# Attack-Resilient Sensor Fusion for Safety-Critical Cyber-Physical Systems Technical Report

Radoslav Ivanov Miroslav Pajic Insup Lee

October 1, 2014

**Abstract**

This paper focuses on the design of safe and attack-resilient Cyber Physical Systems (CPS) equipped with multiple sensors measuring the same physical variable. A malicious attacker may be able to disrupt system performance through compromising a subset of these sensors. Consequently, we develop a precise and resilient sensor fusion algorithm that combines the data received from all sensors by taking into account their specified precisions. In particular, we note that in the presence of a shared bus, in which messages are broadcast to all nodes in the network, the attacker's impact depends on what sensors he has seen before sending the corrupted measurements. Therefore, we explore the effects of communication schedules on the performance of sensor fusion and provide theoretical and experimental results advocating for the use of the *Ascending* schedule, which orders sensor transmissions according to their precision starting from the most precise. In addition, to improve the accuracy of the sensor fusion algorithm, we consider the dynamics of the system in order to incorporate past measurements at the current time. Possible ways of mapping sensor measurement history are investigated in the paper and are compared in terms of the confidence in the final output of the sensor fusion. We show that the precision of the algorithm using history is never worse than the no-history one, while the benefits may be significant. Furthermore, we utilize the complementary properties of the two methods and show that their combination results in a more precise and resilient algorithm. Finally, we validate our approach in simulation and experiments on a real unmanned ground robot.

## Contents

# 1 Introduction

Ensuring the safety of Cyber Physical Systems (CPS) is a challenging problem. Depending on the attacker's goals and resources, the consequences of malicious attacks may range from minor variation in performance to absolute inability to control the system [1, 2]. In addition to the multitude of cyber attacks (e.g., denial of service) developed over the years, the fact that CPS rely on real-time information to interact with the physical world makes them additionally vulnerable to physical attacks (e.g., sensor spoofing). Recent attacks on GPS [3, 4] and anti-braking systems [5] have illustrated that by tampering with values obtained from system sensors, the attacker can seriously compromise the safety of the system.

On the other hand, due to proliferation of sensing technology, modern CPS have many sensors that can be used to estimate the same physical variable. For example, modern automotive systems have multiple ways of estimating speed; combining their sensor data to provide more accurate estimates to the controller can have a significant impact on the system's performance and reliability. Even though these sensors' precisions may be different, their measurements can be fused to produce an estimate that is better than any single sensor's [6]. In

addition, having diverse sensors with different accuracy and reliability decreases the system's dependence on a particular sensor.

Increased sensor diversity, however, raises the question of the vulnerability of sensor fusion to malicious attacks. Consequently, in this work, we investigate the design of an attack-resilient sensor fusion algorithm in order to improve the safety and resiliency of CPS under attack. We consider a standard CPS architecture from Figure 1, where multiple sensors communicate with the controller via a shared (broadcast) bus (e.g., CAN bus in automotive CPS). We assume that communication is implemented in a time-triggered manner – in every frame, each sensor transmits its measurements during its allocated time slot, according to a predefined schedule.

The first consideration when designing a sensor fusion component of CPS is the underlying sensor model. These can be broadly divided into two main categories: *probabilistic* and *abstract*. In the former, the sensor provides the controller with a single measurement that may be corrupted by noise with a known probability distribution (e.g., [7]). In the latter, the sensor produces a set with all possible values for the true state of the variable in question [8]. Each model is a basis for a different kind of analysis – while the probabilistic models allow designers to consider the average case and to ignore events with low probability, the abstract model is usually utilized for worst-case analysis.

Our goal is to guarantee the safety of CPS under attack; hence we focus on worst-case analysis. Accordingly, we adopt the *abstract* sensor model, in which each sensor provides an interval of possible values. The width of the interval reflects its precision - a larger interval implies less confidence in the obtained measurement. Intuitively, the abstract model is well-suited for worst-case analysis for the following reason. Suppose interval $A$ is an unsafe region for a system (e.g., $A = [100\ mph, \infty)$). Then if every sensor's interval has an empty intersection with $A$ one can conclude with certainty that the system is not in state $A$. It is worth noting that this presents a very general sensor model as it does not make any assumptions about the distribution of the sensor measurements or their noise. Instead, the interval is constructed based on manufacturer specifications about precision and accuracy of the sensor, as well as implementation limitations such as sampling jitter and synchronization errors [9].

We assume that an attacker is able to take control of some of the sensors and send any measurements to the controller on their behalf. The attacker's goal is to use these compromised sensor measurements to affect the performance of sensor fusion by forcing it to produce an incorrect output or increasing the uncertainty of the produced value. In particular, if the output is an interval, the attacker would try to maximize its size since a larger interval reduces the confidence in the provided measurements and may indicate that the system is in an unsafe state. Since safety analysis is concerned with the worst case of a system's operation, we assume the attacker has full knowledge of its model; in particular, he is aware of the fusion algorithm used by the system as well as of the sensor and system specifications. In addition, he has access to the shared bus and hence to all messages that are broadcast on it.

The contribution of this work is the design and analysis of a safe and attack-

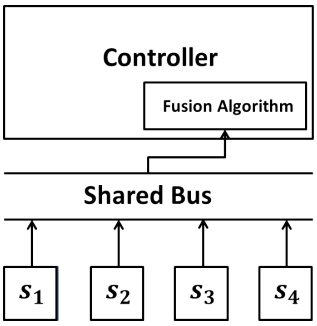



Figure 1: A typical architecture of a CPS, with sensors communicating over a shared bus with a controller. After obtaining sensor measurements, the controller performs a sensor fusion algorithm.

resilient sensor fusion for a system such as the one in Figure 1. We provide a framework for investigating and securing such systems based on their sensors' specifications and dynamics. Specifically, given the sensor model used in this work, our approach is based on the fusion algorithm developed in [8]. This algorithm produces a fusion interval for a bounded number of faulty sensors and is guaranteed to contain the true value (see Section 2 for more detail). In this paper, we propose an improvement to the sensor fusion algorithm as well as a specific communication schedule that aims to minimize the attacker's impact on safety and performance. In addition, we combine the two approaches in order to leverage their complementary properties.

To improve the precision of the original sensor fusion algorithm, we exploit knowledge of system dynamics and incorporate past measurements in the sensor fusion algorithm. To achieve this, we focus on discrete-time linear systems with bounded noise. This paper identifies and compares all possible ways of mapping past measurements to the current time and compare them in terms of the size of the fusion interval that they produce. We also show that the algorithm using history never leads to a larger interval than the no-history one.

Furthermore, to enhance the resiliency of sensor fusion, we note that in shared buses measurements are broadcast to all nodes in the network, including the attacked ones. Consequently, the attacker's capabilities depend on what measurements he has seen before sending his own. Note that, given the chosen fusion algorithm, the attacker's goal is to increase the size of the fusion interval if he cannot produce a wrong interval. In particular, if he sends his intervals last, he can maximize the size of the fusion interval based on the placements of the correct intervals. We show that in the worst case, the attacker does not benefit from compromising the least precise sensors but may achieve the worst case for the system if he takes control of the most precise. Consequently, we argue that system designers should prioritize the protection of the most precise sensors in their systems. In addition, based on these observations, we consider different

communication schedules (based on sensors' precisions), and investigate how they affect attacker's impact on the performance of sensor fusion (i.e., size of the fusion interval). We show that systems adopting the abstract sensor model should also implement the *Ascending* schedule, which orders sensors according to their interval size starting from the most precise.

Finally we validate our approach on an unmanned ground vehicle case study. We use the LandShark robot [10] and illustrate in simulations and experiments the advantages of the Ascending schedule as well as of the use of measurement history for sensor fusion.

This paper is organized as follows. Section 2 introduces formal definitions of the problems addressed in this work. In Section 3, we formalize a model of the attacker (his goals and constraints), and presents worst-case results with respect to the size of the fusion interval. Section 4 compares effects of different communication schedules on the attacker's performance. Section 5 introduces system dynamics and the benefits of the use of measurement history, whereas Section 6 shows the combined effect of the two methods. Finally, in Section 7 we illustrate performance of the proposed sensor fusion approach using simulations and experiments on an autonomous vehicle, before discussing related work (Section 8) and providing some concluding remarks (Section 9).

# 2 Problem Formulation and Preliminaries

This section describes the problems addressed in this work. At a high level, the goal is to use sensor redundancy to improve the system's resiliency to attacks. To this end, we analyze the fusion algorithm and shared bus modules (as shown in Figure 1). We formalize both the system and attack models used in the paper, before stating the two problems considered in the work.

## 2.1 System Model

The system consists of $n$ sensors measuring the same physical variable, e.g., velocity. As mentioned above, we assume *abstract* sensors; therefore, each sensor provides the controller with an interval containing all possible values of the true state. The interval is computed based on the sensor's specification and manufacturer guarantees. Thus, its size reflects the system's confidence in the sensor's precision, i.e., a larger interval means a less precise sensor. A sensor is said to be *correct* if its interval contains the true value and *compromised/corrupted* otherwise.

