i is not attacked. It follows that the $d(i) > a$ sensors which are strongly inconsistent with s_i must be attacked. This is a contradiction because there are at most a attacks. □

Lastly, we note that there exists a constraint on the TFM parameters governing the feasibility of our PI-based approach. The following lemma provides a sufficient condition for the impossibility of attack detection by the PI-based method:

LEMMA 6.9. *If $e_i + e_j \geq \min(w_i, w_j)$ for all distinct i and j, then no attack can be detected by our approach.*

PROOF. Note that the premise implies that no strong inconsistency can be found between any pair of sensors. This is true because even if s_i and s_j are weakly inconsistent in each round, it is possible that the measurements of s_i were faulty in the first e_i rounds and correct in the remaining ones, while the measurements of s_j were correct initially and faulty in the last e_j rounds. In this way both sensors would be within their TFM's, and one cannot conclude that an attack exists. □

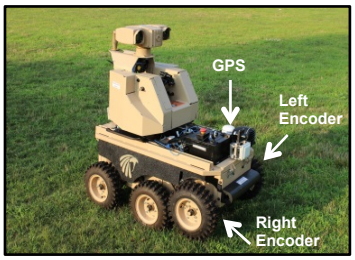


Fig. 3: The LandShark robot.

Table I: Fault models for the sensors on LandShark

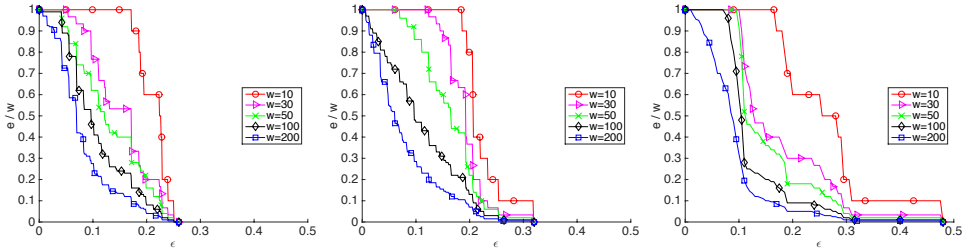
Detector	L. Encoder		R. Encoder		GPS	
	ϵ	e	ϵ	e	ϵ	e
SF	0.26	n.a.	0.32	n.a.	0.48	n.a.
PI_{10}	0.229	2	0.234	2	0.295	2
PI_{30}	0.195	6	0.207	6	0.19	9
PI_{50}	0.195	11	0.199	11	0.19	9
PI_{100}	0.131	26	0.168	22	0.19	9
PI_{200}	0.117	36	0.126	37	0.19	10

7. CASE STUDY

In this section, we evaluate the performance of the different aspects proposed in this work through a case study on the LandShark robotic platform [Black-I Robotics 2015] shown in Fig. 3. The LandShark is an electric unmanned ground vehicle, which contains various sensors including left and right wheel encoders and a GPS unit. Each of these sensors can be filtered to provide a velocity measurement at a rate of 10 Hz. Thus, we use the redundancy of velocity measurements to evaluate the proposed techniques in the presence of transient faults (e.g., tire slip).

7.1. Transient Fault Model Parameter Selection

This subsection illustrates the selection of the TFM parameters following the method described in Section 4. First, we collect the training data by driving the LandShark straight at a constant speed of 1 m/s on the different surfaces such as grass, asphalt and snow, where the environment may cause transient faults (e.g., slipping tires would mean encoders provide higher-than-actual velocity). The gathered training data corresponds to 2400 velocity measurements by each sensor at 10 Hz (i.e., about four minutes). By examining the training data, we obtain Fig. 4, which is the real-data equivalent of Fig. 2.



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

Fig. 4: Empirical plots of the proportion of faults in a window (e/w) against the error bound (ϵ).

