CONTEXT-AWARE SENSOR FUSION FOR SECURING CYBER-PHYSICAL SYSTEMS

Radoslav Svetlozarov Ivanov

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania

in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2017

Supervisor of Dissertation	Co-Supervisor of Dissertation
Insup Lee Professor Computer and Information Science	James Weimer Research Assistant Professor Computer and Information Science

Graduate Group Chairperson

Lyle Ungar
Professor
Computer and Information Science

Dissertation Committee:

Oleg Sokolsky, Research Professor of Computer and Information Science

Rahul Mangharam, Associate Professor of Electrical and Systems Engineering

George J. Pappas, Professor of Electrical and Systems Engineering

Paulo Tabuada, Professor of Electrical Engineering, University of California, Los Angeles

Miroslav Pajic, Assistant Professor of Electrical and Computer Engineering, Duke University

CONTEXT-AWARE SENSOR FUSION FOR SECURING CYBER-PHYSICAL SYSTEMS

To my family.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisors, Insup Lee, James Weimer and Miroslav Pajic. Insup has shown tremendous support throughout my stay at the University of Pennsylvania, not only in the form of academic guidance but also in encouraging and motivating me when the going got tough. He has taught me to think independently and critically but also to keep an open mind and embrace new opportunities. I would definitely not have become the researcher (and person!) that I am today had it not been for our interaction.

I am also very grateful to Jim for endless discussions and arguments in the search for truth (we like to think that we found it every once in a while). His undying enthusiasm has always been a great driving force for me, both at work and in life in general (and in bars, in specific). Furthermore, this page would be incomplete without an honorable mention of my third advisor Miroslav who helped shape a cheeky undergrad into a professional researcher (of course without missing an opportunity to add some Balkan flavor to our conversations).

I would also like to thank my committee, Oleg Sokolsky, Rahul Mangharam, George Pappas and Paulo Tabuada for their insightful feedback and helpful suggestions for improving my thesis and presentation. I thank Junkil for helping with some of the sensor fusion work (including the interesting experiments he performed) as well as Nikolay for his great help with the context-aware work. In addition, I would like to thank all my friends in Philly – beginning once again with Nikolay, Kalin, Arthur, Nikos, Shahin, Justin, Venetia and Fragkiskos, my friends from Colgate – Boyan, Bogdan, Dobromir and Nikolay, and all my friends from Sofia who have maintained their love and support throughout our many years of friendship – Veso, Tosho, Mimi, Kina, Georgi, Rosen, Mitko, Moni and Tseni.

Finally, I thank my family, Gergana, Svetlozar and Atanas, for their constant support and love, for understanding my complaints and for cheering me up when I most needed it. I also thank my grandparents, Kera, Slavcho, Marina and Nasko, for their interest and support. I will never forget my grandmother Kera's words: "I am so proud that we will have a professor in the family!".

ABSTRACT

CONTEXT-AWARE SENSOR FUSION FOR SECURING CYBER-PHYSICAL SYSTEMS

Radoslav Svetlozarov Ivanov

Insup Lee

James Weimer

The goal of this dissertation is to provide detection and estimation techniques in order to ensure the safety and security of modern Cyber-Physical Systems (CPS) even in the presence of arbitrary sensors faults and attacks. We leverage the fact that modern CPS are equipped with various sensors that provide redundant information about the system's state. In such a setting, the system can limit its dependence on any individual sensor, thereby providing guarantees about its safety even in the presence of arbitrary faults and attacks.

In order to address the problem of safety detection, we develop sensor fusion techniques that make use of the sensor redundancy available in modern CPS. First of all, we develop a multidimensional sensor fusion algorithm that outputs a bounded fusion set which is guaranteed to contain the true state even in the presence of attacks and faults. Furthermore, we provide two approaches for strengthening sensor fusion's worst-case guarantees: 1) incorporating historical measurements as well as 2) analyzing sensor transmission schedules (e.g., in a time-triggered system using a shared bus) in order to minimize the attacker's available information and impact on the system. In addition, we modify the sensor fusion algorithm in order to provide guarantees even when sensors might experience transient faults in addition to attacks. Finally, we develop an attack detection technique (also in the presence of transient faults) in order to discard attacked sensors.

In addition to standard plant sensors, we note that modern CPS also have access to multiple environment sensors that provide information about the system's context (e.g., a camera recognizing a nearby building). Since these context measurements are related to the system's state, they can be used for estimation and detection purposes, similar to standard measurements. In this dissertation, we first develop a nominal context-aware filter (i.e., with no faults or attacks) for binary context measurements (e.g., a building detection). Finally, we develop a technique for incorporating context measurements into sensor fusion, thus providing guarantees about system safety even in cases where more than half of standard sensors might be under attack.

Contents

List of Tables

List of Illustrations

Chapter 1

Introduction

The steady advancements in sensing, actuating and computing technology over the past few decades have enabled the development of increasingly sophisticated systems. In fact, fully or partially autonomous systems can now be found in multiple and diverse applications such as aircraft [70], automobiles [201], surgical systems [203], smart grids [40], drones [2] and robotics [202]. What is common between all these systems is the tight coupling of hardware and software capabilities and constraints as well as the combination of heterogeneous components with various assumptions and guarantees about their performance. This great complexity has highlighted the need for the theory of Cyber-Physical Systems (CPS) at the intersection of control and information theory, formal methods and embedded systems.

As the autonomy of modern CPS increases, however, so does the concern about their safety – all of the examples cited in the previous paragraph are systems that are in direct contact with or operate in the presence of people. What is more, safety worries are not just a theoretical artifact; deadly CPS crashes have occurred due to failures of components at all levels of system design, including sensors, actuators, software and human operators (or any combination thereof). The following list contains some recent examples of such failures:

- Sensor faults. An example of such a failure was the recent Tesla autonomous driving crash on May 7, 2016 during which the Tesla driver was killed. Investigators concluded that the crash was (partly) due to the inability of the camera to recognize cross traffic (Tesla avoided blame as their Autopilot system was not responsible for handling cross traffic) [8].

- Actuator faults. We illustrate this class of failures with an example from medical CPS due to their high safety-criticality. The widely used Da Vinci Surgical System [203] was forced to perform a major recall of some parts due to an increased number of accidents. The main cause of malfunction was friction within certain instruments that affects the surgeon's actions [6, 166].
- Imperfect human-CPS interaction. Several aircraft accidents were caused to a different degree by this type of failure. One of the most notable is the crash of Air France Flight 447 off the coast of Brazil on June 1, 2009. According to the official report, the reason for the accident was the deadly combination of incorrect speed readings by the pitot tubes and an inadequate reaction by the crew [63].

The problem of ensuring system safety is further exacerbated by security concerns – due to the quickly rising number of their applications (including military and medical applications), modern CPS are increasingly being subjected to malicious attacks preventing them from correctly performing their tasks. In fact, the potential attack surface has considerably grown with the introduction of computationally powerful and interconnected components. On the one hand, since CPS consist of both physical and cyber components that perform critical functions, they are by definition vulnerable to all conventional physical attacks such as breaking or tampering with a certain part as well as standard cyber attacks such as buffer overflow or denial of service. At the same time, multiple cyber-physical attacks exploiting the interaction between the two layers (e.g., communication vulnerabilities or flawed estimation/control approaches) have now been carried out as well. A few notable classes of CPS-specific attacks are listed below:

- Sensor spoofing. In this class of attacks, sensor measurements are altered by manipulating the system's environment, including the communication medium. Examples of this attack include a yacht that was carried off course due to spoofed GPS readings [172] as well as the RQ-170 Sentinel drone that was captured in Iran [164, 182] (while the details of the capture are not publicly available, it is widely believed that hijackers captured the drone through jamming the GPS signal).

- Software vulnerabilities. As demonstrated by the Stuxnet virus [77], critical industrial infrastructure could be disabled by exploiting software bugs and thereby gaining control of key components.

- Communication protocol vulnerabilities. In some systems, e.g., automotive CPS, it is possible for the attacker to compromise the entire system by gaining access to the internal network and exploiting communication vulnerabilities such as the lack of authenticity in the Controller Area Network (CAN) bus [47, 119].

As illustrated in the above examples, modern CPS can crash in multiple and unpredictable ways, caused both by arbitrary failures as well as by malicious attacks. Furthermore, these crashes are often caused not just by a single component failing but by a combination of Byzantine faults or crafty attacks that are difficult to detect or prevent. Therefore, it is clear that there is a need for a unifying theory for the design and analysis of modern CPS – this theory needs to not only ensure the safety of these systems under nominal conditions but it also has to provide safety and security guarantees even in the presence of unknown faults and attacks in the individual components.

1.1 High-Level Goal

As described in the previous section, ensuring the safety and security of modern CPS requires considering all possible interactions of the system's components as well as all possible ways in which these components can fail or be attacked. As a first step towards addressing this daunting task, we focus on a crucial aspect of any autonomous system's operation, namely providing precise information about its state and context. Precise information (e.g., a vehicle's location estimate) is a necessary condition for guaranteeing the system's safety – for example, it is not possible to ensure automobiles will not crash in traffic that cannot be detected; similarly, a boat cannot guarantee it will reach its destination if its location measurements are spoofed. Thus, the goal of this dissertation is to develop a general approach for detecting when the system is unsafe or insecure; furthermore, this technique needs to provide guarantees about the system's safety **regardless of the fact that some system components might be faulty/attacked**.

To state the above goal more concretely, consider a system (e.g., a robot), denoted by S, operating in an environment E (e.g., surrounding people and obstacles) in the presence of a possible attacker A; the scenario is illustrated in Figure 1.1. The system consists of the three typical components of any autonomous system: (1) a plant (including actuators), (2) sensors, and (3) a control algorithm. The system has an internal state, denoted by x_k, signifying various aspects of its execution at time k such as location, velocity, etc. The state evolves (in discrete time) as a function of the previous state and the applied inputs:

$$x_{k+1} = f(x_k, u_k), \tag{1.1}$$

where u_k is the control input and f is a function mapping current states and inputs to future states. The controls u_k are computed by the control algorithm as a function,

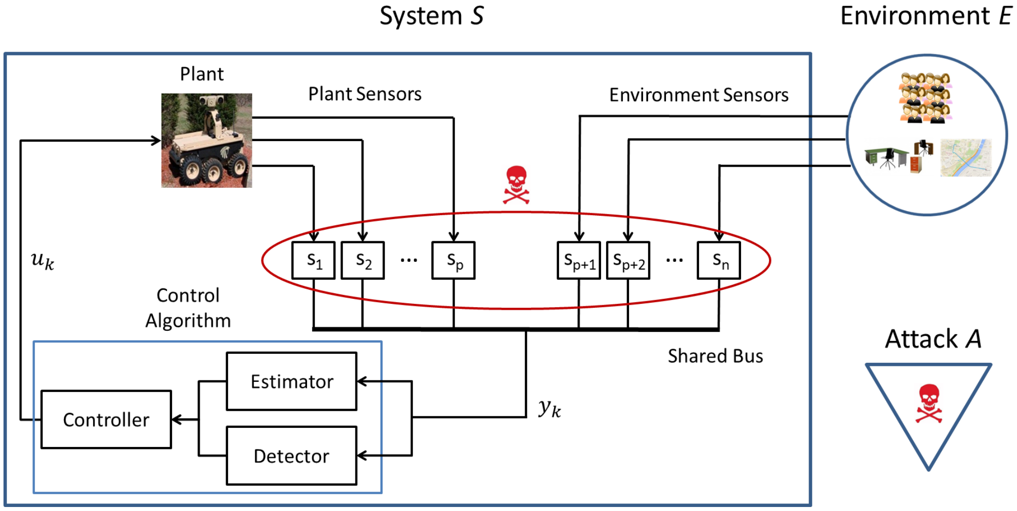


Figure 1.1: Overview of the system architecture considered in this dissertation. The system operates in an environment that may include people, obstacles, etc. Sensors are available to measure both the plant and the environment's states. At the same time, a malicious attacker might be able to compromise all vulnerable system components. In this dissertation, we assume attacks can occur in sensors only.

denoted by g, of all received measurements:

$$u_k = g(y_{0:k}), \tag{1.2}$$

where $y_{0:k}$ denotes all measurements received from time 0 up to k. The system has sensors available for measuring both the system and the environment's state such that at each time k, the received measurements y_k are:

$$y_k = h(x_k, x_k^E), \tag{1.3}$$

where x_k^E is the environment's state, which contains relevant elements to the system, e.g., location of people and obstacles, nearby buildings, texture of surface. The environment's state also evolves in time according to some (unknown) dynamics.