In addition, since most CPS have known dynamics, we assume the system operates according to discrete-time linear dynamics of the form[1]

$$x(t+1) = ax(t) + w,$$

[1]We have addressed the problem of attack-resilient sensor fusion for multidimensional systems in our previous work [11].

where $x \in \mathbb{R}$ is the system's state, $a \in \mathbb{R}$ is the transition matrix and $w \in \mathbb{R}$ is bounded process noise such that $|w| \leq M$ for some constant $M$. All sensors transmit at each point in time.

In addition, we assume that in each round sensors transmit their measurements in a predefined schedule, i.e., each sensor only transmits its interval in an allocated slot. Sensors communicate over a shared bus (e.g., CAN bus) such that all messages are broadcast to all nodes in the network. Therefore, the sensor scheduled to transmit last is able to receive and examine all other measurements before sending.

Once the controller receives all measurements in a given round, it performs the following abstract sensor fusion algorithm as developed by Marzullo [8]. As discussed in Section 1, the algorithm is chosen because it is conservative and fits the safety analysis used in this work.

## 2.2 Fusion Algorithm

The inputs to the algorithm are $n$ intervals and a number, $f$, which denotes an upper bound on the number of corrupted sensors in the system (since this number is unknown, $f$ is usually set conservatively high, e.g., $f = \lceil n/2 \rceil - 1$). The algorithm outputs an interval, referred to as the fusion interval in this work, which spans the smallest to largest point contained in at least $n - f$ intervals. Intuitively, the algorithm is conservative: since there are at most $f$ corrupted sensors, there are at least $n - f$ correct ones, hence the true value can lie in any group of $n - f$ intervals. Thus the algorithm outputs the smallest interval containing all such groups.

The algorithm is illustrated in Figure 2. As can be seen in the figure, when $f = 0$ (i.e., the system is confident that all intervals are correct) the fusion interval is just their intersection. When $f = 1$ the fusion interval contains all points that lie in at least four intervals. As the figure shows, as $f$ increases, so does the size of the fusion interval; in particular, if $f = n - 1$ the fusion interval would just be the convex hull of the union of all intervals. Three results about the size of the fusion interval from the original work by Marzullo are relevant to this paper. First of all, if $f < \lceil n/3 \rceil$, the size of the fusion interval is bounded by the size of some correct interval. If $\lceil n/3 \rceil \leq f < \lceil n/2 \rceil$, the fusion interval can be at most as large as some sensor's interval, not necessarily correct. Finally, if $f \geq \lceil n/2 \rceil$ the fusion interval can be arbitrarily large.

Note that, while the fusion interval can be used for performing closed-loop control (e.g., by selecting its middle point as its "measurement"), in this work it is used for safety analysis. In particular, if the fusion interval contains any points that affect the system's safety, a flag is raised that the system is in a potentially dangerous state.

## 2.3 Attack Model

In this work, we assume that compromised sensors are not randomly faulty but are controlled by a malicious attacker. As discussed above, sensors in the sys-

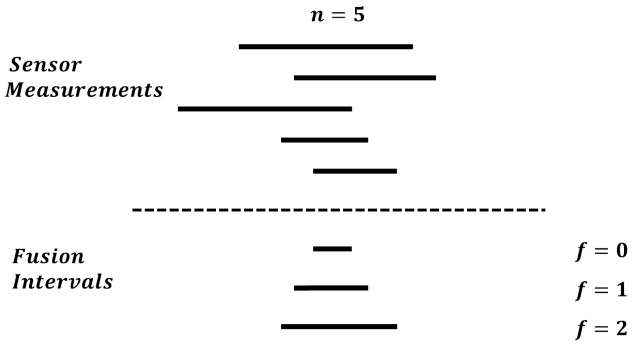



Figure 2: The fusion interval for three values of $f$, for a system with $n = 5$ sensors. Dashed horizontal line separates sensor intervals from fusion intervals in all figures in this work.

tem communicate over a shared bus. Thus, by gaining control over a sensor, an attacker has the ability to inspect all sensor measurements transmitted before his sensor's slot in the current round's schedule as well as all past rounds' measurements. We assume the attacker can send any interval[2] on behalf of the corrupted sensor. In addition, we consider a worst-case scenario where the attacker has unlimited computational power and full system knowledge, including sensor/design specifications and the employed sensor fusion algorithm. The attacker's goal is to disrupt system performance by leading the system to believe it is in an unsafe state. As described above, the strategy used to accomplish this goal (formalized in Section 3) is through maximizing the size of the fusion interval. In addition, the attacker has the constraint that he has to stay undetected throughout the system's operation; while a single pronounced attack (followed by detection) may be considered as a fault and ignored, a consistent uncertainty may be worse for the system. Based on the worst-case result described above, we assume that the number of compromised sensors, denoted by $f_a$, is always less than $\lceil n/2 \rceil$, and we assume $f$ is set conservatively high so that $f \geq f_a$ (for example, this can be guaranteed by setting $f = \lceil n/2 \rceil - 1$).

## 2.4 Problems

Given the above model, we note that the attacker's impact depends on the position of his sensors in the transmission schedule. In particular, if his sensors are last in the schedule, the attacker can examine all other measurements before sending his intervals. This would allow him to place his interval(s) in the way that maximizes damage while not being detected. Therefore, the first problem considered in this paper is the following.

[2]Note that sensors have predefined and known widths of measurement intervals, so the attacker cannot change the width of his sensor's interval if he wants to avoid detection.

**Problem 1.** *How does the sensor communication schedule affect the attacker's impact on the performance of sensor fusion (as measured by the size of the fusion interval) in a given round. Find the schedule that minimizes this impact?*

In the second part of the paper we aim to improve the precision of sensor fusion as that would mitigate the attacker's impact and eliminate certain safety concerns. To this end, we explore the use of system dynamics in the sensor fusion algorithm. Thus, the second problem addressed in this paper is follows:

**Problem 2.** *How can we utilize the knowledge of system dynamics and past measurements to improve the precision of the sensor fusion algorithm for any attack strategy and communication schedule?*

Finally, we analyze the mixture of the solutions of the aforementioned two problems in order to combine the power of the two methods.

### 2.5 Notation

Let $\mathcal{N}(t)$ denote all $n$ intervals measured by sensors at time $t$. In Sections 3 and 4 we omit time notation and write $\mathcal{N}$ since no time is used in the analysis. Let $S_{\mathcal{N}(t),f}$ denote the fusion interval given the sensors in $\mathcal{N}(t)$ and an upper bound $f$. For a given interval $s$, let $l_s$ and $u_s$ be the lower and upper bound of $s$, respectively. By $|s|$ we denote the size of $s$, i.e., $|s| = u_s - l_s$; in particular, $|S_{\mathcal{N}(t),f}|$ is the size of the fusion interval. Finally, let $\mathcal{C}(t)$ denote the (unknown to the system) set of correct sensors at time $t$.

## 3 Attack Strategy and Worst-Case Analysis

This section formalizes the attack strategy considered in this work and illustrates how the attacker's capabilities vary with the utilized transmission schedule. Given this strategy, the second part of the section provides worst-case results to suggest which sensors would be most beneficial for the attacker to corrupt and for the system to defend, respectively. We denote the strategy with $AS_1$; to illustrate its effectiveness from the attacker's point of view, we compare it with another viable strategy in Section 4. Note that this section does not consider the use of previous sensor readings, hence a single round is analyzed in isolation. We introduce the use of measurement history in Section 5.

### 3.1 Attack Strategy

As described in Section 2, the attacker has a goal, maximize the size of the fusion interval, and constraints, stay undetected. This subsection formalizes the two, beginning with the latter.

### 3.1.1 Constraints: Staying Undetected

Formally, the attacker has two modes: *passive* and *active*, as defined below. When in passive mode, the attacker's constraints are tighter, and thus his impact is limited. In active mode, on the other hand, the constraints on the placement of the compromised intervals are looser, hence the attacker can send intervals that would greatly increase the uncertainty in the system.

The attacker begins in passive mode, in which the main goal is to stay undetected. The detection mechanism used in this work is to check whether each interval has a nonempty intersection with the fusion interval;[3]since the fusion interval is guaranteed to contain the true value, any interval that does not intersect the fusion interval must be compromised. Thus, in passive mode, the attacker computes the intersection of all seen measurements, including his own sensors', which is the smallest interval from the attacker's perspective that is guaranteed to contain the true value. We denote this intersection by $\Delta$. Therefore, in passive mode the attacker must include $\Delta$ in his interval (any point that is not contained may be the true value) and has no restrictions on how to place the interval around $\Delta$ (if the interval is larger than $\Delta$[4]).