Table I summarizes the chosen parameters, where setup PI_w uses a window size w for all three sensors and w is varied between 10, 30, 50, 100 and 200. For example, for $w = 50$ in GPS (Fig. 4c), the knee point appears around $\epsilon = 0.19$ and $e/w = 0.18$, corresponding to $e = 9$. Note that the knee points are more clearly visible as the window size increases.

Finally, we note that the original sensor fusion (SF) approach would use the most conservative error bounds (interval sizes) because it is designed for the worst case. Specifically, in Fig. 4, we select the smallest ϵ such that no faulty measurements can be observed (e.g., 0.48 for GPS). Note that the parameters for SF would be equivalent to PI_1. We observe that one benefit of using TFM is that as the window size increases, the size of error bounds is generally reduced, thus allowing more precise sensor fusion (e.g., PI_{200} is more than twice smaller than SF in the size of error bounds).

7.2. Sensor Fusion Performance

We now examine the effect of TFM on sensor fusion performance employing the TFM parameters selected above. Note that no attacks have been introduced yet. As discussed in Section 5, there exists a trade-off between the precision and the accuracy of the fusion interval depending on the choice of f. Thus, we evaluate these metrics using the LandShark data and the selected TFM parameters for different window sizes.

To do this, we proceed as follows: we first collect test data from 17 runs of the LandShark, each consisting of about 500 velocity measurements by each sensor at 10 Hz. The true value is obtained in the same way as the one in the training data. Varying f

Table II: Sensor fusion performance for different f. E ($\hat{E}$) is the theoretical (empirical) worst-case number of rounds with global faults. FI is the average size of correct fusion intervals.

PI_w	$f=0$			$f=1$		
	E	$\hat{E}$	FI	E	$\hat{E}$	FI
PI_{10}	6 (60%)	6 (60%)	0.428	3 (30%)	2 (20%)	0.482
PI_{30}	21 (70%)	9 (30%)	0.329	10 (33%)	3 (10%)	0.397
PI_{50}	31 (62%)	9 (18%)	0.325	15 (30%)	3 (6%)	0.391
PI_{100}	57 (57%)	36 (36%)	0.248	28 (28%)	8 (8%)	0.318
PI_{200}	83 (42%)	68 (34%)	0.211	41 (21%)	27 (14%)	0.263

Table III: Average size of filtered fusion interval for different values for f and noise bound M.

PI_w	$f=0$		$f=1$	
	$M=0.005$	$M=0.001$	$M=0.005$	$M=0.001$
PI_{10}	0.504	0.466	0.499	0.466
PI_{30}	0.545	0.400	0.493	0.397
PI_{50}	0.635	0.403	0.540	0.399
PI_{100}	0.815	0.366	0.598	0.358
PI_{200}	1.036	0.371	0.673	0.334

between 0 and 1,[5] we perform sensor fusion at each round of the test data and check whether the fusion interval contains the true value (i.e., there is a global fault). Then we compute the worst number of rounds (denoted by $\hat{E}$) with global faults in a window and compare that with the theoretical bound E computed by Algorithm 1 given the TFM parameters for each sensor. In addition, we calculate the average size of the correct fusion intervals for each setup (denoted by FI).

Table II shows the performance results, where in addition to the absolute values of E and $\hat{E}$, we show their proportion of the window size in a percentage. $\hat{E}$ is never larger than E but is sometimes equal, hence the worst case is indeed observed in reality. At the same time, as the window size increases, the analytical worst-case becomes less tight. Furthermore, as f increases, so does the average size of fusion interval, but the number of worst-case global faults decreases. Regardless of the choice of f, both metrics generally improve with window size. The reason is that the TFM for a bigger window tends to have a smaller error bound (resulting in better precision) as well as a smaller (e/w) ratio (resulting in better accuracy).

[5]The case of $f=2$ is excluded because $n=3$, and, in that case, the fusion interval cannot be bounded in general.