Finally, as shown by the examples in the previous section, the attacker might compromise multiple system components and modify their behavior in an arbitrary way. At the same time, such an all-powerful attacker makes the problem of providing any kinds of guarantees about the system undecidable. Thus, to keep the problem tractable and as justified in the following section, we assume attacks can occur in sensors only.

Given the description of the three of entities in Figure 1.1, the control algorithm's task is to reach a desired state (e.g., a destination), while preserving a given safety property p_{safe}, which is a predicate on the system and environment's states (e.g., the robot should not collide with people). The control algorithm is split into three components: (1) a state estimator that estimates the system's state and (2) a detector that raises alarms when the system is unsafe or under attack; the estimator and detector's outputs are in turn used by the (3) controller in order to compute control inputs for the actuators. This modular design is preferred in most systems to a unified control approach (e.g., optimal control using dynamic programming [26]) due to its greater flexibility. While classical optimal control techniques such as dynamic programming might find the optimal control policy in many scenarios, they become computationally intractable (or even undecidable) in the case of faults and attacks. The modular formulation, on other hand, can handle such scenarios by ensuring that each of the three components provides guarantees about its output (e.g., the detector does not raise any safety false alarms) – thus the control algorithm can provide guarantees on its final output as well (e.g., the state will be within some epsilon from the desired state).

As described at the beginning of this section, **the high-level problem is to develop detection and estimation techniques for establishing whether the system is safe and secure**, i.e., for the property p_{safe}. These techniques must also provide guarantees about their performance **even in the presence of faulty/attacked system components** (the exact form of these guarantees is stated in Section 1.5; an example is that the detector never says p_{safe} is true when it is false, i.e., there are no false alarms). Given these guarantees on the estimator and detector components, it is **assumed that the controller component is developed in such a way so as to ensure the overall task of the control algorithm**.

1.2 Challenges for Detection and Estimation in Modern CPS

Designing estimators and detectors for general CPS with arbitrary models is a challenging task. First of all, since CPS are a subset of the broad class of control systems, they present the standard challenges involved with modeling complex systems, namely that if the f and h functions are non-linear or discontinuous, then few optimal and computationally tractable algorithms exist. In addition, actual CPS never behave according to their models – sensors and actuators experience faults, control code has bugs, and systems are often subjected to malicious attacks on multiple components. This subsection lists two of the main CPS-specific challenges for detection and estimation algorithms as well as the corresponding simplifying assumptions that we make in order to make these challenges manageable.

1.2.1 Detection and Estimation in the Presence of Faults

As mentioned before, all system components experience faults during their lifetime. For the purpose of this discussion, a fault is defined as any behavior that does not match the system model, i.e., the component's expected behavior. Faults are dangerous because they disrupt system performance – since most systems are developed with a given (parameterized) model/expectation in mind, faults may render these systems unsafe and lead to crashes. Faults can be transient and recover on their own (e.g., a GPS losing signal in a tunnel but regaining it after that) but can also be permanent and irreparable (e.g., a broken actuator).

In some well-studied systems, it might be possible to plan for faults and react accordingly. However, given the complexity of modern CPS, it is impossible to predict and guard against all kinds of faults in such systems. Therefore, it is difficult to justify any assumption about the class or timing of faults in CPS. At the same time, making no assumptions at all means the problem is undecidable – if faults are allowed to happen at any time and in any way, then no control algorithm can meet its goals in the worst case.

We alleviate this problem by restricting our attention to sensor faults only and assuming that actuators (and other components) behave as modeled. As will be explained in Section 1.3, the reason it is possible to provide guarantees even with arbitrary sensor faults is that we can utilize the inherent sensor redundancy available in modern CPS. On the other hand, no such redundancy can be exploited in actuators or other components; thus, handling more general faults in actuators (and other components) is left for future work.

Assumption. *We assume that (arbitrary) faults can occur in sensors only. Actuators and other components are assumed to behave as modeled.*

1.2.2 Detection and Estimation in the Presence of Attacks

Similar to faults, malicious attacks are another major challenge for CPS analysis and development. Once an attacker finds a vulnerability in a component, it is often possible to gain full control of that component and force it to behave in any desired way. Thus, attacks are similar to faults in the sense that they force the attacked component to behave unexpectedly. At the same time, they are different as they are always targeted and malicious whereas faults are often benign and transient; therefore, attacks require a special treatment when ensuring systems' resilience.

Once again similar to the fault case, the problem of securing the system becomes undecidable if we assume that attacks can occur in all components and in all ways. To overcome this problem, we leverage the redundant-sensor framework as well – similar to the previous subsection, we focus on sensor attacks only and leave the problem of providing resilience to attacks on all components as part of future work.

Assumption. *We assume that attacks can occur in sensors only. Actuators and other components are assumed to behave as modeled.*

1.3 Multi-Sensor Systems

As argued in the previous section, ensuring the safety of arbitrary CPS becomes challenging when one considers the full generality of the problem. That is why, in this dissertation we restrict our attention to faults/attacks in sensors only. Even in this case, however, if sensors are modified by faults or attacks in a manner that causes them to behave in unpredictable ways, then developing detection and estimation techniques is not possible without additional assumptions. That is why, traditional approaches usually make simplifying assumptions so as to scope the problem (a more thorough review of related work is provided in Chapter 2); more specifically, **at least one of the following assumptions is made in most standard approaches to fault/attack detection**:

1. **Rich training data containing attacks/faults is available.** Such systems are able to detect and recover from attacks and faults by examining and learning from previous occurrences of the same anomaly [46].

2. **The system nominally operates in a known state**. This a very common assumption in related works, including intrusion detection systems [22, 131, 137], fault detection approaches [221], etc. Starting from a known nominal condition allows these approaches to perform a variant of change detection where an alarm is raised when an unexpected/unlikely behavior is observed.

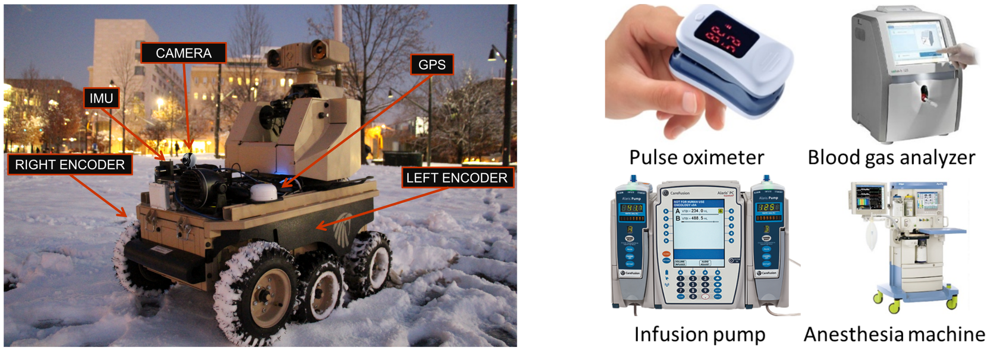
3. **The fault/attack has a known effect or comes from a known class of faults/attacks**. This assumption makes it possible to design detectors aimed specifically at these classes of faults/attacks (e.g., by detecting a change of system parameters [23, 24, 177, 178]).

While the above assumptions might be reasonable in simpler or well-understood systems, they are difficult to justify in modern CPS. In particular, as shown in the introductory examples, CPS can fail in intricate and unpredictable ways that have not been observed before (thus invalidating assumptions 1 and 3). In addition, they are designed to operate in hostile and rough environments – in fact, these systems can never assume they are in known nominal states (assumption 2) since undetected data injection attacks can produce arbitrary errors in nominal state estimation algorithms, e.g., in the case of a perfectly attackable system [128, 147].

This is why, in this dissertation we do not make any of the above assumptions and take a different approach, namely we leverage the fact that modern CPS are equipped with multiple sensors that can be used to provide redundant information about certain aspects of the system's state (e.g., a GPS and an IMU can both be used to estimate speed, even though neither one measures it directly). In this setup, the dependence on any individual sensor can be reduced – while some sensors might provide spurious data, whether due to a fault or an attack, the system will still be safe if it uses the sensors that are operating normally. This means that no assumptions are necessary about how and when faults/attacks might occur in any specific sensor, information that might be difficult to obtain a priori.

At the same time, this framework comes at a cost as well – at least half of the sensors must operate according to specification in order for the approach to work. However, by developing multiple and diverse sensors, system designers can make it unlikely for many sensors to fail at the same time. Increased sensor diversity also makes it harder for an attacker to corrupt a large number of sensors; for example, all known spoofing techniques, whether physical attacks [185] or cyber spoofing [47, 119], require significant effort and time investment, which makes it difficult for attackers to simultaneously control different sensors in a given system.

To better illustrate the benefit of multi-sensor systems, we now provide two examples of systems where multiple sensors are available. First consider the LandShark



(a) The LandShark robot [7]. It has access to five sensors that can be used to estimate velocity: left and right encoders, an IMU, GPS and a camera.

(b) Typical medical devices used in an operating room: a pulse oximeter [4], a blood gas analyzer [3], an infusion pump [1] and an anesthesia machine [5].

Figure 1.2: Example systems with multiple sensors that can be used to estimate the same variables.

robot [7], as illustrated in Figure 1.2a. The LandShark is a heavy-duty vehicle, designed to operate on rough terrain and hostile territory; therefore, it needs to be resilient to multiple kinds of sensor and actuator failures as well as malicious attacks. The LandShark is equipped with five sensors: two wheel encoders, a GPS, an IMU and a camera. While these sensors measure different physical variables, they can all be used to estimate the vehicle's velocity as well. Thus, if one of these sensors is faulty or attacked, it might be possible to detect that its respective measurements are inconsistent with the other sensors' and raise an alarm.

Similarly, Figure 1.2b shows that modern operating rooms (ORs) also have multiple measuring devices as well. At minimum, a typical OR has a pulse oximeter [4], a blood gas analyzer [3], an anesthesia machine [5] and an infusion pump [1]; depending on the case, other devices might be present as well such as a ventilator. These devices provide clinicians with various data about the patient's vital signs – although it is not straightforward to map some vital signs to others, certain relationships between them can be used in order to conclude that a sensor is not behaving according to specification (e.g., if the pulse oximeter is showing low hemoglobin oxygen saturation but the blood gas analyzer measured a high blood oxygen concentration, then

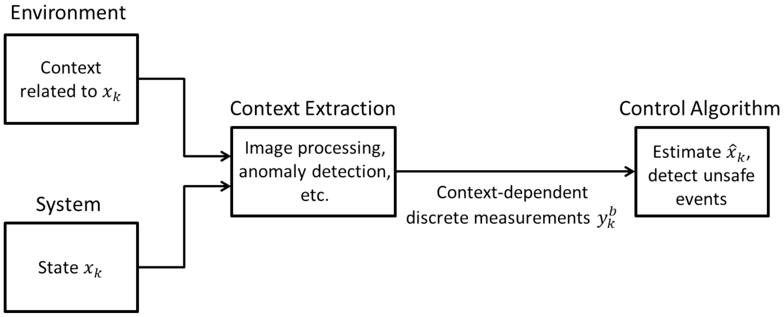

Environment
Context related to x_k
System
State x_k
Context Extraction
Image processing, anomaly detection, etc.
Context-dependent discrete measurements y_k^b
Control Algorithm
Estimate $\hat{x}_k$, detect unsafe events

Figure 1.3: General architecture of a system with access to context measurements.

one of the two readings must be incorrect).