The attacker may switch to active mode when at least $n - f - f_{ar}$ measurements have been transmitted, where $f_{ar}$ is the number of unsent compromised intervals. At this point, the attacker may send an interval that does not contain $\Delta$ because he is aware of enough sent measurements, i.e., he can prevent his sensor from being detected because he has exactly $f_{ar}$ remaining intervals to send and can guarantee each interval overlaps with $n - f - 1$ sensors and with the fusion interval, consequently. When in active mode, the attacker is not constrained when sending his intervals as long as overlap with the fusion interval is guaranteed.

### 3.1.2 Goal: Maximizing the size of the fusion interval

When maximizing the size of the fusion interval, the attacker's strategy consists of two different cases depending on the position of the attacker's intervals in the transmission schedule: one to target the largest interval and another to target the largest expected interval.

Specifically, if all the attacker's sensors are scheduled to transmit last, meaning that the attacker will be aware of all measurements prior to sending his, his strategy can be stated through the following optimization problem, where variables $a_1, \ldots, a_{f_a}$ represent the attacked intervals:

$$\begin{aligned} &\max_{a_1,\ldots,a_{fa}} \quad |S_{\mathcal{N},f}| \\ &\text{s.t. } S_{N,f} \cap a_i \neq \emptyset, \quad i = 1, \ldots, f_a. \end{aligned} \tag{1}$$

[3]In Section 5, we use historical measurements to further improve the system's detection capabilities.

[4]Note that it cannot be smaller than $\Delta$ since $\Delta$ includes the intersection of all measurements of the corrupted sensors.

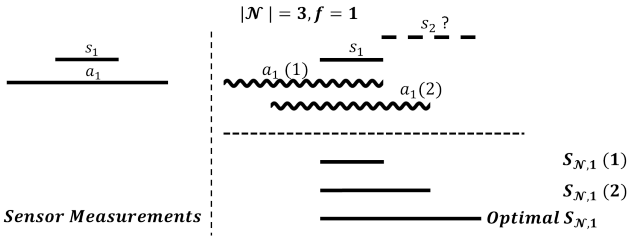



Figure 3: An example showing that if attacker (sinusoid) has not seen all intervals then he has no strategy that guarantees fusion interval is maximized.

Since the solution to this problem can be obtained with **full information** about the correct sensors' measurements, we call this solution and the strategy that led to it, respectively, optimal.

**Definition 1.** *The attack strategy obtained as a solution to the optimization problem* (1) *(i.e., the placements of the attacked intervals that achieve the solution) is called* optimal *(from the attacker's point of view) given the correct sensors' measurements. Any attack strategy that achieves this solution is also referred to as* optimal.

Note that the attack strategy described by optimization problem (1) is optimal by definition. However, there are scenarios in which there exists no optimal strategy for the attacker if his sensors are not last in the schedule. For example, consider the scenario depicted in Figure 3, where out of three sensors, $a_1$ is under attack. Suppose that the attacker transmits second in the schedule so that he is aware of $s_1$'s and his own sensor's measurement but not of $s_2$'s. Given the measurements shown in the figure, the attacker cannot guarantee that the fusion interval will be maximized regardless of the interval that he sends. In particular, if $a_1$ is sent to the left of $s_1$ ($a_1(1)$ in the figure) then $s_2$'s measurement could appear as shown, in which case $a_1(2)$ would have resulted in a larger fusion interval. Other attacks could be similarly shown to not be optimal for any measurement that can be obtained from $s_2$.

Consequently, in cases such as the one in Figure 3, a reasonable strategy for the attacker is to maximize the expected size of the fusion interval. The expectation is computed over all possible placements of the **unseen** correct and compromised intervals.[5] Formally, for each compromised interval $a_k$ the attack strategy can be described with the following optimization problem

$$\begin{aligned} &\max_{a_k,\ldots,a_{fa}} \quad \mathop{\mathbb{E}}_{\mathcal{C}_k^R} |S_{\mathcal{N},f}| \\ &\text{s.t. } S_{N,f} \cap a_i \neq \emptyset \quad i = k, \ldots, f_a, \end{aligned} \tag{2}$$

[5]To compute the expectation, the attacker is implicitly assuming intervals are uniformly distributed around $\Delta$. If additional information is available about the distribution of sensor measurements, it can be incorporated in the optimization problem (2).

where $\mathcal{C}_k^R$ is the set of all possible placements of the correct intervals that will be transmitted after $a_k$, and $\mathbb{E}$ is the expectation operator.

As shown in Figure 3, there are scenarios in which no optimal strategy exists; yet, there do exist cases in which there is an optimal solution even if the attacker is not last in the schedule (and the strategy obtained as a solution to the optimization problem (2) leads to that solution). In particular, there exist scenarios in which if the unseen intervals are small enough it is possible for the attacker to obtain an optimal strategy.

To formalize this statement, we introduce the following notation. Let $\mathcal{C}^S$ be the set of seen correct intervals and let $\mathcal{C}^R$ be the set of correct sensors that have not transmitted yet. Let $l_{n-f-f_a}$ be the $(n-f-f_a)^{th}$ smallest seen lower bound and let $u_{n-f-f_a}$ be the $(n-f-f_a)^{th}$ largest seen upper bound. Finally, let $a_{min}$ be the attacked sensor with smallest width.

**Theorem 1.** *Suppose $n-f-f_a \leq |\mathcal{C}^S| < n-f_a$. There exists an optimal attack strategy if one of the following is true:*

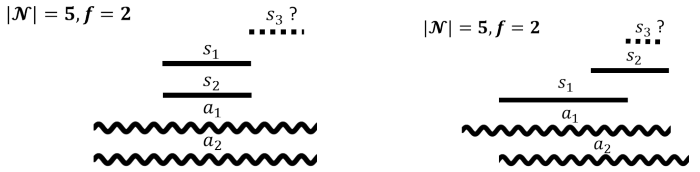
*(a)* $\forall s_i, s_j \in \mathcal{C}^S, l_{s_i} = l_{s_j}, u_{s_i} = u_{s_j}$ *and* $\forall s \in \mathcal{C}^R$, $|s| \leq (|a_{min}| - |S_{\mathcal{C}^S \cup \Delta,0}|)/2$

*(b)* $|a_{min}| \geq u_{n-f-f_a} - l_{n-f-f_a}$ *and* $\forall s \in \mathcal{C}^R$,
$|s| \leq \min\{l_{S_{\mathcal{C}^S \cup \Delta,0}} - l_{n-f-f_a}, u_{n-f-f_a} - u_{S_{\mathcal{C}^S \cup \Delta,0}}\}$

**Remark.** *Note that the conditions in the theorem state that either all seen correct intervals coincide with one another, and the attacker can attack around them (a); or that the unseen correct intervals are small enough so that they cannot change the extreme points contained in at least $n-f-f_a$ seen correct intervals (b), in which case the attacker can attack around these points.*

*Proof.* First suppose the first statement is true. We argue that the optimal strategy for the attacker is to attack on both sides of seen intervals. For any $s \in \mathcal{C}^R$, $s$ must overlap with at least one point in $S_{\mathcal{C}^S \cup \Delta,0}$ (the overlap must contain the true value) and since $|s| \leq (|a_{min}| - |S_{\mathcal{C}^S \cup \Delta,0}|)/2$ then $s$ will necessarily overlap with all malicious sensors implementing the above strategy. Note that since $f < \lceil n/2 \rceil$, the fusion interval cannot be larger than the union of all correct intervals. Therefore, this strategy is optimal because the attacker can guarantee that all her intervals contain all correct intervals. Figure 4a illustrates this case. All seen correct intervals coincide, and the attacker's intervals are large enough to guarantee that attacking on both sides will make sure all unseen intervals are included.

Now suppose the second case is true. Then the attacked intervals are large enough to contain both $l_{n-f-f_a}$ and $u_{n-f-f_a}$, thus making sure the fusion interval is $[l_{n-f-f_a}, u_{n-f-f_a}]$. This attack is optimal since the unseen intervals are all small enough to not change the positions of points $u_{n-f-f_a}$ and $l_{n-f-f_a}$. Figure 4b presents an example of this case. The unseen interval, $s_3$, cannot change the largest and smallest points contained in at least one correct interval. □




(a) Attacker has seen $s_1$ and $s_2$, while the unseen $s_3$ is small enough.

(b) Attacker has seen $s_1$ and $s_2$, while the unseen $s_3$ is small enough.

Figure 4: Examples of the two cases of Theorem 1. Attacked intervals are indicated by sinusoids.

## 3.2 Worst-Case Analysis

Given the attack strategy described in the previous subsection, we now analyze worst-case results based on the sizes of the attacked and correct sensors. The first result puts the problem in perspective - it provides an absolute upper bound on the size of the fusion interval.