Table IV: False alarm rate

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

In addition, we also computed the filtered fusion interval at each round for the different setups. Since a constant input was used to drive the LandShark, the vehicle's state does not change except for process noise. Since the noise is not known exactly, we used two different bounds to compute the filtered fusion interval. Table III presents the average size of the filtered fusion interval for the two values of f and for noise bounds equal to either 0.005 m/s or 0.001 m/s. For larger values of the noise, the proposed filtering algorithm does not perform very well with large windows due to the increased uncertainty that it introduces. Yet, for the smaller noise bound using larger windows is still more beneficial for the system. Since the filtered fusion interval always contains the true value and its size is not significantly larger than the average size of the fusion interval in a given round, we argue that systems with small noise should utilize the filtered fusion interval as a correct conservative estimate of their state.

7.3. Attack Detection Performance

In this subsection, we evaluate the performance of the attack detectors for the selected TFM parameters employing various attack scenarios explained below. We use the same test data mentioned in the previous subsection.

We first evaluate the **false alarm** rates of the attack detectors; the false alarm rate is calculated as the number of incorrect alarms over the total number of tests. Note that all raised alarms are considered to be incorrect because no attacks are present yet. We perform the first test as soon as w measurements are available; consequently, whenever a new measurement arrives from each sensor, a new test is performed using the last w measurements. Table IV shows the false alarm rates for the TFM parameters of Table I.[6] The results show that for window sizes 200 and 50, the false alarm rate is zero, but it is non-zero for window sizes 10 and 1 (i.e., the SF-based detector). The reason is that the false alarms result from transient faults and they do not appear too often in larger windows. On the other hand, the SF-based approach has a low false

[6] PI_{30} and PI_{100} are excluded for the rest of the paper to avoid clutter.

Table V: 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

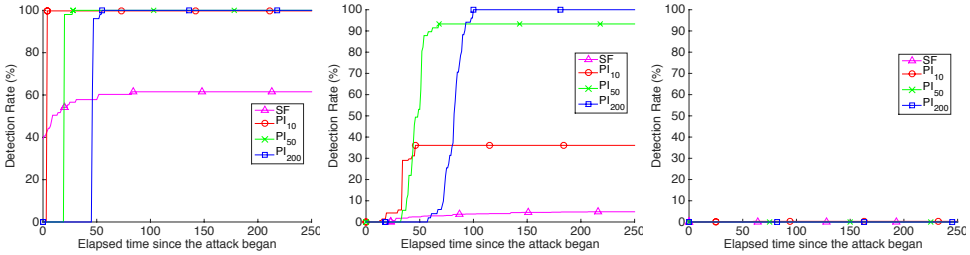
alarm rate because it uses conservative error bounds; it raises some false alarms because the largest faulty measurement observed in the training data was less than the one in the test data.

We now evaluate the **attack detection** rate assuming that only one (unknown to us) out of the three sensors is attacked. We consider three different attack scenarios: (1) bias attack; (2) random attack; (3) greedy attack. The bias attack adds a constant of 0.8 m/s to the attacked sensor. The random attack adds a uniformly distributed random noise between 0 and 0.8 m/s.[7]. The greedy attack replaces the measurement of the attacked sensor with a specially crafted measurement designed to maximize the uncertainty (i.e., the fusion interval size) in the system; this is also a stealthy attack as discussed in [Ivanov et al. 2014a].[8] Note that the attack is present in every round in the detection rate test, thus all raised alarms are true alarms.

To evaluate the attack detection rate, we employ the same test data as above and augment it by simulating each attack scenario described above. Table V summarizes the detection rates for each detector and attack scenario. The detection rate improves in general as the window size increases. The only exception is greedy attack, where most of the detectors raise no alarms. This indicates that given enough knowledge and computational power, the attacked sensor can pretend as if it is a correct one while it negatively affects the system. Note that the SF-based approach's detection rate is lower than the PI-based one's because it uses conservative error bounds.