Combining data from diverse, but redundant, sensors lies in the broad field of sensor fusion. The sensor fusion framework used in this dissertation is formalized in Section 1.5. In the following section, we present another source of information that can be used to enhance the capabilities of sensor fusion.

1.4 Context-Aware CPS

Although sensor redundancy enables us to avoid making unrealistic assumptions about the occurrence of faults and attacks, it has its own limitations as well. In particular, in order to provide guarantees about its output, it requires that at least half of all sensors operate correctly – this is a theoretical barrier that cannot be overcome without using additional information [78]. One such source of information is state-related context that can be extracted from the system's environment.

With the proliferation of sensing and computing technology, modern CPS have access to a wealth of information about their environment as provided by their environment sensors. This information is rarely useful for estimation purposes since it is too low-level and challenging to map to the state. However, given the recent improvements in machine learning, it is now possible to obtain high-level representations of this information. For example, if a robot detects a known building using image processing, the robot can conclude that it is near that building; similarly, if a medical device raises an alarm that a vital sign is above a certain threshold, it might be possible to conclude that the patient is in a critical state. Consequently, these discrete-valued context data can be viewed as measurements of (functions of) the system state, similar to conventional continuous sensors such as IMU or GPS (this notion is illustrated in Figure 1.3). Thus, context measurements can be used for estimation and detection both as a single source of information and in scenarios when some of the continuous plant sensors are noisy/biased (e.g., GPS in an urban environment [118] or medical sensors disrupted by moving artifacts [107]) or in security applications when some sensors might be attacked (e.g., GPS spoofing [164]).

In this dissertation, we are specifically interested in binary measurements as an important subclass of context measurements, i.e., each measurement takes on a value of 1 or -1. Binary measurements capture a rich class of scenarios and events that might occur during a system's operation. Examples of binary context measurements include a medical device alarm that a vital sign exceeds a certain threshold (e.g., if the patient's oxygen saturation is above a certain threshold, then the overall oxygen content (the state) must be above a certain threshold [108]) as well as occupancy grid mapping where a binary measurement is received as the robot gets close to an obstacle [196].

Since using context measurements for estimation and detection purposes is a novel idea in itself, one of the contributions of this dissertation is the development of nominal estimation algorithms using context measurements (i.e., without faults or attacks). In this setting, context measurements are defined as any binary data that have a known probability of occurring given the system state. Context measurements are especially useful when they represent low-level data that cannot be easily expressed as a function of the state (e.g., it is challenging to functionally map raw images to the state of a robot) – thus, by using the probability distribution of context measurements given the state, one may use them for estimation in a rigorous manner.

In addition to nominal state estimation, context measurements can be used to enhance sensor fusion techniques for safety detection as well. Since we are interested in providing worst-case guarantees in this framework, when a context measurement is received, a set is constructed that contains all possible values for the true state (e.g., a rectangle around a building inside which the building could be detected using image processing). As formulated in the following section, this treatment of context measurements fits exactly into the standard sensor fusion setup.

1.5 Problem Formulation

Having motivated the problem, the shortcomings of existing work as well as our approach in the previous sections, in this section we summarize all components and provide the specific problem statements addressed in this dissertation.

1.5.1 Problem Space

In order to provide context for the specific problem statements, we first outline the general problem space of this dissertation. As described in Section 1.1, **we focus on the detection and estimation components of standard CPS that might experience faults and attacks in their sensors only**. At a high level, there are four levels of increasing attack resilience in modern CPS (illustrated in Figure 1.4):

1. **Nominal State Estimation**. When no attacks are present (or suspected), the system performs nominal state estimation.

2. **Attack Prevention**. This includes techniques such as encryption [65, 168, 180], authentication [14], trusted computing [123, 205] and others that attempt to prevent attackers from disrupting the system's performance.

3. **Attack Detection and Resilient State Estimation**. If the system operates in a hostile environment where attacks are possible, then it needs to develop

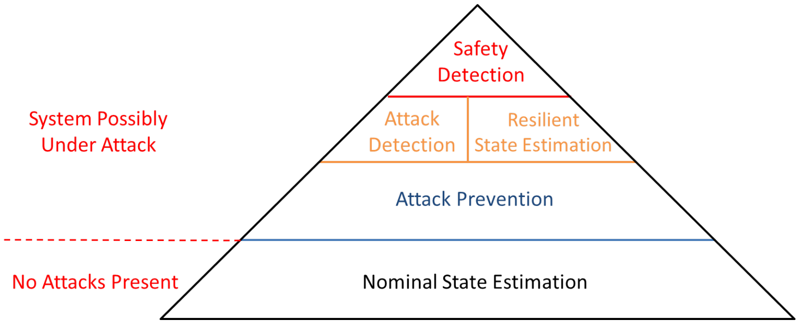
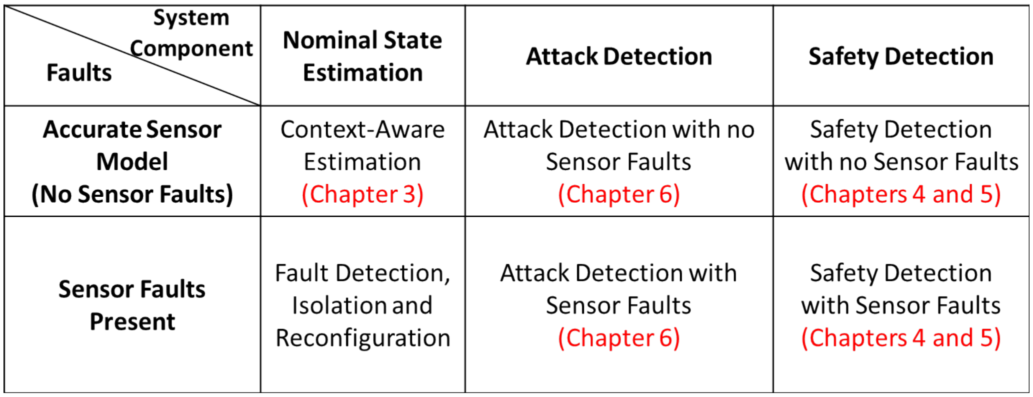


Figure 1.4: Increasing levels of attack resilience in modern CPS. When the system is under no threat of attacks, it performs nominal state estimation. If attacks might be present, the system has access to attack prevention, attack detection, resilient state estimation and, as a last line of defense, safety detection. This dissertation focuses mostly on safety detection but also investigates related aspects of attack detection and nominal context-aware estimation.

corresponding detection and estimation techniques. First of all, detectors are developed for attacks in various system components. In addition, the system performs resilient state estimation such that it provides guarantees on its state estimates even if some sensors are under attack.

4. **Safety Detection**. Regardless of whether an attack exists, the system runs the Safety Detection component in order to verify that it is in a safe state.

In this dissertation, we are primarily interested in item 4 in the above list; safety detection can be considered as a last line of defense for the system – even if attacks are present, the system might be able to avoid crashing if it can detect when it is in an unsafe state. Furthermore, note that safety detection is aided by attack detection as well – if a sensor is identified as attacked, it can be discarded, thereby improving the performance of all other system components; thus, **we also develop techniques for sensor attack detection/identification**. Finally, as noted in Section 1.4, we make use of context measurements in addition to classical plant measurements; since no approaches exist to incorporate such discrete measurements in estimation/detection algorithms, **we develop a technique for context-aware state estimation in the nominal case as well**. Note that the



System Component Faults	Nominal State Estimation	Attack Detection	Safety Detection
Accurate Sensor Model (No Sensor Faults)	Context-Aware Estimation (Chapter 3)	Attack Detection with no Sensor Faults (Chapter 6)	Safety Detection with no Sensor Faults (Chapters 4 and 5)
Sensor Faults Present	Fault Detection, Isolation and Reconfiguration	Attack Detection with Sensor Faults (Chapter 6)	Safety Detection with Sensor Faults (Chapters 4 and 5)

Figure 1.5: Overview of the problem space considered in this dissertation.

problems of resilient state estimation and attack prevention are not considered in this dissertation – they are orthogonal to our approach and can be applied in parallel with the techniques presented here (a review of the related work on both topics can be found in Chapter 2).

Given this setup, we investigate multiple problems in this problem space. In particular, as shown in Figure 1.5, we consider the three problems discussed above, namely context-aware state estimation, attack detection and safety detection; in addition, each problem is made more challenging by the existence of sensor faults, which adds a second dimension to the problem space. The first problem in the space is nominal state estimation when no attacks or faults exist – in this case the problem we address is context-aware state estimation. If no attacks exist but faults are introduced, then the system should perform a modified version of state estimation by implementing some of the established Fault Detection, Isolation and Reconfiguration (FDIR) techniques available in the literature [49, 80, 81, 103] (a thorough review of FDIR approaches is provided in Chapter 2).

If the system operates in an environment where it might be subjected to attacks, then it needs to perform attack detection. Once again, we distinguish between nominal attack detection where non-attacked sensors behave according to their models, on the one hand, and attack detection in the presence of sensor faults, on the other. Note that both detection problems are very general – we do not make any assumptions about the timing or class of sensor faults and attacks that the system might experience. As argued in Section 1.3 and as illustrated in Figure 1.6, the reason we can address problems of this generality is the fact that we focus on multi-sensor systems and leverage sensor redundancy in order to detect inconsistencies between sensor measurements (as also noted in Section 1.3, the multi-sensor framework requires a different assumption, namely that at least half of the sensors operate correctly).

Finally, as a last line of defense, we consider the problem of safety detection. Similar to the other problems, here we also distinguish between the nominal case where sensors behave according to their models and the case where sensors might experience faults as well. In order to analyze the system's safety, we employ sensor fusion techniques – sensor fusion exploits sensor redundancy and provides worst-case guarantees about the system's safety regardless of the way some sensors might fail or be attacked (again, assuming at least half of all sensors are correct). We investigate different ways of improving the output of sensor fusion such as incorporating measurement history, including context measurements as well as analyzing different schedules of transmitting sensor measurements in order to limit the attacker's information.