**Theorem 2.** *Let $s_{c_1}$ and $s_{c_2}$ be the two largest-width correct sensors. Then $|S_{\mathcal{N},f}| \leq |s_{c_1}| + |s_{c_2}|$.*

*Proof.* Let $s_l$ and $s_u$ be the two correct intervals with smallest lower bound and largest upper bound, respectively. Since $f < \lceil n/2 \rceil$, the lower bound of $S_{\mathcal{N},f}$ cannot be smaller than the lower bound of $s_l$ and its upper bound cannot be larger than the upper bound of $s_u$. Thus, the width of $S_{\mathcal{N},f}$ is bounded by the sum of the widths of $s_l$ and $s_u$ because any two correct intervals must intersect. Hence, the width of $S_{\mathcal{N},f}$ is bounded by the sum of the two largest correct intervals. □

Theorem 2 provides a conservative upper bound on the size of the fusion interval because it does not directly take into account the sizes of the attacked intervals. The following results analyze how the worst case varies with different attacked intervals.

To formulate the theorems, we use the following notation. Let $\mathcal{L}$ be the set of predefined lengths of all intervals. We use $S_{na}$ to denote the worst-case (largest width) fusion interval when no sensor is attacked. Similarly, let $S_{\mathcal{F}}$ be the worst-case fusion interval for a fixed set of attacked sensors $\mathcal{F}$, $|\mathcal{F}| = f_a$, whereas $S^{wc}_{f_a}$ is the worst-case fusion interval for a given number of attacked sensors, $f_a$. Finally, we refer to the set of $n$ fixed (i.e., specific) measurement intervals as a "configuration". Note that $|S_{na}| \leq |S_{\mathcal{F}}| \leq |S^{wc}_{f_a}|$ by definition. The first inequality is true since when there are no attacks, all intervals must contain the true value, which is not the case in the presence of attacks, hence the worst-case is at least the same. The second inequality is true since the worst-case with $f_a$ attacks may not be achieved for any $\mathcal{F}$ with $|\mathcal{F}| = f_a$.

**Theorem 3.** *If the $f_a$ largest intervals are under attack, then $|S_{na}| = |S_{\mathcal{F}}|$.*

*Proof.* Note that $|S_{\mathcal{F}}| < |S_{na}|$ is impossible since the attacker can send the correct measurements from her sensors. Thus, suppose $|S_{\mathcal{F}}| > |S_{na}|$. Let $S_{\mathcal{C},0}$ be the intersection of the correct intervals in the configuration that achieves $S_{\mathcal{F}}$. Suppose $S_{\mathcal{F}}$ extends $S_{\mathcal{C},0}$ on the right (note that the argument for the left side is symmetric) by some distance $d$ and let $A$ be the rightmost point contained in $S_{\mathcal{F}}$. Since $f < \lceil n/2 \rceil$, $A$ must lie in at least one correct interval $s_c$. Since $s_c$ is correct it must contain $S_{\mathcal{C},0}$, which implies $d + |S_{\mathcal{C},0}| \leq |s_c| \leq |s_{max}|$, where $s_{max}$ is the largest correct interval. Let $s$ be any attacked interval that contains $A$. Because $|s| \geq |s_{max}|$, $s$ can be placed to contain both $A$ and $S_{\mathcal{C},0}$. Since this can be done for all attacked intervals containing $A$, the same worst-case fusion interval can be achieved if no intervals were attacked. □

The theorem is illustrated in Figure 5a. The attacked intervals $a_1$ and $a_2$ both do not contain the true value, which is at the intersection of the other sensors. Since $a_1$ and $a_2$ are the largest intervals, they can be moved and can be made correct while preserving the size of the fusion interval. Hence, the same worst case can be achieved with correct intervals.

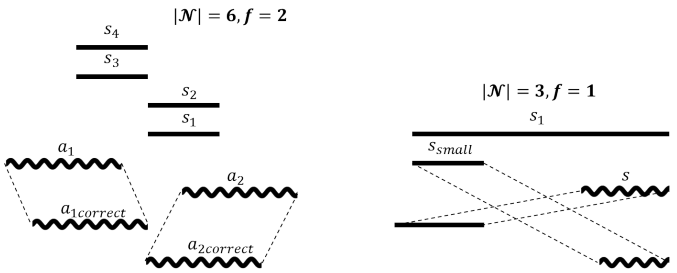
**Theorem 4.** *$|S^{wc}_{f_a}|$ is achievable if the $f_a$ smallest intervals are under attack.*

*Proof.* Note that if $|S^{wc}_{f_a}| = |S_{na}|$, the theorem follows trivially. Consider the case $|S^{wc}_{f_a}| > |S_{na}|$. Suppose $|S^{wc}_{f_a}|$ is not achievable if the $f_a$ smallest intervals are attacked. Let $\mathcal{S}$ be the configuration with $f_a$ corrupted intervals that achieves $|S^{wc}_{f_a}|$ and let $A$ be the rightmost point in $S^{wc}_{f_a}$. Since $|S^{wc}_{f_a}| > |S_{na}|$ there exists an interval $s \in \mathcal{S}$ that does not contain the true value but contains $A$. Let $\mathcal{N}_{small}$ be the set of $f_a$ smallest intervals. If $s \in \mathcal{N}_{small}$ for all such $s$ then $S^{wc}_{f_a}$ is achievable if $\mathcal{N}_{small}$ is under attack and the theorem follows.

Now suppose there exists an $s$ as above such that $s \notin \mathcal{N}_{small}$. Then there exists an interval $s_{small} \in \mathcal{N}_{small}$ that is not under attack. If we swap $s$ and $s_{small}$ such that $s_{small}$ now contains $A$ and $s$ contains the old interval $s_{small}$, $s$ is made correct and $s_{small}$ corrupted while preserving the size of the fusion interval. Since we can do the same for all such $s$, $|S^{wc}_{f_a}|$ can be achieved if $\mathcal{N}_{small}$ is under attack. □

Figure. 5b illustrates the theorem. The worst-case for the setup can be achieved when either $s$ or $s_{small}$ is attacked.

A few conclusions can be drawn from the results shown in this subsection. First of all, from Theorem 2, the smaller the correct intervals are, the smaller the fusion interval will be in the worst case, regardless of the attacker's actions. In addition, as shown in Theorems 3 and 4, the attacker benefits more from compromising precise sensors as opposed to less precise ones. Intuitively, this is true because imprecise sensors produce large intervals even when correct; attacking precise sensors, however, and moving their intervals on one side of large correct intervals, with the true value on the other, may significantly increase the uncertainty in the system. Therefore, one may conclude that it is better for system designers to prioritize the protection of the most precise sensors.




(a) Attacking the biggest intervals does not change the worst case in the system.

(b) Attacking the smallest intervals can achieve the absolute worst case.

Figure 5: Illustrations of Theorems 3 and 4.

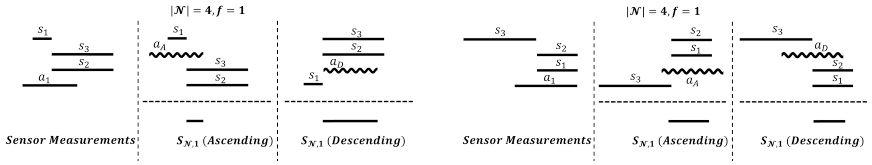
# 4 Schedule Comparison and Analysis

In this section, we analyze the schedule design for communication over the shared bus in Figure 1. It builds on the analysis in Section 3 by considering how different schedules affect the capabilities of the attacker. In particular, we examine the effect of each schedule on the size of the fusion interval.

We first note that the only information available a priori to system designers is the sensors' accuracy and their intervals' sizes, consequently; additional information considerations are discussed in Section 9. Thus, any investigated schedule must be based on interval lengths alone. We focus on the two schedules, named *Ascending* and *Descending*, which schedule sensor transmissions in order starting from the most and least precise, respectively. Other schedules are discussed in Section 9.

We first note that neither schedule is better than the other in all scenarios. Figure 6 shows two examples in which different schedules are better, i.e., they produce smaller fusion intervals. In Figure 6a the fusion interval obtained with the Descending schedule is larger because the attacker is aware of the position of the largest interval. Figure 6b, however, shows that knowing the largest interval does not necessarily bring the attacker any useful information because he can only increase the fusion interval by overlapping with $s_1$ and $s_2$. Hence, if he is aware of $s_3$ when sending his interval he would send $a_D$ but that would be worse for the attacker than sending $a_A$ which would be the case if the attacker had seen $s_1$ and $s_2$ instead.

Since the two schedules cannot be compared in the absolute sense, we consider the average case over all possible sensor measurements. In particular, we investigate the expected size of the fusion interval for a fixed set of sensors with fixed precisions. One may consider all possible measurements of these sensors and all possible attack combinations (with $f_a < \lceil n/2 \rceil$), and compute the average length of the fusion interval over all combinations. Note that there are two main considerations when computing this expectation: (1) what is the distribution of sensor measurements around the true value (e.g., uniform over the




(a) An example where the *Ascending* schedule is better for the system.