Note that the false alarm rate improves with window size, whereas, for the same reason, the attack detectors with a large window size may be slow to detect attacks. Therefore, we also evaluate the detection rate vs. the elapsed time since the attack begins. The results for the various TFM parameters are shown in Fig. 5, where the steady-state detection rates correspond to the detection rates in Table V. Fig. 5c shows

[7]The magnitudes of the bias and random attacks are selected to be roughly as large as the interval size of the most imprecise sensor (i.e., GPS).
[8]We assume the greedy attack knows the other abstract measurements, as possible if sensor communication occurs on a shared medium, e.g., CAN bus.



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

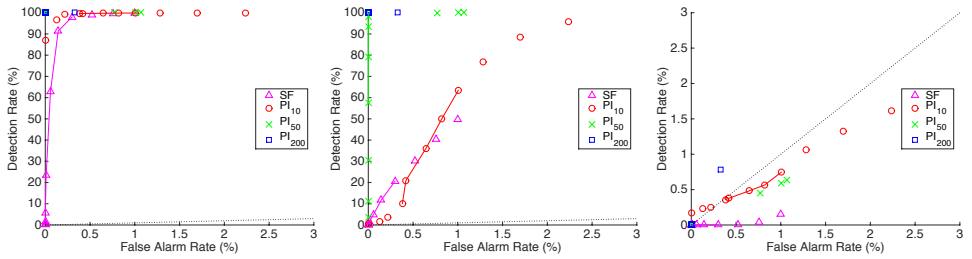
Fig. 5: Time to detection plots under the three classes of attacks.

that all detectors rarely detect any greedy attacks. From the cases of biased and random attacks, Fig. 5 shows that the steady-state detection rate improves with window size, and the time needed to reach the steady-state detection level increases only marginally.

To compare the attack detectors in greater depth and to examine their robustness to the choice of the TFM parameters, we vary the error bounds of the TFM parameters selected in Section 7.1. Specifically, varying ϵ of each sensor from 50% to 150% of their magnitudes, we calculate the false alarm rate and detection rate for each setup. By examining the robustness of attack detector regarding the TFM parameters, we can qualitatively demonstrate the importance of accurate parameter selection. The results for the varied TFM parameters for each window size are depicted as the receiver operator characteristic (ROC) curve in Fig. 6, which is a classical way to measure a detector's performance. Note that the 45° line is a dotted line and is moved lower to make comparative performance clear.[9]

Note that data points which trend towards the upper left corner indicate a better detector because the detector would have a larger detection rate and a smaller false alarm rate [Wald 1973]. We can qualitatively evaluate that one detector is more robust than another if the ROC data points cluster nearer to the upper left corner when varying its parameters [Wald 1973]. Therefore, the robustness of the PI-based detectors improves with window size in general. Note that PI_{10} performs marginally better than the SF-based detector, and PI_{200} and PI_{50} apparently outperform the others. Lastly, the ROC curves for the greedy attack scenario lie on the 45° line, which implies that

[9]Only 13 points are used to show the general trend and avoid overcrowding.



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

Fig. 6: 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. Note the scale is different in the greedy attack case.

when the most powerful attacker is present, the performance of the attack detectors is not better than a coin flip.

The results presented in this section suggest that the false alarm rate, the detection rate and the robustness of PI-based detectors improve with window size, at a cost of a marginal increase of time-to-detection. In addition, the PI-based detector outperforms the SF-based one as the window size increases.

Finally, we only briefly highlight the attack identification performance because it shows the almost identical result to the detection one. Note that in general, the identification rate also improves with window size, experiencing only a marginal increase in time-to-identification.

8. CONCLUSION

In this paper, we considered the security of CPS with redundant sensors, some of which can be attacked while others may be transiently faulty. Employing TFM's, we presented an algorithm to detect and identify sensor attacks in the presence of transient faults. Since reliable TFM parameters may not be given by manufacturers, we provided a method to identify such parameters from training data. We examined the effect of TFM on sensor fusion performance, and provided an algorithm to find the filtered fusion interval which is guaranteed to contain the true value. These approaches were evaluated on real data from a robotic platform. For future work, we plan to enhance the detection method by incorporating a system's dynamical model (currently only used for the filtered fusion interval).