1.5.2 Problem Statements

To formalize the problems considered in this dissertation, we first summarize the system model components developed in Section 1.1 and make them more concrete as needed. We consider a system with a known state dynamics model:

$$x_{k+1} = f(x_k, u_k) + w_k, \tag{1.4}$$

where $x_k \in \mathbb{R}^d$ is the state, f models the state dynamics and w is noise that captures the fact that f cannot perfectly explain all possible system dynamics. Note that

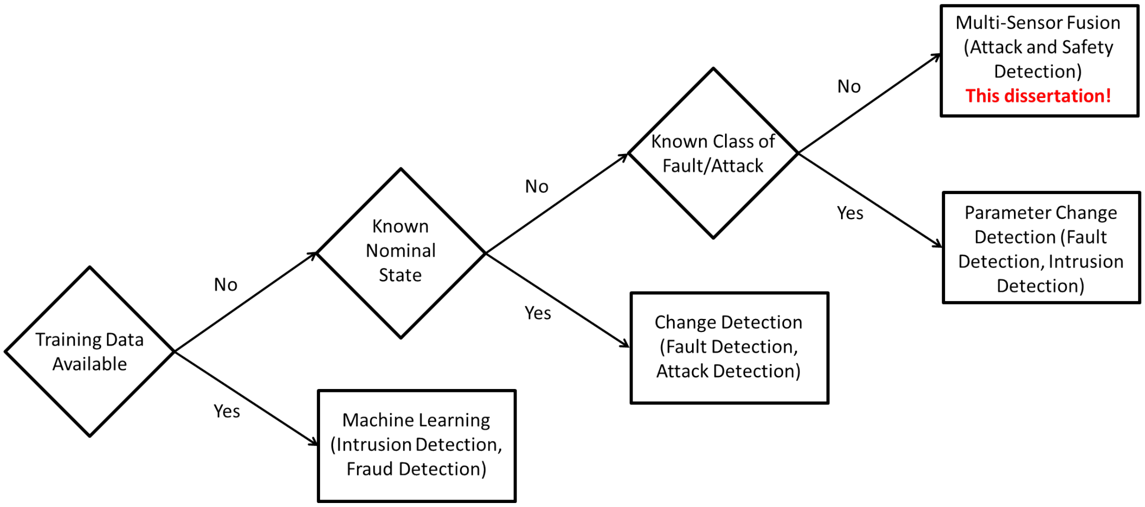

Training Data Available
No
Yes
Machine Learning (Intrusion Detection, Fraud Detection)
Known Nominal State
No
Yes
Change Detection (Fault Detection, Attack Detection)
Known Class of Fault/Attack
No
Yes
Multi-Sensor Fusion (Attack and Safety Detection)
This dissertation!
Parameter Change Detection (Fault Detection, Intrusion Detection)

Figure 1.6: High-level overview of the requirements of existing attack/fault detection approaches. All techniques that do not utilize sensor redundancy either need training data or make simplifying assumptions about the timing or class of attacks/faults.

depending on the specific problem that is considered, different assumptions will be made about f and w in the following chapters.

As described above, the system has access to two kinds of sensors available to it: plant (continuous) and context (binary). Each plant sensor is assumed to provide a measurement that is a linear function of the state:

$$y^c_{i,k} = C_{i,k} x_k + v_k, \tag{1.5}$$

where we denote the measurement by $y^c_{i,k} \in \mathbb{R}^m$, v_k is measurement noise, and matrix $C_{i,k}$ has appropriate dimensions.

Context sensors, on the other hand, do not measure the system's state but rather provide information about its context. We define context as a finite set $\mathcal{C} = \{c_1, \dots, c_N\}$, where each c_j is a context element that can be detected by a context sensor from certain states; example context elements include nearby buildings with known positions on a map or a vital sign exceeding a certain predefined threshold. For each j, a measurement $y^b_{j,k}$ is received that is 1 if c_j is detected and -1 otherwise. Thus, each context measurement $y^b_{j,k}$ can be modeled as a function h_j

of the state and the context element, i.e.,

$$y^b_{j,k} = h_j(x_k, c_j), \text{where } h_j(x_k, c_j) \in \{-1, 1\}. \tag{1.6}$$

Given this model, we may now ask questions of increasing difficulty and explore the problem space defined in Figure 1.5.

Nominal State Estimation

The first problem that we address is nominal context-aware estimation, i.e., we assume no faults or attacks are present and all sensors operate as modeled. Note that estimation is usually solved in a probabilistic setting as probabilities are a natural way of explaining measurement distributions and obtaining expected state estimates given the available measurements. Therefore, in the nominal setting we assume that both process and continuous measurement noises are random variables (Gaussian, in particular). In addition, we model context measurements in terms of the probability of obtaining a measurement given the current state, i.e.,

$$y^b_{j,k} = \begin{cases} 1 & w.p. \quad p_d(c_j \mid x) \\ -1 & w.p. \quad 1 - p_d(c_j \mid x), \end{cases} \tag{1.7}$$

where p_d is a function of the system state.

Problem 1 (Estimation)**.** *Given the system models in Equations* (1.4)-(1.7)*, with both process and continuous measurement noise following Gaussian distributions, the first problem we address is how to develop a state estimation algorithm that computes the exact probability distribution of the state given the measurements.*

Safety Detection and Attack Detection

Once we have developed algorithms for context-aware estimation in the nominal case, we can now ask the question of how to use sensor redundancy in order to perform safety detection as well as sensor attack detection/identification.

In order to formulate the problems, first note that safety and resilience are inherently worst-case concepts, i.e., they need to be established even in the worst case (e.g., it is not enough to claim that the system is safe 99% of the time). Thus, in this setting we adopt an abstract sensor framework (also known as a set membership framework [140]) instead of the probabilistic approach used in the nominal estimation problem. **Abstract sensors provide a set instead of a single (multidimensional) value** – this set is constructed around the actual measurement of the physical sensor and represents all possible values for the true state given the measurement. By keeping track of these sets over time, it is possible to draw conclusions about the system's safety and security even in the worst case.

Furthermore, note that in this framework we assume that all sensors measure the state directly. In other words, we abstract away the functional formulation in both (1.5) and (1.6) – instead, a set of possible values is derived using those functional formulations; the size and shape of this set depend on the particular sensor. This technique allows us to consider a truly redundant setting with multiple sensors "measuring" the same variable, using different processes and with different precision. These redundant measurements can then be used in a sensor fusion algorithm that outputs a fusion set that is guaranteed to contain the true state (assuming at least half of all sensors operate correctly). The system is considered safe if the fusion set does not contain any sets that are deemed unsafe.

With the above points in mind, we first state the safety detection problems. It is clear from the previous paragraph that the smaller the output of sensor fusion is, the stronger the guarantees about the system's safety are. Thus, we explore multiple ways of reducing the size of the fusion set by adding additional pieces of information and modifying the sensor fusion algorithm accordingly. In particular, first we solve the sensor fusion problem in a single time step, i.e., we only consider the measurements obtained at that step. As a first extension, we also make use of system dynamics in order to incorporate historical measurements and thus reduce the size of the fusion set. In the second extension, we note that in a shared bus (e.g., CAN bus in automotive CPS) setting, such as the one shown in Figure 1.1, each system component may observe all transmitted sensor measurements. In particular, this allows the attacker to observe correct sensor measurements before deciding what spoofed measurements to send on behalf of the corrupted sensors. Thus, we explore the effect of different measurement transmission schedules on the attacker's impact and what the best schedule from the system's point of view is.

Problem 2 (Safety Detection with no Sensor Faults)**.** *Given the system models in Equations* (1.4)-(1.6) *and assuming the abstract sensor framework where all (non-attacked) sensors are correct (and transmit their measurements over a shared bus), what is the smallest set that is guaranteed to contain the true state at a given time step? What is the smallest set if historical measurements are used as well? Which measurement transmission schedule minimizes the attacker's impact on the output of sensor fusion?*

Note that the above problems can be formulated both in the nominal case where sensors can only be attacked but are not faulty as well as in the case where sensors can be faulty. When we introduce sensor faults, however, the assumptions of nominal sensor fusion techniques (i.e., that at least half of all sensor measurements are correct in a given round) might be invalidated due to the fact that all sensors might provide faulty measurements at the same time. Thus, we investigate ways of incorporating system dynamics (and historical measurements) in order to still provide worst-case guarantees in the presence of both sensor attacks and faults.

Problem 3 (Safety Detection with Sensor Faults)**.** *Given the system models in Equations* (1.4)-(1.6) *and assuming the abstract sensor framework where (non-attacked) sensors might experience faults, can we find a bounded set that is guaranteed to contain the true state at any time step?*

Finally, we state the sensor attack detection problems in the abstract sensor framework as well. Once again, two problems can be formulated, one where sensors operate correctly if not attacked and one where sensors might be transiently faulty as well.

Problem 4 (Attack Detection with no Sensor Faults)**.** *Given the system models in Equations* (1.4)-(1.6) *and assuming the abstract sensor framework where all (non-attacked) sensors are correct, can we detect sensor attacks?*

Problem 5 (Attack Detection with Sensor Faults)**.** *Given the system models in Equations* (1.4)-(1.6) *and assuming the abstract sensor framework where (non-attacked) sensors might experience faults, can we detect sensor attacks? How do we distinguish attacks from faults and not raise false alarms due to transient faults?*

1.6 Contributions

At a high level, the goal of this dissertation is to develop detection and estimation techniques for analyzing the safety and security of CPS in the presence of sensor attacks and faults. We approach the problem by using the fact that modern CPS are equipped with multiple and diverse sensors measuring related system states. These sensors not only measure the plant's state but they also provide information about the system's environment; thus, the context measurements extracted from the environment sensors can be used as regular (discrete) measurements in addition to classical continuous measurements. Combining these sensors' data (both continuous and discrete) not only results in better state estimates but also enables us, by finding inconsistencies between measurements, to draw conclusions about the system's safety as well as about each sensor being faulty/attacked. The benefit of this approach is that we do not need to make any assumptions about how each individual sensor might fail or be attacked – instead we rely on redundant information to detect unsafe states.

In summary, this dissertation addresses three aspects of the problem space described above: 1) nominal state estimation, 2) safety detection and 3) attack detection. The specific contributions of each are listed below:

- **Nominal state estimation**. Since using discrete context measurements is a novel idea in itself, our first contribution is the development of nominal context-aware state estimation. We consider systems with access to both plant (continuous) and context (binary) measurements – we investigate two ways of modeling context measurements and derive closed-form Kalman-like filters for both cases. The theoretical properties of the resulting context-aware filters are analyzed as well. In addition, we illustrate the benefits of both filters via several case studies; first, we provide simulations of two robot localization scenarios with imperfect sensors that can be improved upon using context measurements. Finally, we present a real-data medical CPS case study where context measurements are used in order to improve the estimation of blood oxygen content, a critical vital sign that cannot be reliably measured during surgery.

- **Safety detection**. Using the abstract sensor framework, we employ sensor redundancy and sensor fusion techniques in order to detect when the system is unsafe even in the presence of sensor faults and attacks (without assuming anything about their timing or functional form). We develop a sensor fusion algorithm in order to find the smallest set that is guaranteed to contain the true state. We also explore several approaches to improve the output of sensor fusion, namely using historical measurements as well as selecting a sensor transmission schedule that minimizes the attacker's impact and available information. Finally, we extend the sensor fusion algorithm in order to handle context measurements as well, thereby further improving the algorithm's accuracy. Evaluations of these techniques are provided both in simulation and in experiments using an unmanned ground vehicle.

- **Attack detection**. Sensor attack detection and identification is used to improve the performance of safety detection by discarding attacked sensors. The proposed attack detection techniques are also grounded in the abstract multi-sensor setting. We note that such detectors often treat faults and attacks in the same way, thus raising unnecessary alarms due to transient faults. Consequently, we develop a framework for modeling transient faults and for distinguishing them from attacks – a corresponding attack detector in the presence of transient faults is presented as well. Both detectors are evaluated both in simulation and in experiments using an unmanned ground vehicle.

1.7 Outline of the Dissertation

The outline of this dissertation closely follows the highlighted problems in Figure 1.5. We first present in Chapter 2 a general overview of the related work on CPS safety and security, including nominal state estimation, anomaly detection as well as security approaches (both from a purely cyber and a combined cyber-physical point of view). Chapter 3 addresses the problem of nominal context-aware estimation; we develop two context-aware filters, analyze their theoretical properties and evaluate them, both in simulation and on real data. In Chapter 4, we consider the main problem of this dissertation, namely safety detection in the presence of sensor attacks and faults; we develop sensor fusion techniques and consider multiple ways of improving the output of sensor fusion by incorporating measurement history and by analyzing different schedules of measurement transmissions; all techniques are evaluated in simulation and in experiments with the LandShark robot. The guarantees provided by sensor fusion are further strengthened in Chapter 5 where we add another piece of information, namely context measurements; we develop a context-aware sensor fusion algorithm and evaluate it in simulation of a perfectly attackable system that can only detect it is unsafe with the addition of context measurements. Finally, Chapter 6 addresses the third major problem of this dissertation, namely sensor attack detection in the presence of transient faults; we develop a sound detection/identification algorithm and evaluate it on real data from the LandShark sensors. Concluding remarks and some avenues for future work are provided in Chapter 7.