(b) An example where the *Descending* schedule is better for the system.

Figure 6: Two examples that show that neither the Ascending nor the Descending schedule is better for the system in all situations. The first column shows the measurements by the sensors, including the attacked one. The other columns contain the intervals sent to the controller, and the corresponding fusion interval.

interval? normal?) and (2) what is the likelihood of different sensors being attacked.

In the following analysis we investigate two possible distributions, uniform and normal,[6] and assume that all sensors are equally likely to be compromised. Since obtaining closed form formulas for the expected sizes of the fusion intervals under the two schedules was not possible, we computed the values for specific systems. In particular, we varied the number of sensors from 3 to 5, the sensor lengths from 5 to 20 with increments of 3, and the number of attack sensors from 1 to $\lceil n/2 \rceil$. For each setup, we generated all possible measurement configurations[7] and for each computed the size of the fusion interval under the two schedules; finally, we computed their weighted sum (depending on the distribution and likelihood of obtaining each configuration), i.e., our best estimate of the real expected size of the fusion interval for a given schedule and system. For all setups, we used $f = \lceil n/2 \rceil - 1$ as input to the sensors fusion algorithm.

Table 1 presents the obtained results. Due to the very large number of setups tried, only a small subset is listed in this work. During simulations, it was noticed that the schedules produce similar-size expected intervals when the interval lengths are close to one another. The schedules differed greatly, however, in systems with a few very precise sensors and few imprecise sensors. Hence, setups in Table 1 were chosen such that they represent classes of combinations according to these observations. As the table shows, **for all analyzed systems**, the expected fusion interval under the Ascending schedule was never larger than that under Descending. In addition, the gains were significant in some cases. This is also true of all other setups that are not shown in this paper. We note that while these results are not sufficient to conclude that the Ascending schedule produces a smaller fusion interval for any sensor configuration, the same framework can be used for any particular system to compare impacts of com-

[6] To approximate a normal distribution, we assumed the length of the interval is equal to six standard deviations, i.e., about 99% of the values of a normal distribution.

[7] We discretized the real line with sufficient precision in order to enumerate the possible measurements.

Table 1: Comparison of the two sensor communication schedules.

| | $\mathbb{E}_U \lvert S_{\mathcal{N},f} \rvert$ *Asc.* | $\mathbb{E}_U \lvert S_{\mathcal{N},f} \rvert$ *Desc.* | $\mathbb{E}_N \lvert S_{\mathcal{N},f} \rvert$ *Asc.* | $\mathbb{E}_N \lvert S_{\mathcal{N},f} \rvert$ *Desc.* |
|---|---|---|---|---|
| $n = 3, f_a = 1,$ $\mathcal{L} = \{5, 11, 17\}$ | 10.77 | 13.58 | 10.87 | 13.18 |
| $n = 3, f_a = 1,$ $\mathcal{L} = \{5, 11, 11\}$ | 9.43 | 10.16 | 9.89 | 10.39 |
| $n = 4, f_a = 1,$ $\mathcal{L} = \{5, 8, 17, 20\}$ | 7.66 | 9.44 | 8.07 | 10.17 |
| $n = 4, f_a = 1,$ $\mathcal{L} = \{5, 8, 8, 11\}$ | 6.32 | 6.53 | 6.99 | 7.23 |
| $n = 5, f_a = 1,$ $\mathcal{L} = \{5, 5, 5, 5, 20\}$ | 6.13 | 6.15 | 5.66 | 5.7 |
| $n = 5, f_a = 1,$ $\mathcal{L} = \{5, 5, 5, 14, 20\}$ | 7.22 | 9.18 | 6.86 | 9.09 |
| $n = 5, f_a = 2,$ $\mathcal{L} = \{5, 5, 5, 5, 20\}$ | 6.71 | 10.32 | 6.43 | 9.77 |
| $n = 5, f_a = 2,$ $\mathcal{L} = \{5, 5, 5, 14, 17\}$ | 8.17 | 11.85 | 8.11 | 11.04 |

munication schedules (based on sensors' precisions when no other information is available a priori) on performance of attack-resilient sensor fusion.

To conclude this section, we analyze another possible attack strategy, denoted by $AS_2$, and show that the optimization strategy $AS_1$ is worse for the system, i.e., it is a more powerful attack. In $AS_2$, a constant positive offset is added to the attacked sensors' measurements. Once again, the attacker has to guarantee overlap with the fusion interval to avoid detection. Therefore, the schedule and the seen intervals determine if introducing the whole offset would lead to detection, in which case the offset is reduced to the maximal one that would not result in detection.

To compare the two strategies, we note that they can only be compared when the attacker is not last in the schedule, in which case he always has an optimal strategy (specified by $AS_1$). Thus, we only investigate cases in which the attacker has control of the sensors in the middle of the schedule. Similar to the above results, we compute the expected size of the fusion interval for each strategy for different setups. The results are shown in Table 2, where a maximal offset of 3 was introduced and the strategies are compared using the Ascending schedule (the results using the Descending schedule are similar but not shown in the interest of clarity). Note that strategy $AS_1$ always produces a larger expected fusion interval than the $AS_2$, which means it is expected to lead to more powerful attacks.

Table 2: Comparison of the two attack strategies when Ascending schedule is used – $S_1$ is the expectation optimization strategy; $S_2$ is the constant offset strategy.

| | $\mathbb{E}\,\lvert S_{\mathcal{N},f}\rvert$ *Ascending*, $S_1$ | $\mathbb{E}\,\lvert S_{\mathcal{N},f}\rvert$ *Ascending*, $S_2$ |
|---|---|---|
| $n = 3, f_a = 1,$ $\mathcal{L} = \{5, 11, 17\}$ | 10.17 | 9.79 |
| $n = 3, f_a = 1,$ $\mathcal{L} = \{5, 11, 11\}$ | 8.65 | 8.44 |
| $n = 4, f_a = 1,$ $\mathcal{L} = \{5, 8, 17, 20\}$ | 7.54 | 7.16 |
| $n = 4, f_a = 1,$ $\mathcal{L} = \{5, 8, 8, 11\}$ | 6.17 | 5.66 |
| $n = 5, f_a = 1,$ $\mathcal{L} = \{5, 5, 5, 5, 20\}$ | 6.61 | 5.92 |
| $n = 5, f_a = 1,$ $\mathcal{L} = \{5, 5, 5, 14, 20\}$ | 7.35 | 6.92 |
| $n = 5, f_a = 2,$ $\mathcal{L} = \{5, 5, 5, 5, 20\}$ | 7.35 | 5.99 |
| $n = 5, f_a = 2,$ $\mathcal{L} = \{5, 5, 5, 14, 17\}$ | 8.78 | 6.96 |

# 5 Using Measurement History for Attack-Resilient Sensor Fusion

In this section, we explore a complementary approach to improve both the precision and detection capabilites of the sensor fusion algorithm. In particular, we note that most autonomous systems have known dynamics. In this paper, we assume a linear time-invariant system in one dimension $x(t+1) = ax(t)+w$, as outlined in Section 2. Given such a system, this section describes different ways of mapping past measurements to the current round in order to reduce the size of the fusion interval.

First note that the general assumptions of the model used in this work restrict the number of ways of using history. In particular, it is not possible to only map subsets of intervals from previous rounds to the current one as that may not guarantee that the fusion interval will contain the true value. Thus, in our previous work [11] we enumerated the possible ways of using history given our assumptions and identified five ways of mapping past measurements to the current round. Due to space limitations, we only consider three here: the most intuitive one as well as the two best ones, as measured by the size of the obtained fusion interval.

To simplify the equations, we introduce the map

$$m(x(t)) = \{y \in \mathbb{R} \mid ax(t) + w = y, |w| \leq M\}.$$

Here, the mapping of an interval one round to the future is the image of the interval under $m$. In addition, let $R_{\mathcal{N}(t),f}$ denote the set of all intersections of $n-f$ intervals, and let $S_{\mathcal{N}(t),f} = \text{conv}(R_{\mathcal{N}(t),f})$, where conv denotes the convex hull. Note that we use convex hull since the union of disjoint intervals is not an interval. There are three ways to use past measurements as follows:

1. *map_n*: In this algorithm all intervals from time $t$ are mapped to time $t+1$, resulting in $2n$ intervals at time $t+1$ with $2f$ as the new bound on the number of corrupted intervals. Formally the fusion interval can be described as
$$S_{m(\mathcal{N}(t))\cup\mathcal{N}(t+1),2f}.$$

2. *map_R_and_intersect*: This algorithm first maps $R_{\mathcal{N}(t),f}$ and intersects it with $R_{\mathcal{N}(t+1),f}$, after which the convex hull is computed. Formally we describe this as
$$\texttt{conv}(m(R_{\mathcal{N}(t),f}) \cap R_{\mathcal{N}(t+1),f}).$$

3. *pairwise_intersect*: This mapping performs pairwise intersection. Pairwise intersection, denoted by $\cap_p$, means intersecting the mapping of each sensor $s$'s interval from time $t$ to $t+1$ with the same sensor's interval at time $t+1$. This object again contains $n$ intervals. The parameter $f$ used in the fusion algorithm remains the same but an additional assumption is required as discussed below. Formally we capture this as
$$S_{m(\mathcal{N}(t))\cap_p\mathcal{N}(t+1),f}.$$

We now compare the three methods through the size of the fusion interval obtained from each.