ACKNOWLEDGMENTS

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. This work was also supported in part by NSF CNS-1505701 grant and a grant from Intel.

REFERENCES

Michèle Basseville, Igor V Nikiforov, and others. 1993. *Detection of abrupt changes: theory and application*. Vol. 104. Prentice Hall Englewood Cliffs.

Black-I Robotics. 2015. LandShark UGV. (2015). http://blackirobotics.com/LandShark_UGV_UC0M.html.

R. R. Brooks and S. S. Iyengar. 1996. Robust Distributed Computing and Sensing Algorithm. *Computer* 29, 6 (June 1996), 53–60.

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

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

P. Chew and K. Marzullo. 1991. Masking failures of multidimensional sensors. In *SRDS'91: Proc. 10th Symposium on Reliable Distributed Systems*. 32–41.

Mark HA Davis. 1975. The application of nonlinear filtering to fault detection in linear systems. *Automatic Control, IEEE Transactions on* 20, 2 (1975), 257–259.

Cristobald De Kerchove and Paul Van Dooren. 2010. Iterative filtering in reputation systems. *SIAM J. Matrix Anal. Appl.* 31, 4 (2010), 1812–1834.

Nicolas Falliere, Liam O Murchu, and Eric Chien. 2011. W32. stuxnet dossier. *White paper, Symantec Corp., Security Response* (2011).

Hamza Fawzi, Paulo Tabuada, and Suhas Diggavi. 2011. Secure state-estimation for dynamical systems under active adversaries. In *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*. IEEE, 337–344.

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

Paul M Frank and X Ding. 1997. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of process control* 7, 6 (1997), 403–424.

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

Andy Greenberg. 2015. Hackers Remotely Kill a Jeep on the Highway—With Me in It. (July 2015). http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/.

Inseok Hwang, Sungwan Kim, Youdan Kim, and C.E. Seah. 2010. A Survey of Fault Detection, Isolation, and Reconfiguration Methods. *Control Systems Technology, IEEE Transactions on* 18, 3 (May 2010), 636–653.

Rolf Isermann. 1984. Process fault detection based on modeling and estimation methods—a survey. *Automatica* 20, 4 (1984), 387–404.

R. Ivanov, M. Pajic, and I. Lee. 2014a. Attack-Resilient Sensor Fusion. In *DATE'14: Design, Automation and Test in Europe*.

R. Ivanov, M. Pajic, and I. Lee. 2014b. Resilient Multidimensional Sensor Fusion Using Measurement History. In *HiCoNS'14: High Confidence Networked Systems*.

D. N. Jayasimha. 1994. Fault Tolerance in a Multisensor Environment. In *SRDS'94: Proc. 13th Symposium on Reliable Distributed Systems*. 2–11.

Guofei Jiang, Haifeng Chen, and K. Yoshihira. 2006. Modeling and Tracking of Transaction Flow Dynamics for Fault Detection in Complex Systems. *Dependable and Secure Computing, IEEE Transactions on* 3, 4 (Oct 2006), 312–326.

K.R. Joshi, M.A. Hiltunen, W.H. Sanders, and R.D. Schlichting. 2011. Probabilistic Model-Driven Recovery in Distributed Systems. *Dependable and Secure Computing, IEEE Transactions on* 8, 6 (Nov 2011), 913–928.

R. E. Kalman. 1960. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering* 82, Series D (1960), 35–45.

Man Ho Kim, Suk Lee, and Kyung Chang Lee. 2010. Kalman Predictive Redundancy System for Fault Tolerance of Safety-Critical Systems. *Industrial Informatics, IEEE Transactions on* 6, 1 (Feb 2010), 46–53.

K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage. 2010. Experimental Security Analysis of a Modern Automobile. In *SP'10: IEEE Symposium on Security and Privacy*. 447–462.