Chapter 2

Related Work

This chapter reviews the related work in the general problem space defined in Chapter 1. We begin by presenting the most popular approaches to filtering and state estimation in general before discussing works in the area of fault detection, isolation and reconfiguration (FDIR) and robustness. The chapter concludes with security and attack detection. More specific topics such as sensor fusion, sensor selection and context-aware filtering are reviewed in their respective chapters.

2.1 State Estimation

State estimation is a very well studied problem in the control theory literature. Some of the first approaches were developed in the 1940s with the introduction of the Wiener filter and Wiener theory in general [219]. Consequently, the Wiener filter was extended and transformed into the classical Kalman filter [113], which is still the default choice for estimator in many applications due to its easy recursive implementation and intuitive appeal. Multiple extensions of the Kalman filter have been developed since then depending on the system and its assumptions, including the Gaussian Mixture filter for multimodal distributions [99, 193], the consensus Kalman filter for distributed systems [155, 156] and many others [14, 73, 76, 112].

While the Kalman filter is the best linear unbiased estimator for linear systems with Gaussian noise [26] (i.e., the function f is linear in (1.4) and the noises w in (1.4) and v in (1.5) have Gaussian distributions), it is challenging to develop algorithms with such strong properties in other settings, even in the linear-system framework. For example, in linear systems where the noise is distributed according to a truncated Gaussian distribution (e.g., in a turbofan engine model [188]) it is difficult to obtain a closed-form filter, and a popular approach is to estimate the posterior probability density function (pdf) using Monte Carlo techniques [69, 188]. Other, heavy tailed, noise distributions have been considered as well, such as a Student-t distribution [97], but they lead to noise disturbances that are not identically distributed, thus preventing researchers from obtaining closed-form estimates. Robust filters for general distributions have been developed as well by deriving an optimal time-varying smoothing boundary layer [84]. Alternatively, instead of a probabilistic approach, one may assume bounds on the noise. For example, one may derive a Kalman-like set membership filter given energy bounds on the noise [27]; if instead the noise is bounded by quadratic inequalities, then the estimation problem can be efficiently approximated by a convex optimization problem [74].

Unlike linear systems where a multitude of problems and setups have been addressed, estimation in general nonlinear systems is in still an open and challenging problem. One of the standard approaches to nonlinear estimation is the extended Kalman filter (EKF) [14], which works by linearizing the system dynamics and observation model and applying the standard Kalman filter to the linearized system. Another popular approach is the unscented Kalman filter (UKF) [112], which reconstructs the posterior mean and covariance matrix by propagating a minimal set of sample (sigma) points. Similarly, the smooth variable structure filter is an iterative algorithm that has also been shown to work well in smooth nonlinear systems [95]. All of these approaches work well in practice for mildly nonlinear systems with Gaussian-like noise but they do not perform as well in highly nonlinear systems. In such cases, one might use non-parametric approximation methods such as Gaussian Process filtering [99]. Yet another popular approach to nonlinear filtering is particle filtering [87], which employs Monte Carlo techniques in order to approximate nonlinear functions with arbitrary probability distributions; particle filters have been shown to work well in practice in robotics applications [62, 148, 202]. Particle filters have also been combined with Kalman filters in special problems where subsets of the state can be estimated in closed form [67, 90]. Finally, similar to the case of linear systems, nonlinear systems with bounded (not probabilistic noise) have been considered as well – e.g., set membership filters have been developed for systems with bounded noise and bounded derivatives of state dynamics and have been shown to outperform the EKF in highly nonlinear systems [139, 141].

A further complication to the estimation task is added by the fact that most system models are inaccurate or at least parameterized by multiple variables that differ across systems. System identification techniques can be utilized in such cases in order to obtain a model of the system based on observed data [17, 105, 129]. Approaches in this domain can be broadly classified as white-box (i.e., first principles) [151], gray-box [34, 121, 207], or black-box (i.e., data-driven) [111, 191, 197] depending on the assumptions on the underlying model. In addition, it is also possible to identify the model in an iterative fashion by gradually perturbing parameter values and minimizing a given cost function of the difference between predicted and measured states – this technique has been successfully applied in expectation maximization approaches [132, 186] as well as in multiple smoothing algorithms [116, 124, 176, 210].

To summarize, all nonlinear estimation techniques involve approximations, in addition to often being computationally expensive. A main reason for these disadvantages is the generality of the considered problems – most approaches attempt to develop estimators for all nonlinear systems or broad subsets thereof (e.g., all systems with differentiable dynamics). In contrast, in this dissertation we focus on a specific class of nonlinear measurements, i.e., binary measurements, and derive the exact posterior distributions in an efficient way.

2.2 Fault Detection, Isolation and Reconfiguration

The existing literature on FDIR is very broad and mature, including multiple exhaustive survey papers [49, 80, 81, 103]. FDIR methods work by developing a model of the system's operation and analyzing the residual between the expected and actual (as measured) behaviors [49]. There are two main approaches in this framework: in the first an initial condition is assumed such that the residuals can be computed at each time step as the system evolves [105]. In this setting, observers are developed that generate alarms when residuals are unlikely to have come from their nominal probability distributions or other expected behaviors [25, 52, 94, 130, 138, 221]. In the alternative approach, a certain type or direction of a fault is assumed such that fault detection can then be performed by detecting a change of system parameters using a generalized likelihood ratio test or a sequential probability ratio test [23, 24, 177, 178].

While the FDIR techniques work well in the presence of good models, many systems have complex models that may be difficult to derive. In such cases, system designers might develop simpler models and then apply techniques that guarantee the system is robust to modeling errors and faults [163]. Such approaches work by robustifying the employed estimator algorithms, e.g., by using the minimum covariance estimator [171], robust measure of data spread [57, 101] or robust principal component analysis [200].

2.3 Security and Attack Detection

The literature on security and attack detection can be broadly split into general computer security as well as techniques specifically tailored to the needs of CPS. Computer security is a widely researched area, with multiple established approaches and practices [15]. Some of the most notable branches of the area include intrusion detection systems [22, 131, 137], encrypted communication protocols [65, 168, 180], trusted computing [123, 205], security software verification [48, 68] and other software defenses [35, 56].

What is common between the standard computer security techniques is that they focus exclusively on cyber attacks. Modern CPS, on the other hand, are much more complex – a holistic analysis of the attack surface of CPS reveals that they are not only vulnerable to standard cyber threats, but are also subject to attacks exploiting their physical environment, thereby modifying sensor/actuator behavior and affecting the overall system model [42, 43, 44]. There are various approaches to securing CPS – the first class are those using fault-detection-like techniques such as developing state observers and change detectors [145, 147, 198]. In addition, attack-resilient state estimators have been introduced for systems with bounded noise [158, 159]. Researchers have also investigated ways to secure communication channels [10, 11, 12, 162] and more specifically the Control Area Network (CAN) bus [92, 96, 126, 127, 208], which is known to have multiple security vulnerabilities [98]. Finally, techniques specific to the cyber-physical coupling of CPS have been developed such as applying carefully chosen control inputs so as to expose the attacker [146], injecting and defending against false data injection attacks [128, 145, 147] or employing game theory analysis in order to model the attacker's behavior [9, 39, 91].

The common theme among the discussed approaches in both the fault detection and security domain is that certain assumptions are made about the occurrence of faults and attacks, i.e., either that they have a known functional form/direction or that the system begins in a known nominal state. Instead of making such assumptions, which might be unrealistic in modern CPS, we provide resilience and robustness through sensor redundancy and sensor fusion techniques [136] (described in more detail in the following chapters).

Chapter 3

Context-Aware Estimation

In this chapter we investigate the problem of context-aware state estimation, i.e., continuous state estimation using both continuous and discrete context-dependent measurements. As described in Chapter 1, context measurements are high-level representations of environment-sensor data that cannot be easily mapped to the state – by using the probability distribution of context measurements given the state, one may use them for estimation in a rigorous manner. The probabilistic formulation makes sense intuitively – for example, if a building is far from the robot and appears small in images, it might be recognized in just a few images; on the other hand, if the building is nearby, we expect to recognize it in most images, i.e., the probability of receiving a context measurement would be high for states close to the building.

In this dissertation we focus on binary measurements, i.e., each measurement takes on a value of 1 or -1. Binary measurements form an interesting subset of discrete measurements as they appear in a lot of CPS applications: 1) a medical device alarm that a vital sign exceeds a certain threshold (e.g., if the patient's oxygen saturation is above a certain threshold, then the overall oxygen content must be above a certain threshold [108]) as well as 2) occupancy grid mapping where a binary measurement is received as the robot gets close to an obstacle [196].

The concept of estimation with context-based measurements was originally explored in radar target tracking where measurements may also arrive irregularly and could take on discrete values; one notable technique developed in this domain is the probability hypothesis density (PHD) filter [134]. At the same time, the models considered in this domain are very general, which makes it challenging to derive exact theoretical results and instead leads to approximations that might be computationally expensive to obtain. Other general non-linear filtering methods have been developed as well, such as the hybrid density filter (HDF) [100], the set-membership filter [139], as well as the assumed density filter (ADF) [106] (the context-aware filter is actually a type of ADF for which we can compute the moments of the posterior distribution). Due to their generality, however, these filters do not provide strong theoretical guarantees about specific classes of non-linear systems.

Context measurements are also similar to quantized measurements in that they take on discrete values [83, 167]. At the same time, quantized measurements are different because they are derived from standard continuous measurements whereas context measurements are only related to the state through the probability of detection. System identification with binary measurements [213] has also been investigated although no approaches exist for the probabilistic setting considered in this work.

Context-aware filtering is also similar to Kalman filtering with intermittent observations [183, 190] and unreliable links [93, 104, 155, 179] in that measurements arrive irregularly; the frequency of measurement arrivals affects the filter's performance in these cases. Related to this is the concept of sensor scheduling where different sensors are used at different times so as to minimize interference or power consumption [110, 209, 220]. Yet another similar problem has been considered in the wireless sensor networks area where multiple sensors are deployed over a large area such that the receipt of each sensor's measurement could be considered a context measurement [75, 135].

Due to their discrete nature, context measurements can also be modeled with hybrid systems [102], where different modes contain different models of context measurements. Such models include Markov chain switching [55, 192], deterministic switching [66, 161] and other more general models [214]. However, due to their complexity, all of these approaches rely on approximations in order to perform the estimation task.

Different notions of context are also widely used in robotics for the purpose of localization and mapping [31] by using scene categorization [85] and object class information [13, 19]. However, these papers do not provide theoretical guarantees for their developed approaches. The work that is closest in its setup and assumptions to this dissertation addresses the problem of indoor localization by using both continuous and discrete measurements [19]; however, the particle filter that is used to combine the two types of measurements does not provide any theoretical guarantees for a finite set of particles and may suffer from particle deprivation problems in high-dimensional spaces. Finally, context-aware filtering could also be related to machine learning (e.g., Gaussian process classification [165]) in the sense that the objective is to learn a continuous probability distribution from discrete-valued data. In particular, the Expectation Propagation (EP) algorithm [143] is similar to the context-aware filter in that posteriors are approximated with Gaussian distributions as well – at the same time, no convergence results exist for EP.

In contrast with existing works, we develop a context-aware filter for linear systems with access to binary measurements – no knowledge is assumed about the measurements other than their probability of occurring given the state. In particular, we focus on two classes of functions that lead to (near) closed-form solutions and that represent a wide variety of detection scenarios observed in practice.