**Theorem 5.** *The interval obtained from* map_R_and_intersect *is a subset of the one produced by* map_n.[8]

*Proof.* Consider any point $p \in m(R_{\mathcal{N}(t),f}) \cap R_{\mathcal{N}(t+1),f}$. Then $p$ lies in at least $n-f$ intervals in $\mathcal{N}(t+1)$, and there exists a $q$ such that $p \in m(q)$ that lies in at least $n-f$ intervals in $\mathcal{N}(t)$. Thus, $p$ lies in at least $2n-2f$ intervals in $m(\mathcal{N}(t)) \cup \mathcal{N}(t+1)$, i.e., $p \in R_{m(\mathcal{N}(t))\cup\mathcal{N}(t+1),2f}$, implying

$$\begin{aligned}\texttt{conv}(m(R_{\mathcal{N}(t),f}) \cap R_{\mathcal{N}(t+1),f}) &\subseteq \texttt{conv}(R_{m(\mathcal{N}(t))\cup\mathcal{N}(t+1),2f}) \\ &= S_{m(\mathcal{N}(t))\cup\mathcal{N}(t+1),2f}.\end{aligned}$$

□

To compare *pairwise_intersect* and *map_R_and_intersect*, however, we note that different mappings make different assumptions about the definition of a corrupted sensor. In particular, with the definition used in the first part of this work (i.e., if a sensor is correct in a given round then its interval contains

[8] *map_R_and_intersect* also produces a smaller fusion interval than the other methods described in our previous work that are not listed here.

the true value), one cannot use the pairwise intersection method as it does not guarantee that the fusion interval will contain the true value.[9] In this case, a stricter definition is necessary, namely that a sensor is correct if its interval contains the true value at all time steps. With this in mind, it is possible to strengthen the attack detection mechanism by two more conditions. First of all, if the same sensor's intervals (mapped from previous rounds and the current one) do not intersect, then it must be compromised in at least one of the rounds. Second of all, if the same sensor's intervals' intersection does not intersect the fusion interval obtained from pairwise intersection, then the sensor must also be compromised. If such a definition of correctness is not realistic for a system (e.g., for sensors with transient failures), it could still use *map_R_and_intersect*, which works with the weaker definition of correct sensor.

Another consideration when using history is what the system does when a sensor has been detected to be compromised. If the sensor is believed to be easy to compromise, then the system may choose to ignore its measurements completely. On the other hand, the system may choose to just ignore the current intervals and resume using the sensor in several rounds. We do not investigate the advantages and disadvantages of each approach here; instead, we assume this is a design decision (i.e., input), and leave its analysis for future work.

**Theorem 6.** *Suppose a system discards a sensor's measurements in both the current and previous round if it is detected to be compromised. The interval produced by* pairwise_intersect *is a subset of* map_R_and_intersect*.*

*Proof.* The assumption stated in the first sentence implies that for the given system, each remaining (i.e., non-discarded) interval intersects the fusion interval in the same round. In addition, each sensor's two intervals intersect each other, and this intersection intersects the fusion interval obtained by *pairwise_intersect* method. We assume $n$ and $f$ have been updated accordingly (after compromised sensors are detected).

Without loss of generality, assume that $a > 0$. Let $p$ be the smallest point in $S_{m(\mathcal{N}(t))\cap_p \mathcal{N}(t+1),f}$. Then, $p$ must belong to at least $n-f$ pairwise intersections, and hence lie in at least $n - f$ intervals in $m(\mathcal{N}(t))$.

Consider $R_{\mathcal{N}(t),f}$. It is a collection of, possibly disjoint, intervals that represent the intersections of all combinations of $n - f$ intervals in $\mathcal{N}(t)$. Let $s \in R_{\mathcal{N}(t),f}$ be the interval with lowest lower bound, i.e., $l_s$ is the smallest point contained in $R_{\mathcal{N}(t),f}$. Let $\mathcal{S}(t)$ denote the set of $n - f$ intervals whose intersection yields $s$.

It remains to show that $p \in m(s)$ since then the theorem follows because $p \in m(R_{\mathcal{N}(t),f})$ and $p \in R_{\mathcal{N}(t+1),f}$.

Suppose for a contradiction that $p \notin m(s)$. Note that this implies that $a(u_s) + M < p$; if $a(l_s) - M > p$ then there would not be $n - f$ pairwise intersections that contain $p$. Let $q(t) \in \mathcal{S}(t)$ be the interval with smallest upperbound. Then $p \notin m(q(t))$. But this implies $p \notin m(q(t)) \cap q(t+1)$.

[9] This is true because if different sets of sensors are attacked over time, it is possible that not enough pairwise intersections will contain the true value.

However, this leads to a contradiction since $p$ is in the fusion interval obtained using *pairwise_intersect*, whereas $m(q(t)) \cap q(t+1)$ does not intersect the fusion interval. □

These results suggest that systems that could justify the stronger definition of sensor correctness should use the *pairwise_intesect* method. For other systems we should resort to the *map_R_and_intersect* algorithm. Regardless of which approach is followed, the following results show that using historical measurements is never worse than computing the fusion interval in just one round in isolation, while the benefits are sometimes significant.

**Proposition 1.** *The fusion interval computed using* pairwise_intersect *is never larger than the fusion interval computed without using history.*

*Proof.* Each of the intervals (e.g., $m(P_1(t)) \cap P_1(t+1)$) computed after pairwise intersection is a subset of the corresponding interval when no history is used (e.g., $P_1(t+1)$). Consequently, the fusion interval will always be a subset of the fusion interval obtained when no history is used. □

**Proposition 2.** *The fusion interval computed using* map_R_and_intersect *is never larger than the fusion interval computed without using history.*

*Proof.* Since $m(R_{\mathcal{N}(t),f}) \cap R_{\mathcal{N}(t+1),f} \subseteq R_{\mathcal{N}(t+1),f}$, then
$\text{conv}(m(R_{\mathcal{N}(t),f}) \cap R_{\mathcal{N}(t+1),f}) \subseteq \text{conv}(R_{\mathcal{N}(t+1),f})$, and the proposition follows. □

# 6 Unified Approach for Attack-Resilient Sensor Fusion

In this section, we analyze how the use of an optimal transmission schedule and measurement history in sensor fusion can be combined to complement each other and further improve the performance of the sensor fusion algorithm. We assume the stronger definition of correctness and use the *pairwise_intersect* method in the following comparisons. We also assume that the attacker does not have any limitations, i.e., he is aware of all previous sensor measurements and is able to implement *pairwise_intersect* as well (or any other algorithm).

Similar to the one-round comparison of schedules, we note that no schedule is better than the other in the absolute sense. Therefore, we compare them using the expected size of the fusion interval. As no closed-form solution for this size is available, we compute the value for the same setups as the ones described in Table 1. The system dynamics were assumed to be $x(t+1) = x(t) + w$, with $|w| \leq 1$. Table 3 presents the results. Two things are worth noting. Firstly, once again the Ascending schedule produces smaller-size fusion intervals for all setups. Secondly, as compared with the same setups in Table 1, by adding history the system can further reduce the expected sizes for all setups, even when the attacker also has access to historical measurements.

Table 3: Comparison of the two sensor communication schedules when historical measurements are used.

| | $\mathbb{E}_U \lvert S_{p\_i} \rvert$ *Asc.* | $\mathbb{E}_U \lvert S_{p\_i} \rvert$ *Desc.* | $\mathbb{E}_N \lvert S_{p\_i} \rvert$ *Asc.* | $\mathbb{E}_N \lvert S_{p\_i} \rvert$ *Desc.* |
|---|---|---|---|---|
| $n = 3, f_a = 1,$ $\mathcal{L} = \{5, 11, 17\}$ | 8.59 | 9.65 | 10.03 | 11.37 |
| $n = 3, f_a = 1,$ $\mathcal{L} = \{5, 11, 11\}$ | 7.77 | 8.05 | 9.19 | 9.61 |
| $n = 4, f_a = 1,$ $\mathcal{L} = \{5, 8, 8, 11\}$ | 4.9 | 5 | 6.61 | 6.79 |

**Algorithm 1** Implementation of the *pairwise_intersect* algorithm

**Input:** $f$, an upper bound on the number of corrupted sensors

```
past_meas ← ∅
for each step t do
    cur_meas ← get_meas(t)
    if past_meas == ∅ then
        past_meas ← cur_meas
    else
        past_meas = pair_inter(cur_meas, past_meas)
    end if
    S ← fuse_intervals(past_meas, f)
    send_interval_to_controller(S)
end for
```

Note that *pairwise_intersect* does not add significant computational and memory complexity to the sensor fusion algorithm. In fact, the only additional computation it imposes is the intersection of $n$ pairs of intervals. Furthermore, it requires storing at most $n$ intervals to represent past measurements - intuitively they are the "intersection" of all past measurements.