Myeong-Hyeon Lee and Yoon-Hwa Choi. 2008. Fault detection of wireless sensor networks. *Computer Communications* 31, 14 (2008), 3469–3475.

Ren C Luo, Chih-Chen Yih, and Kuo Lan Su. 2002. Multisensor fusion and integration: approaches, applications, and future research directions. *Sensors Journal, IEEE* 2, 2 (2002), 107–119.

K. Marzullo. 1990. Tolerating failures of continuous-valued sensors. *ACM Trans. Comput. Syst.* 8, 4 (Nov. 1990), 284–304. DOI:http://dx.doi.org/10.1145/128733.128735

M. Milanese and C. Novara. 2004. Set Membership identification of nonlinear systems. *Automatica* 40, 6 (2004), 957–975.

M. Milanese and C. Novara. 2011. Unified Set Membership theory for identification, prediction and filtering of nonlinear systems. *Automatica* 47, 10 (2011), 2141–2151.

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

Junkil Park, Radoslav Ivanov, James Weimer, Miroslav Pajic, and Insup Lee. 2015. Sensor Attack Detection in the Presence of Transient Faults. In *Cyber-Physical Systems (ICCPS), 2015 ACM/IEEE International Conference on*.

S. Peterson and P. Faramarzi. 2011. Iran hijacked US drone, says Iranian engineer. *Christian Science Monitor, December* 15 (2011).

Mohsen Rezvani, Aleksandar Ignjatovic, Elisa Bertino, and Somesh Jha. 2015. Secure data aggregation technique for wireless sensor networks in the presence of collusion attacks. *Dependable and Secure Computing, IEEE Transactions on* 12, 1 (2015), 98–110.

Aviva Hope Rutkin. 2014. 'Spoofers' Use Fake GPS Signals to Knock a Yacht Off Course. MIT Technology Review. (August 2014).

M. Serafini, P. Bokor, N. Suri, J. Vinter, A. Ademaj, W. Brandstätter, F. Tagliabò, and J. Koch. 2011. Application-Level Diagnostic and Membership Protocols for Generic Time-Triggered Systems. *Dependable and Secure Computing, IEEE Transactions on* 8, 2 (March 2011), 177–193.

M. Serafini, A. Bondavalli, and N. Suri. 2007. On-Line Diagnosis and Recovery: On the Choice and Impact of Tuning Parameters. *Dependable and Secure Computing, IEEE Transactions on* 4, 4 (Oct 2007), 295–312.

D. Shepard, J. Bhatti, and T. Humphreys. 2012. Drone Hack. *GPS World* 23, 8 (2012), 30–33.

Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava. 2013. Non-invasive Spoofing Attacks for Anti-lock Braking Systems. In *Cryptographic Hardware and Embedded Systems - CHES 2013*. Lecture Notes in Computer Science, Vol. 8086. 55–72.

Bruno Sinopoli, Luca Schenato, Massimo Franceschetti, Kameshwar Poolla, Michael I Jordan, and Shankar S Sastry. 2004. Kalman filtering with intermittent observations. *Automatic Control, IEEE Transactions on* 49, 9 (2004), 1453–1464.

A. Teixeira, D. Pérez, H. Sandberg, and K. H. Johansson. 2012. Attack Models and Scenarios for Networked Control Systems. In *Proceedings of the 1st International Conference on High Confidence Networked Systems (HiCoNS '12)*. ACM, New York, NY, USA, 55–64.

Abraham Wald. 1973. *Sequential analysis*. Courier Corporation.

Jon S Warner and Roger G Johnston. 2002. A simple demonstration that the global positioning system (GPS) is vulnerable to spoofing. *Journal of Security Administration* 25, 2 (2002), 19–27.

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

L. Xiao, S. Boyd, and S. Lall. 2005. A scheme for robust distributed sensor fusion based on average consensus. In *IPSN'05*. Article 9, 63–70 pages.

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