The first class of probability of detection functions are inverse-exponential functions. With this class of functions, the probability of detection is high when the state is close to a certain value (e.g., the robot is close to a building) and decreases rapidly as the state moves away. We show that this class of functions leads to a closed-form filter with Gaussian Mixtures without any approximations. The second class of functions are sigmoid functions defined as the probit function [152]. The probit function resembles a step function, i.e., for small inputs it is close to 0 but once a threshold is crossed, it increases rapidly and converges to 1. This class of functions capture the threshold medical alarms described above as well as threshold detection scenarios (e.g., occupancy grid mapping).

Similar to the inverse-exponential filter, we develop the probit-based context-aware filter by deriving the exact posterior distribution of the state given a context measurement. At the same time, it is not known how to compute the posterior for multiple context measurements since the integrals become intractable. As a result, we approximate the posterior distribution after the receipt of each context measurement with a Gaussian distribution with the same first two moments as the true posterior. The approximating Gaussian distribution is then used as a prior for the next measurement, thus obtaining a recursive context-aware filter.

In order to understand the asymptotic nature of the probit-based filter, we also analyze its theoretical properties. We first show that the posterior distribution is unimodal, so that the Gaussian approximation is indeed justified. In addition, we show that, for a scalar system, the expected variance of the filter's estimates is bounded provided that the probability of receiving both a measurement of 1 and -1 is at least some positive number η. This result is similar to a corresponding fact about Kalman filtering with intermittent observations [190] in the sense that the system needs to perform "useful" updates often enough in order to keep the uncertainty bounded. Generalizing this result to multidimensional systems, however, is challenging due to the fact that we aim to estimate continuous variables using discrete measurements only; at the same time, the intuition from the one-dimensional result could be used to prove a similar claim in the multidimensional case as well.

To provide further intuition about the probit-based filter's performance in the multidimensional case, we show convergence results about systems with no dynamics. In particular, we show that the eigenvalues of the filter's covariance matrix converge to 0 if and only if a persistence-of-excitation condition holds for the context measurements. This result is the context equivalent to an observability claim in a standard linear system – intuitively, it says that if there exist context measurements that observe all states, then the filter's uncertainty decreases over time. Furthermore, we show that as the eigenvalues of the covariance matrix converge to 0, the expressions for the moments of the Gaussian approximations converge to the Newton method [36], which suggests that the estimates themselves likely converge to the true state, since the posterior distribution is unimodal. This result provides a parallel with the widely used Expectation Propagation [143] algorithm where similar Gaussian approximations are employed at each step – thus, the results presented in this Chapter might be of interest to the machine learning community as well.

Finally, both context-aware filters are thoroughly evaluated in Sections 3.6, 3.7 and 3.8. We first provide simulations of two robot localization scenarios that illustrate real-world applications of context measurements. In addition, we provide additional simulations to illustrate the saw-shaped nature of the estimation curve induced by the probit-based filter as well as to illustrate a case in which the probit-based filter does converge for moving systems as well. Finally, we provide an application of the probit-based filter on real-patient data from the Children's Hospital of Philadelphia (CHOP) where the context-aware filter is used to estimate the patient's blood oxygen concentration.

3.1 System Model and Problem Formulation

This section formalizes the system model and states the estimation problem addressed by the context-aware filter. We consider a linear discrete-time system of the form

$$x_{k+1} = A_k x_k + \nu_k^p, \tag{3.1}$$

where $x_k \in \mathbb{R}^d$ is the system state, $x_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$, $\nu_k^p \sim \mathcal{N}(0, Q)$ is Gaussian process noise, and A_k is a matrix of appropriate dimensions.[1]

As described in Chapter 1, the system has two kinds of sensors available to it: plant (continuous) and context (binary). Plant sensors measure (subsets of) the state directly. The system has a linear observation model for plant sensors of the form

$$y_k^c = C_k x_k + \nu_k^m, \tag{3.2}$$

where we denote plant sensors' measurements by $y_k^c \in \mathbb{R}^m$, $\nu_k^m \sim \mathcal{N}(0, R)$ is Gaussian measurement noise, and matrix C_k has appropriate dimensions. Note that y_k^c is a big vector with all individual sensor measurements $y_{i,k}^c$ stacked on top of one another; v_k and C_k are obtained similarly.

Context sensors, on the other hand, do not measure the system's state but rather provide binary information about the system's context; example context measurements include detecting nearby objects with known positions on a map or a vital sign exceeding a certain predefined threshold. At each time k, a measurement y_k^b is received that is equal to 1 if a detection occurs and -1 otherwise.[2] We assume that y_k^b is equal to 1 with a known probability of detection given the state, denoted by $p_k^d(y_k^b \mid x)$, i.e.,

$$y_k^b = \begin{cases} 1 & w.p. \quad p_k^d(y_k^b \mid x_k) \\ -1 & w.p. \quad 1 - p_k^d(y_k^b \mid x_k), \end{cases} \tag{3.3}$$

where p_k^d is a function of the system state. As noted in the introduction of this chapter, p_k^d is close to 1 when the system is in a state that is highly correlated with receiving a context measurement (e.g., a robot is close to a building). Note that p_k^d is time-varying, i.e., different binary measurements may be received at different times.

[1] Note that we do not consider inputs u_k in order to simplify notation. All results presented in this chapter hold in the addition of inputs as well.

[2] Note that our framework can handle more than one binary measurement per time by repeated updates. We make the one-measurement assumption in order to simplify notation.

It is assumed that, conditioned on the state, context measurements are mutually independent.

Problem. *Given the system defined in* (3.1)-(3.3) *and a prior pdf*

$$p_{k|k}(x) = p(x \mid y^c_{0:k}, y^b_{0:k})$$

the goal is to compute the posterior density

$$p_{k+1|k+1}(x) := p(x \mid y^c_{0:k+1}, y^b_{0:k+1}),$$

describing the system's state given all available measurements and inputs.

3.2 Challenges with a Bayesian Approach

The problem formulation in Section 3.1 naturally lends itself to a recursive Bayesian approach with a predict and an update phase of the form

$$\begin{aligned} &\textbf{Predict:}\ p_{k+1|k}(x) = \int p_f(x \mid z) p_{k|k}(z) dz, \\ &\textbf{Update:}\ p_{k+1|k+1}(x) = \xi_{k+1} p_o(y^c_{k+1}, y^b_{k+1} \mid x) p_{k+1|k}(x), \end{aligned} \tag{3.4}$$

where $p_f(x_{k+1} \mid x_k, u_k)$ is the conditional pdf of the state at time $k+1$ given the state and input at time k, $p_o(y^c_{k+1}, y^b_{k+1} \mid x_{k+1})$ is the joint pdf of all available measurements (plant and context) given the state and ξ_{k+1} is a normalization constant [202].

While (3.4) provides a compact representation of the filtering problem, in general it is impossible to obtain a closed-form expression for the densities and the corresponding integrals. The notable exception is the linear Gaussian case which results in the Kalman filter, as noted in Chapter 2. However, the discrete measurements considered in this paper do not lead to clean analytic derivations such as the one in the Kalman filter. In such a case it might be possible to use some of the nonlinear estimation techniques described in Chapter 2 such as the ADF, PHD filter, HDF and others. At the same time, as argued in Chapter 2, all these approaches involve approximations and have variable performance on real problems. Therefore, in this thesis we focus on two specific probability of context detection functions (i.e., $p_k^d(y_k^b \mid x)$ in (3.3)) that lead to closed-form filters. We argue that each of these functions captures a sufficiently large class of scenarios so as to be useful in a lot of modern systems. The following two sections present these functions, both formally and intuitively, before deriving the resulting filters.

3.3 Context-Aware Filter with Inverse-Exponential Functions

The first class of probability of context detection functions considered in this dissertation are inverse-exponential functions.

Assumption. *Suppose the probability of context detection functions are inverse-exponential functions that are defined as follows:*

$$p_k^d(y_k^b \mid x_k) = e^{-\frac{1}{2}(G_k x_k - \theta_k)^T V_k^{-1}(G_k x_k - \theta_k)}, \tag{3.5}$$

which are parameterized by $\theta_k \in \mathbb{R}^q$ and $V_k \in \mathbb{R}^{q \times q}$, and $G_k \in \mathbb{R}^{q \times d}$, which can be thought of as a selection matrix when $q < d$. This probability is 1 when $G_k x_k = \theta_k$ and approaches 0 when $G_k x_k - \theta_k$ gets large. Note that G_k, θ_k and V_k are possibly time-varying, i.e., a different context measurement could be received at each time step.

We argue that inverse-exponential functions capture a wide class of context measurements observed in reality. In particular, they are designed so that the probability of detecting a context element is large when the system is in the vicinity of that element and is small otherwise. For example, in the case of detecting a nearby building using image processing, the probability of getting a detection is very high when the camera is close to the building but starts decreasing rapidly as the system moves away [19]. As another example, consider a vehicle trying to localize by detecting frequency modulation (FM) radio signals – since certain FM signals can only be detected in certain regions, receiving such a known signal may help the vehicle improve its localization estimate. Inverse-exponential functions can be used to model this scenario as well, since wireless signals are also known to greatly decay as the receiver gets far away from the transmitter [86].

Having fixed (3.5) as the probability of context detection, with θ_k, G_k and V_k known at each time step (or potentially learned from data), we now derive the resulting context-aware filter. Note that, due to the shape of the function in (3.5) (i.e., it resembles a non-normalized Gaussian pdf), incorporating the context measurements in the filter results in a Gaussian Mixture (GM) distribution. A GM is a distribution whose pdf is defined as a weighted sum of Gaussian pdfs:

$$g_{GM}(x) = \sum_{i=1}^{M} w_i \phi(x; \mu_i, \Sigma_i), \tag{3.6}$$

where $\phi(x; \mu_i, \Sigma_i)$ is the pdf of a Gaussian distribution with mean μ_i and covariance matrix Σ_i, and w_i are weights such that $\sum_{i=1}^{M} w_i = 1$. GMs have two properties that make them suitable for modeling multimodal distributions. First of all, they are linear combinations of Gaussian pdfs; thus, a recursive filter using a GM can be developed with a bank of Kalman filters, one for each element in the GM. In particular, this means that the context-aware filter developed in this chapter has a closed-form solution for GMs, i.e., if the prior is a GM (including a single Gaussian distribution, which is a special case of a GM), then so is the posterior. The second useful property of GMs is that, given a sufficient number of elements, a GM can be used to approximate any continuous pdf [99]. For these reasons, GMs have been extensively studied and appear in a lot of the popular nonlinear filters.

We now present the main result of this section, namely the context-aware filter with inverse-exponential functions.

Proposition 1. *Consider a system with linear dynamics*

$$x_{k+1} = A_k x_k + \nu_k^p,$$

linear state observation model

$$y_k^c = C_k x_k + \nu_k^m,$$

and context observation model of the form

$$p_k^d(y_k^b \mid x_k) = e^{-\frac{1}{2}(G_k x_k - \theta_k)^T V_k^{-1}(G_k x_k - \theta_k)}.$$

Assuming that the state prior $p_{k|k}$ is a Gaussian Mixture, then the predicted and updated pdf's, $p_{k+1|k}$ and $p_{k+1|k+1}$ respectively, are also Gaussian Mixtures ***without any approximation****.*

Proof. Note that, unlike the conventional Kalman filter that has a predict and an update stage, the proposed filter has three steps: prediction, continuous update and discrete update. There is also an optional mixture reduction step discussed at the end of the section.

3.3.1 Predict

For the predict stage, we note that

$$p_{k+1|k}(x) = \sum_{i=1}^{M} w_i \int \phi(x; A_k z, Q)\phi(z; \mu_i, \Sigma_i)dz$$

$$
\begin{aligned}
&= \sum_{i=1}^{M} w_i \phi(x; A_k \mu_i, A_k \Sigma_i A_k^T + Q) \\
&= \sum_{i=1}^{M} w_i \phi(x; \mu_i^p, \Sigma_i^p),
\end{aligned}
$$

which is the usual form of the Kalman filter predict equations (e.g., see [113]). The resulting distribution is again a GM.