The implementation of *pairwise_intersect* is shown in Algorithm 1. In essence, at each point in time $n$ intervals (the pairwise intersections) are stored. Thus, *past_meas* represents the "pairwise intersection" of all previous measurements of each sensor. In addition to being more efficient in terms of the size of the fusion interval, the algorithm also needs very little memory – the required memory is linear in the number of sensors irrespective of how long the system runs.

# 7 Case Studies

This section illustrates how the framework proposed in this work can be implemented on an unmanned ground vehicle. We provide both simulation and experimental results using the LandShark [10] robot (shown in Figure 7). The LandShark is mainly used in missions in hostile environments in order to carry

Figure 7: LandShark vehicle [10].

injured people or for reconnaissance of rough terrain.

## 7.1 Simulations

For our simulations, we used the LandShark's velocity sensors. It has four sensors that can estimate speed, namely two wheel encoders, a GPS and a camera. The encoders' intervals were determined based on the measurement error and sampling jitter provided by the manufacturer, whereas the GPS and camera intervals were determined empirically, i.e., the LandShark was driven in the open and largest deviations from the actual speed (as measured by a high-precision tachometer) were recorded for each sensor. The interval sizes (at a speed of 10 mph) were computed to be 0.2 mph for the encoder, 1 mph for the GPS, and 2 mph for the camera.

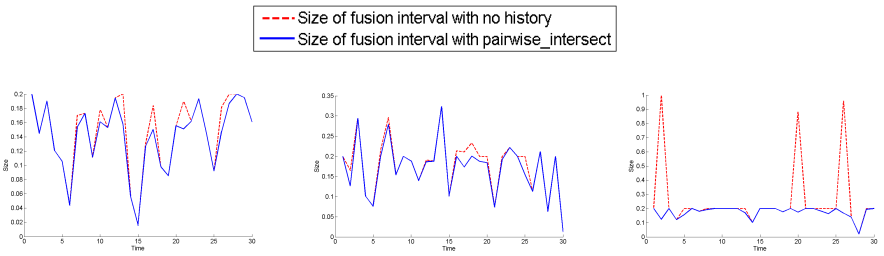
We simulated two different scenarios in order to illustrate the effectiveness of the two approaches discussed in this paper both as separate components and as a unity. The following subsections describe each evaluation in greater detail.

### 7.1.1 Utilize Dynamics and Measurement History

To validate the use of measurement history, we analyzed the fusion interval for the LandShark's velocity when moving straight at a constant speed of 10 mph; we examined the fusion interval when measurements history is used and compared it to the no-history case. In order to use *pairwise_intersect*, we assume that only one sensor is compromised during one run of the system. Thus, we simulated three cases, each one with a different sensor under attack. Since schedules were not investigated in this scenario, an offset attack strategy was chosen such that a constant offset of 1 mph was added to each interval.[10] If an attack is detected, both the current and the previous measurements of the sensor are discarded (thereby reducing both $n$ and $f$ by 1), and new measurements are again collected in the following round.

Figure 8 shows the results for the three cases. As can be seen in the figure, using history never results in larger fusion intervals, whereas in some cases the

[10]Note that this scenario is equivalent to one where a schedule is used and the attacker has to transmit first without a detection constraint.




(a) GPS under attack. (b) Camera under attack. (c) Encoder under attack.

Figure 8: Sizes of velocity fusion intervals for each of the three simulated cases; Dashed line – volume of the fusion interval when measurement history is not considered, Solid line – volume of the fusion interval obtained using *pairwise_intersect*.

reductions in size are significant. Notably, in agreement with Theorem 4, when an attack on an encoder is detected, the resulting fusion interval is much smaller.

### 7.1.2 Unified Approach

To illustrate the advantages of the Ascending schedule, the following scenario was simulated – three LandSharks are moving away from enemy territory in a straight line. The leader sets a target speed of $v$ mph, and the two vehicles behind it try to maintain it for safety reasons. Each vehicle's velocity must not exceed $v + \delta_1$ as that may cause the leader to crash in an unseen obstacle or one of the other two LandSharks to collide with the one in front. Speed must also not drop below $v - \delta_2$ as that may cause the front two vehicles to collide with the one behind. If either of these conditions occurs, a high-level algorithm takes control, switching to manual control of the vehicles. These constraints were encoded via the size of the fusion interval - if the fusion interval contains a point less than or equal to $v - \delta_2$ or greater than or equal to $v + \delta_1$, then a critical violation flag is raised.

We simulated multiple runs of this scenario, each consisting of two rounds. To satisfy the stronger assumption of sensor correctness, the same sensor (randomly chosen at each run) was assumed attacked during the two rounds. In each round random (but correct) measurements were generated for each sensor and then fusion intervals were computed at the end of the second round under the Ascending and Descending schedules (using strategy $AS_1$). For completeness, a different Random schedule was used during each round in order to investigate other schedules that were not analyzed in depth. For each schedule, the fraction of runs was computed that led to a critical violation, as defined in the previous paragraph. The target speed was set to be 10 mph, with $\delta_1 = 0.5$ and $\delta_2 = 0.5$, and system dynamics were assumed to be $x(t+1) = x(t) + w$, with $|w| \leq 10$. The results are shown in Table 4. As can be seen, no critical violations were recorded under the Ascending Schedule, whereas the Descend-

Table 4: Simulation results for each of the three schedules used in combination with *pairwise_intersect*. Each entry denotes the proportion of time that the corresponding schedule generated a critical violation when there was none.

| | Ascending | Descending | Random |
|---|---|---|---|
| **History Used** | | | |
| More than 10.5 mph | 0% | 2.98% | 4.9% |
| Less than 9.5 mph | 0% | 2.63% | 4.8% |
| **No History Used** | | | |
| More than 10.5 mph | 0% | 15.29% | 5.22% |
| Less than 10.5 mph | 0% | 16.8% | 5.61% |

ing and Random schedules both produced some.[11] In addition, adding history has greatly reduced the number of violations, both for the Descending and the Random schedules.

## 7.2 Experimental Validation

In addition to the simulations shown above, experiments were performed using the LandShark robot. They were used to compare the two attack strategies described in the paper as well as to illustrate the advantages of the Ascending schedule regardless of the attack strategy used.

As argued in Section 4, attack strategies can only be compared when the compromised sensors are not at the beginning or end of the communication schedule but in the middle instead. Thus, in the experiments only the mid-schedule sensors were compromised. In the experiments, the LandShark was driven straight and the size of the fusion interval for each scenario was computed as soon as measurements were obtained from all sensors. Note that three sensors were used in the experiments (GPS and two encoders), with the right encoder being in the middle of the schedule, i.e., under attack.

Figure 9 presents the results of the experiments.[12] During the run of the LandShark, the attack (as computed by $AS_1$ and $AS_2$) on the right encoder was turned on and off several times, and we only recorded the fusion interval sizes at the rounds with an attack. Since the rounds were independent, they were concatenated in Figure 9 as if the system was always under attack. The four curves represent the size of the fusion interval for each scenario. As is apparent from the figure, the Ascending curves are almost invariably below, but never above, the Descending. This confirms our results that the use of the Ascending communication schedule reduces the attacker's impact on the performance of

[11] Note that all critical violations recorded under the Descending and Random schedules are false alarms, i.e., the system is not in an unsafe state but is led to believe it is in one due to the attack.

[12] A video with the experiments is available at *http://www.seas.upenn.edu/∼pajic/research/CPS_security.html#videos*.

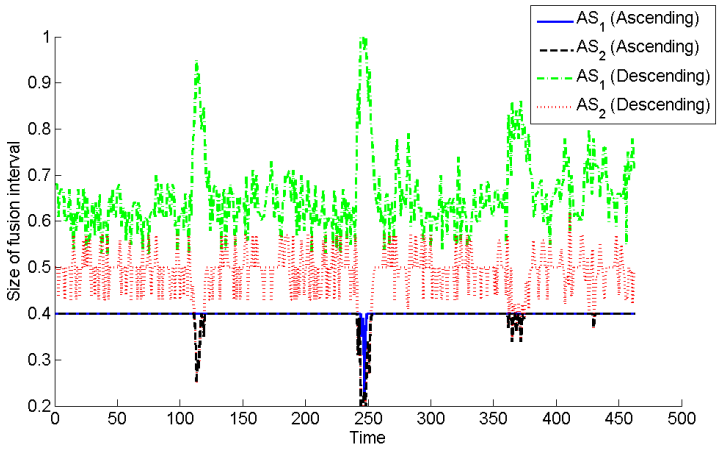



Figure 9: Comparison of the sizes of the fusion intervals as obtained with the two attack strategies, optimization ($AS_1$) and offset ($AS_2$), and two schedules.

Table 5: Average size of the fusion interval for each of the four scenarios.

| | Ascending schedule | Descending schedule |
|---|---|---|
| Optimization strategy | 0.399m/s | 0.652m/s |
| Offset strategy | 0.395m/s | 0.483m/s |

sensor fusion. In addition, it is clear that the optimization attack strategy (i.e., $AS_1$) outperforms the offset one (i.e., $AS_2$) at virtually every round and with both schedules. Finally, Table 5 summarizes the results by providing the average size of the fusion interval for each scenario.

# 8 Related Work

Providing security for Cyber-Physical and Embedded Systems is a challenging task [12]. One way of addressing it is through exploiting the fact that these systems have multiple sensors, whose data may be calibrated and fused for better closed-loop performance [13]. Yet, the term "sensor fusion" has different interpretations and applications in different fields of research. In some areas it is considered to be the process of collecting and combining the data from similar sensors measuring the same variable. In others, however, it is synonymous to the broader term "state estimation", in which different sensors measure different aspects of the system's state. This paper uses the first definition as it implies sensor redundancy, which is a part of the model used in this work.

The work on sensor fusion can be divided according to the sensor model used. The far more prevalent approach is to use a probabilistic model and

derive expected results such as in the pivotal work in this domain, the Kalman filter [6]. In that work, assumptions about sensor precisions are combined with a known system dynamics model in order to produce a best linear estimator of the true state. In addition, distributed versions of this model have been proposed as well [14, 7]. All probabilistic works, however, are concerned with the average performance of a system and are not well-suited for the analysis of low-probability rare events.

On the other hand, the abstract sensor model is usually employed for worst-case analysis. One of the first works in this field [8] assumes that sensors provide one-dimensional intervals and shows worst-case results regarding the size of the fused interval based on the number of faulty sensors in the system. A variation of [8] relaxes the worst-case guarantees in favor of obtaining more precise fused measurements through weighted majority voting [15]. Another extension combines the abstract and probabilistic models by assuming a probability distribution of the true value inside the interval and casting the problem in the probabilistic framework [16]. Finally, sensors can be assumed to not only provide intervals but also multidimensional rectangles and balls [17] and more general sets as well [18, 19]. Another advantage of the abstract sensor model is that it can be used not only for safety analysis but for fault detection as well [8, 20].

Finally, some works propose performing sensor fusion independently of the sensor model. In particular, if instead of a measurement, the sensor's output is a decision such as whether to raise an alarm or not, a higher-level fusion algorithm has to combine the sensor decisions instead of their measurements. This problem is usually solved with a voting scheme [21, 22] or a fuzzy voting technique [23].

Another term that is used differently across areas of research is "sensor scheduling". While in some works, including this one, it refers to the communication schedule of sending measurements during every round of system operation, in others the schedule refers to which sensors should be utilized in a given round. Thus, the difference between the two is that in the former all sensors are utilized at all time steps, whereas in the latter only subsets of the sensors are used at each time in order to minimize energy consumption or interference. Different approaches for the latter definition of sensor scheduling have been proposed, ranging from pruning techniques [24] to convex optimization [25] to information theory [26].

# 9 Discussion and Conclusion

In this paper we described an attack-resilien sensor fusion algorithm for multiple sensors measuring the same variable. We introduced security concerns by formalizing an attack strategy that attempts to maximize the uncertainty in the system by increasing the set of possible measurement values obtained from sensor fusion. Two approaches of improving the precision and resiliency of sensor fusion were investigated. On the one hand, we showed that different transmission schedules affect the information and capabilities of the attacker.

Our results showed that the Ascending schedule is expected to produce the most precise fusion intervals by either providing the attacker with too little information (acting first in the schedule when compromising more precise sensors) or too little power (when compromising imprecise sensors). On the other hand, we showed that knowledge of system dynamics can be utilized with sensor measurement history in order to further improve the precision and resilience of the algorithm. Finally, we showed that by using the optimal communication schedule (i.e., Ascending) and the sensor fusion algorithm with measurement history, we can further reduce the attacker's impact on the system. We validated our findings and illustrated the use of the proposed sensor fusion approach on a real-world case study, velocity estimation on an unmanned ground vehicle.

There are several ways in which the algorithm can be improved. Naturally, if information is available as to which sensors are harder to compromise, it can be incorporated by scheduling those sensors to transmit last, thus precluding the attacker from seeing their measurements. In addition, while this work focused mainly on the Ascending and Descending schedules, other schedules were considered in Section 7.[13] Even there, however, the Ascending schedule produced no violations, whereas the Random schedules led to a few.

Regarding the problem of using measurement history, it was noted that the accuracy of the different mapping algorithms depends on the definition of a compromised sensor. However, in this work we assume that sensors are either correct or compromised. A next step would be to allow a fault model for sensors to be included in the fusion algorithm. For example, we could introduce a temporal fault model where sensors are allowed to be faulty in a few rounds without being immediately discarded as compromised.

# References


[1] 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*, 2010, pp. 447–462.

[2] 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*, 2011, pp. 6–6.

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

[4] J. Warner and R. Johnston, "A simple demonstration that the global positioning system (gps) is vulnerable to spoofing," *Journal of Security Administration*, vol. 25, pp. 19–28, 2003.


[13] Although these schedules were collectively called Random, the system was run for sufficient time in order to generate all possible other schedules.

[5] Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava, "Non-invasive spoofing attacks for anti-lock braking systems," in *Cryptographic Hardware and Embedded Systems - CHES 2013*, ser. Lecture Notes in Computer Science, 2013, vol. 8086, pp. 55–72.

[6] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[7] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *IPSN'05*, 2005, pp. 63–70.

[8] K. Marzullo, "Tolerating failures of continuous-valued sensors," *ACM Trans. Comput. Syst.*, vol. 8, no. 4, pp. 284–304, Nov. 1990.

[9] 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*, 2014, pp. 163–174.

[10] "The LandShark," 2009, http://blackirobotics.com/LandShark_UGV_UC0M.html.

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

[12] D. N. Serpanos and A. G. Voyiatzis, "Security challenges in embedded systems," *ACM Transactions on Embedded Computing Systems*, vol. 12, no. 1s, pp. 66:1–66:10, Mar. 2013.

[13] R. Tan, G. Xing, X. Liu, J. Yao, and Z. Yuan, "Adaptive calibration for fusion-based cyber-physical systems," *ACM Transactions on Embedded Computing Systems*, vol. 11, no. 4, pp. 80:1–80:25, Jan. 2013.

[14] V. Delouille, R. Neelamani, and R. Baraniuk, "Robust distributed estimation in sensor networks using the embedded polygons algorithm," in *IPSN'04: Proc. 3rd International Symposium on Information Processing in Sensor Networks*, 2004, pp. 405–413.

[15] R. R. Brooks and S. S. Iyengar, "Robust distributed computing and sensing algorithm," *Computer*, vol. 29, no. 6, pp. 53–60, Jun. 1996.

[16] Y. Zhu and B. Li, "Optimal interval estimation fusion based on sensor interval estimates with confidence degrees," *Automatica*, vol. 42, no. 1, pp. 101–108, 2006.

[17] P. Chew and K. Marzullo, "Masking failures of multidimensional sensors," in *SRDS'91: Proc. 10th Symposium on Reliable Distributed Systems*, 1991, pp. 32–41.

[18] M. Milanese and C. Novara, "Set membership identification of nonlinear systems," *Automatica*, vol. 40, no. 6, pp. 957–975, 2004.

[19] ——, "Unified set membership theory for identification, prediction and filtering of nonlinear systems," *Automatica*, vol. 47, no. 10, pp. 2141–2151, 2011.

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

[21] N. Katenka, E. Levina, and G. Michailidis, "Local vote decision fusion for target detection in wireless sensor networks," *Signal Processing, IEEE Transactions on*, vol. 56, no. 1, pp. 329–338, Jan 2008.

[22] Z. Chair and P. Varshney, "Optimal data fusion in multiple sensor detection systems," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. AES-22, no. 1, pp. 98–101, Jan 1986.

[23] S. Blank, T. Fohst, and K. Berns, "A fuzzy approach to low level sensor fusion with limited system knowledge," in *Information Fusion (FUSION), 2010 13th Conference on*, July 2010, pp. 1–7.

[24] M. P. Vitus, W. Zhang, A. Abate, J. Hu, and C. J. Tomlin, "On efficient sensor scheduling for linear dynamical systems," *Automatica*, vol. 48, no. 10, pp. 2482–2493, 2012.

[25] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2009.

[26] J. Williams, "Information theoretic sensor management," Ph.D. dissertation, MIT, 2007.