3.3.2 Continuous Update

As described above, we perform the update separately for state (continuous) and context (discrete) sensors. Upon receiving a measurement y_{k+1}^c, the continuous update is:

$$
\begin{aligned}
p_{k+1|k+1}^c(x) &= \frac{p(y_{k+1}^c \mid x) p_{k+1|k}(x)}{\int p(y_{k+1}^c \mid z) p_{k+1|k}(z) dz} \\
&= \frac{\phi(y_{k+1}^c; C_k x, R) \sum_{i=1}^{M} w_i \phi(x; \mu_i^p, \Sigma_i^p)}{\int \phi(y_{k+1}^c; C_k z, R) \sum_{j=1}^{M} w_j \phi(z; \mu_j^p, \Sigma_j^p) dz} \\
&= \sum_{i=1}^{M} \left(\frac{w_i \gamma_i^c}{\alpha^c} \right) \frac{\phi(y_{k+1}^c; C_k x, R) \phi(x; \mu_i^p, \Sigma_i^p)}{\int \phi(y_{k+1}^c; C_k z, R) \phi(z; \mu_i^p, \Sigma_i^p) dz} \\
&= \sum_{i=1}^{M} w_i^c \phi(x; \mu_i^c, \Sigma_i^c),
\end{aligned}
$$

where

$$
\begin{aligned}
\alpha^c &:= \sum_{i=1}^{M} w_i \gamma_i^c \\
\gamma_i^c &:= \int \phi(y_{k+1}^c; C_k z, R) \phi(z; \mu_i^p, \Sigma_i^p) dz \\
&= \phi(y_{k+1}^c; C_k \mu_i^p, C_k \Sigma_i^p C_k^T + R) \\
\mu_i^c &:= \mu_i^p + K_i^c (y_{k+1}^c - C_k \mu_i^p) \\
\Sigma_i^c &:= (I - K_i^c C_k) \Sigma_i^p
\end{aligned}
$$

$$K_i^c := \Sigma_i^p C_k^T (C_k \Sigma_i^p C_k^T + R)^{-1}.$$

Note that the posterior distribution is also a GM with the same number of elements but with possibly rescaled weights.

3.3.3 Discrete Update

For the discrete update, first note that the posterior distribution depends on whether y_{k+1}^b is -1 or 1 as the probabilities of getting either one are different. Consider first the case when $y_{k+1}^b = 1$:

$$\begin{aligned}
p_{k+1|k+1}(x) &= \frac{p(y_{k+1}^b = 1 \mid x) p_{k+1|k+1}^c(x)}{\int p(y_{k+1}^b = 1 \mid z) p_{k+1|k+1}^c(z) dz} \\
&= \frac{\phi(\theta_k; G_k x, V_k) \sum_{i=1}^M w_i^c \phi(x; \mu_i^c, \Sigma_i^c)}{\int \phi(\theta; G_k z, V_k) \sum_{j=1}^M w_j^c \phi(z; \mu_j^c, \Sigma_j^c) dz} \\
&= \sum_{i=1}^M \left(\frac{w_i^c \gamma_i^d}{\alpha^d} \right) \frac{\phi(\theta_k; G_k x, V_k) \phi(x; \mu_i^c, \Sigma_i^c)}{\int \phi(\theta_k; G_k z, V_k) \phi(z; \mu_i^c, \Sigma_i^c) dz} \\
&= \sum_{i=1}^M \left(\frac{w_i^c \gamma_i^d}{\alpha^d} \right) \phi(x; \mu_i^d, \Sigma_i^d),
\end{aligned}$$

where α^d, γ_i^d, μ_i^d, Σ_i^d and K_i^d are defined similar to their continuous analogues.

Finally, when $y_{k+1}^b = -1$, the update becomes

$$\begin{aligned}
p_{k+1|k+1}(x) &= \frac{\big(1 - p(y_{k+1}^b = 1 \mid x)\big) p_{k+1|k+1}^c(x)}{\int \big(1 - p(y_{k+1}^b = 1 \mid z)\big) p_{k+1|k+1}^c(z) dz} \\
&= \sum_{i=1}^M \frac{w_i^c \big(1 - p(y_{k+1}^b = 1 \mid x)\big) \phi(x; \mu_i^c, \Sigma_i^c)}{1 - \sum_{j=1}^M w_j^c \int p(y_{k+1}^b = 1 \mid z) \phi(z; \mu_i^c, \Sigma_i^c) dz} \\
&= \sum_{i=1}^M \frac{w_i^c}{1 - \sum_{j=1}^M w_j^c \beta_j} \phi(x; \mu_i^c, \Sigma_i^c) \; + \frac{-w_i^c \beta_i}{1 - \sum_{j=1}^M w_j^c \beta_j} \phi(x; \mu_i^d, \Sigma_i^d)
\end{aligned}$$

where $\beta_i := \gamma_i^d \sqrt{(2\pi)^q \det(V_k)}$. □

Thus, we have inductively shown that for the probability of detection function considered in this paper, the localization filter can be computed in closed form and results in a GM distribution of the posterior. Note that the number of elements in the GM doubles every time $y_k^b = -1$, thus an additional step may be necessary in order to bound the number of elements.

3.3.4 Mixture Reduction

The proof of Proposition 1 provides an exact form for the posterior. However, the number of elements in the GM doubles every time a measurement of -1 is received; hence, this number may increase exponentially over time. Many approaches for reducing the number of elements have been proposed in the literature, ranging from keeping the elements with highest weights to merging or discarding elements based on certain notions of distance between them [193]. Note that most available techniques assume weights are positive, yet the GM developed in this paper may have negative weights as well. In such cases, one may use a Gibbs Sampler [195] in order to reduce the size of the GM. A Gibbs Sampler draws random samples from the distribution and can approximate it with a GM with a desired number of elements. Note that in order to sample from a distribution with negative weights such as the one developed in this paper, accept-reject sampling may be utilized [150].

3.4 Context-Aware Filter with Sigmoid Functions

Although inverse-exponential functions capture a wide variety of context measurements that occur in practice, there are other interesting scenarios that cannot be explained with this class of functions. In particular, a major limitation of inverse-exponential functions is that they are symmetric around their θ parameter; thus, they would not be well suited for modeling inherently non-symmetric context measurements such as a vital sign crossing a predefined threshold (e.g., the blood-oxygen saturation is less than 90%). Similarly, inverse-exponential functions cannot be used to model a scenario in which a building can only be detected from certain angles (e.g., because of occlusions).

In order to overcome these limitations, in this section we investigate a second broad class of context detection functions, namely sigmoid functions.

Assumption. *Suppose the probability of context detection functions are sigmoid functions that are defined as the probit logistic function [153]:*

$$p_k^d(y_k^b \mid x_k) = \Phi(y_k^b(b_k^T x_k + a_k)), \tag{3.7}$$

where Φ is the cumulative distribution function of the standard Normal distribution, $b_k \in \mathbb{R}^d$ is a vector of known weight parameters, and $a_k \in \mathbb{R}$ is a known parameter offset. Note that $p_k^d(y_k^b = 1 \mid x_k) = 1 - p_k^d(y_k^b = -1 \mid x_k)$ due to the rotational symmetry of Φ, i.e., $\Phi(-x) = 1 - \Phi(x)$. We assume there is a finite set of size C of context weights and offsets $\mathcal{V} = \{(b^1, a^1), \ldots, (b^C, a^C)\}$.

Note that the inner function in (3.7) defines a hyperplane, determined by the values of a_k and b_k, that can be intuitively considered as the detection threshold, i.e., the probability of getting a detection is very low when the state x_k is below the "threshold" and increases rapidly as x_k crosses the "threshold". To explain the name of this class of context detection functions, note that in the one-dimensional case, this function greatly resembles the classical sigmoid function: $f(x) = 1/(1 + e^{-x})$, which also exhibits this pattern of values close to 0 as x approaches $-\infty$ and close to 1 for large x, with a very quick transition period in between. Due to this step-like shape, sigmoid functions are well suited for modeling the scenarios presented above – it is expected that once a signal exceeds a certain threshold, even inaccurate sensors will be able to detect the event and raise an alarm.

Assumption. *In this section, to simplify notation we assume the system has access to context measurements only (but not continuous plant measurements). All results* *presented in this section hold in the presence of continuous (linear) measurements as well.*

Developing an exact filter incorporating probit-based measurements is not straightforward, however, due to the fact that the posterior distribution, once context measurements have been received, is not the same as the prior (even if the prior is Gaussian). At the same, as argue below, a Gaussian distribution with the same mean and covariance matrix is a good approximation for the posterior distribution.

We now present the phases of the sigmoid-based filter, in a similar fashion to the GM-based one (excluding the continuous update). In this case we assume the prior $p_{k-1|k-1}$, at time $k \geq 1$, is a **single Gaussian distribution with mean** $\mu_{k-1|k-1}$ **and covariance matrix** $\Sigma_{k-1|k-1}$.

3.4.1 Predict

The predict phase is the classical Kalman filter prediction:

$$\begin{aligned} p_{k|k-1}(x) &= \int \phi(x; A_{k-1}z, Q)\phi(z; \mu_{k-1|k-1}, \Sigma_{k-1|k-1})dz \\ &= \phi(x; A_{k-1}\mu_{k-1|k-1}, A_{k-1}\Sigma_{k-1|k-1}A_{k-1}^T + Q) \\ &= \phi(x; \mu_{k|k-1}, \Sigma_{k|k-1}), \end{aligned}$$

where $\phi(x; \mu, \Sigma)$ denotes the pdf of a Gaussian distribution with mean μ and covariance matrix Σ.

3.4.2 Update

The posterior distribution after the receipt of a binary measurement y_k^b is shown in Proposition 2 below (all proofs are given in the Appendix).

Proposition 2. *Upon receipt of a discrete measurement $y_k^b \in \{-1, 1\}$, the discrete* *update is as follows:*

$$p_{k|k}(x) = \frac{\Phi(y_k^b(b_k^T x + a_k))\phi(x; \mu_{k|k-1}, \Sigma_{k|k-1})}{Z_k}, \tag{3.8}$$

where

$$Z_k = \Phi\left(\frac{y_k^b(b_k^T \mu_{k|k-1} + a_k)}{\sqrt{b_k^T \Sigma_{k|k-1} b_k + 1}}\right).$$

Approximation. *We approximate the posterior distribution in* (3.8) *with a Gaussian distribution with the same mean and covariance matrix.*

Note that the posterior distribution after incorporating context measurements is no longer Gaussian. However, a Gaussian still seems to be a good approximation for (3.8). In particular, as shown in Proposition 3 below, the distribution in (3.8) is log-concave; log-concavity, in turn, implies unimodality, as discussed in Corollaries 1 and 2. Thus, we approximate the posterior in (3.8) with a Gaussian with the same mean and covariance matrix as the distribution in (3.8) – these quantities are computed in Proposition 4 below.

Proposition 3. *The distribution in* (3.8) *is log-concave, i.e., the function*

$$g(x) = \ln(p_{k|k}(x)) \tag{3.9}$$

is concave.

Corollary 1 ([64])**.** *In one dimension, the distribution in* (3.8) *is* ***unimodal****, i.e., there exists a point x^* such that $p_{k|k}(x)$ is increasing for $x \leq x^*$ and $p_{k|k}(x)$ is decreasing for $x \geq x^*$.*

Corollary 2 ([64])**.** *In many dimensions, the distribution in* (3.8) *is* ***star-unimodal*** *(a random variable $X \in \mathbb{R}^n$ is said to have a star-unimodal distribution if for ev-*

ery bounded non-negative Borel measurable function f on $\mathbb{R}^n$, $t^n\mathbb{E}[f(tX)]$ is non-decreasing for $t \in [0, \infty)$).[3]

Proposition 4. *The mean of the distribution in* (3.8) *is:*

$$\mu_{k|k} = \mu_{k|k-1} + \Sigma_{k|k-1} b_k (b_k^T \Sigma_{k|k-1} b_k + \chi_k)^{-1} y_k^b, \tag{3.10}$$

where

$$\chi_k = \frac{\sqrt{b_k^T \Sigma_{k|k-1} b_k + 1} - b_k^T \Sigma_{k|k-1} b_k \alpha(M_k)}{\alpha(M_k)} \tag{3.11}$$

$$\alpha(x) = \phi(x; 0, 1)/\Phi(x) \tag{3.12}$$

$$M_k = \frac{y_k^b (b_k^T \mu_{k|k-1} + a_k)}{\sqrt{b_k^T \Sigma_{k|k-1} b_k + 1}}. \tag{3.13}$$

The covariance matrix of the distribution in (3.8) *is:*

$$\Sigma_{k|k} = \Sigma_{k|k-1} - \Sigma_{k|k-1} b_k (b_k^T \Sigma_{k|k-1} b_k + \gamma_k)^{-1} b_k^T \Sigma_{k|k-1} \tag{3.14}$$

where

$$\gamma_k = \frac{(1 - h(M_k))\, b_k^T \Sigma_{k|k-1} b_k + 1}{h(M_k)} \tag{3.15}$$

$$h(x) = \alpha(x)(x + \alpha(x)). \tag{3.16}$$

Remark. *Note that the context-aware filter is similar to Kalman filtering with intermittent observations [190] in that measurements arrive in a stochastic manner. Thus* (3.14) *resembles a standard Riccati equation (update), where the non-linear term γ_k could be considered as the equivalent of measurement noise.*

Note also that the functions α and h defined in (3.12) and (3.16), respectively,

[3] Note that while there is a standard definition of unimodality in one dimension, many definitions exist in multiple dimensions (consult [64] for an exhaustive discussion).

have been studied extensively in the statistics community. The ratio α is known as the inverse Mills ratio; some properties of the inverse Mills ratio that are used throughout this dissertation are summarized below.

Definition. *The inverse Mills ratio is defined as the ratio of the pdf and cdf of a standard Normal distribution, respectively, i.e.,*

$$\alpha(x) = \phi(x; 0, 1)/\Phi(x).$$

Proposition 5 ([173])**.** *The following facts are true about the inverse Mills ratio:*

1. $h(x) := -\alpha'(x) = \alpha(x)(x + \alpha(x))$
2. $0 < h(x) < 1, \forall x \in \mathbb{R}$
3. $h'(x) < 0, \forall x \in \mathbb{R}$.

Remark. *Since* $0 < h(x) < 1$*, we can conclude that* $\gamma_k > 1$*.*

3.5 Convergence Properties of the Sigmoid-Based Context-Aware Filter

In this section we analyze the convergence properties of the sigmoid-based context-aware filter. Due to the fact that the task is to estimate a continuous variable using only discrete measurements, proving convergence is hard in general, especially given the random and time-varying nature of the filter. Ideally, one could hope to prove that the expected covariance matrix is bounded under some conditions on the initial covariance matrix and the probability of measurement arrivals (i.e., similar to the result for Kalman filtering with intermittent observations [190]). However, note that there is an extra non-linear γ_k term in the Riccati equation for the covariance matrix update in (3.14). The presence of γ_k makes it challenging to analyze the system when dynamics are also considered since γ_k cannot be upper-bounded in general (as shown in Proposition 5, the function h can be arbitrarily close to 0). Such an upper bound can be derived in the special case of a scalar system as shown in the next subsection.

To provide further intuition about the filter's convergence, we also show results for a non-moving system. In particular, in Subsections 3.5.2 and 3.5.3 we provide an observability-like claim for the filter, i.e., the eigenvalues of the covariance matrix converge to 0 if and only if a persistence-of-excitation condition is true for the weight vectors v_k over time. Furthermore, we show that, as the eigenvalues of the covariance matrix converge to 0, the discrete update of the filter converges to a Newton Method step, which is an intuitive result given that the filter approximation matches the first two moments of the true posterior distribution.

3.5.1 Bounded Variance for a Scalar System

In this section we analyze conditions that result in a bounded variance of the context-aware filter given a scalar system:

$$x_{k+1} = ax_k + \nu_k^p, \tag{3.17}$$

where $x_k, a \in \mathbb{R}$, and $\nu_k^p \sim \mathcal{N}(0, q)$.

First note that the update in (3.14) looks like a standard Riccati equation, except for the non-linear term γ_k. Thus, one way to show that the context-aware filter's variance is bounded is by providing an upper bound on γ_k such that (3.14) is bounded (with some positive probability) by a standard Riccati equation. In such a case, our problem can be reduced to Kalman filtering with intermittent observations [190], and we can use some of the known facts for that scenario.

One case in which γ_k can be bounded (with positive probability) is when the probability of receiving both a measurement of 1 or -1 is at least some positive number η. In such a case, γ_k can be upper-bounded (with probability at least η) by

$((1-h(0))b_k\sigma_k b_k+1)/h(0)$ by using the properties of h, i.e., $h'(x) < 0$ for all x. This condition leads to the following result, similar to a result from Kalman filtering with intermittent observations.

Theorem 1. *Consider the system in* (3.17) *and suppose that, for all* x_k, $p_k^d(y_k^b \mid x_k) \geq \eta$ *for* $y_k^b = \pm 1$. *Then there exists some* $\eta_c \in [0, 1)$ *such that*

$$\forall \sigma_0, \mathbb{E}[\sigma_k] \leq M_{\sigma_0}, \textit{for } \eta_c < \eta \leq 1,$$

where M_{σ_0} *is a constant that depends on the initial condition.*

Theorem 1 says that the filter's expected uncertainty is bounded at all times if the probability of receiving "useful" context measurements is sufficiently high (by "useful" we mean that a measurement can be both 1 or -1 with probability at least η such that receiving the measurement does provide significant information). This result makes sense intuitively – if the system is moving away from all available context measurements (i.e., if $b^T x + a$ is very large in absolute value for all $(b, a) \in \mathcal{V}$), we cannot expect to be able to estimate the state; conversely, if context measurements are available throughout the system's execution, then the filter's uncertainty should be low.

Note that the proof of Theorem 1 does not generalize immediately to the multidimensional case, as the bound on γ_k in the multidimensional case does not lead to a standard-Riccati-equation bound on the expected covariance matrix. At the same time, we believe the same intuition could be used to obtain a similar result for the multidimensional case as well.

3.5.2 Covariance Matrix Convergence for Non-Moving System

While we cannot bound the filter's expected uncertainty in the multidimensional case, we provide such a result in the special case of a non-moving system. In particular, we show that for a system with no dynamics, the eigenvalues of the covariance matrix converge to 0 if and only if a persistence-of-excitation condition (formalized below) is true for the weight vectors b_k over time. To simplify notation and since no dynamics predictions are performed in this section, we drop the prediction notation in the rest of this section (i.e., we write Σ_k instead of $\Sigma_{k|k} = \Sigma_{k+1|k}$).

Before presenting the main result of this subsection, we first describe the behavior of the covariance matrix after multiple binary updates, as presented in the following lemma.

Lemma 1. *After applying N updates at time k, the covariance matrix update from* (3.14) *can be written as:*

$$\Sigma_{k+N} = \Sigma_k - \Sigma_k B_k^T (B_k \Sigma_k B_k^T + \Gamma_k)^{-1} B_k \Sigma_k, \tag{3.18}$$

where $B_k = [b_{k+1}, \ldots, b_{k+N}]^T$, $[\Gamma_k]_{(i,j)} = \gamma_{k+i}$ if $i = j$ and $[\Gamma_k]_{(i,j)} = 0$ otherwise.

The update in Lemma 1 looks similar to a standard Riccati equation (without the dynamics elements). Thus, it is not surprising that convergence of the covariance matrix depends on similar conditions on the matrix B_k as for a C_k matrix in a standard linear system. One such property is the widely used persistence of excitation [89].

Definition (Persistence of Excitation)**.** *The sequence of context weights and offsets, (b_k, a_k), is said to be* ***persistently exciting*** *if there exist d linearly independent weight vectors with corresponding offsets $\mathcal{P} = \{(b^1, a^1), \ldots, (b^d, a^d)\}$ that appear infinitely often, i.e., for every k, there exists $l_k \in \mathbb{N}$ such that*

$$\forall (b^i, a^i) \in \mathcal{P}, \exists t \in \{k, \ldots, k + l_k\} \text{ s.t. } (b_t, a_t) = (b^i, a^i).$$

Persistence of excitation is a standard assumption in estimation and system identification [89].[4] Intuitively, it means that there exists a set of context measurements that are received infinitely often such that their corresponding weights span $\mathbb{R}^d$.[5] The offsets are also important because even if the same weights repeat over time, the change of offsets might still affect the probability of receiving new context measurements.

Theorem 2. *Suppose the system has no dynamics (i.e., $A_k = I$, the identity matrix, and $Q = 0$). Let $\lambda_k^j > 0$ be the eigenvalues of Σ_k. Then $\lambda_k^j \xrightarrow{p} 0$ as $k \to \infty$ if and only if (b_k, a_k) is persistently exciting.*

Theorem 2 is essentially an observability result. It suggests that if some states are not observed through binary measurements, then the uncertainty about those states does not decrease over time. On the other hand, if all states are observed, then the uncertainty is reduced over time in a manner similar to the standard Kalman filter with a persistently exciting C_k matrix.

At the same time, even if the covariance matrix converges to zero, it is not clear whether the mean of the estimator converges to the true state. However, as shown in Section 3.7, simulations suggest that the estimates do converge to the true state. Furthermore, similar convergence results exist for the Expectation Propagation (EP) algorithm (which also approximates the posterior distribution with a Gaussian with the same moments), namely 1) EP converges to the true state for strongly log-concave observation models [60] (the probit model is log-concave but is not strongly log-concave) and 2) in the limit, EP has a fixed point at the true state if the observation model has bounded derivatives [61] (true for the probit model). Thus, it is likely that the context-aware filter's mean also converges to the true state but we leave proving this result for future work.

[4] The definition used in our work is a special case of standard definitions since we have a finite set of context weights.

[5] Note that persistence of excitation does not require the received context measurements to take on a specific value, i.e., they can be either -1 or 1. Intuitively, the definition only requires the same classifiers to run infinitely often.

3.5.3 Convergence of "Site" Approximations

In an effort to better understand the asymptotic behavior of the sigmoid-based context-aware filter for systems with no dynamics, in this subsection we analyze the effect of a single update in the limit. In particular, we show that as more data is available, each discrete update resembles a Newton Method step (this result is similar to a recent result about the limit behavior of EP [61]).

Definition. *The Newton Method for finding the minimum of a twice-differentiable function f is computed as follows: given the previous iteration point x_n, the next step is [36]:*

$$x_{n+1} = x_n - \left[f^{''}(x_n)\right]^{-1} f^{'}(x_n).$$

The significance of this property is that the Newton Method converges to the optimal value (i.e., the peak of the distribution) of concave or quasi-concave functions. Since the posterior distribution considered in this work is log-concave (i.e., quasi-concave), there is strong evidence to believe that the context-aware filter with the probit observation model does indeed converge to the true state.

Before presenting the result, we first note that each update of the context-aware filter could be viewed as a Gaussian approximation of the observation model itself (i.e., of the probit model). More specifically, the posterior Gaussian approximation could be considered as a Gaussian distribution that resulted from an update in which the observation model was also a Gaussian distribution with the appropriate parameters (also known as a "site" approximation in machine learning).

Definition (Site Approximation)**.** *Given a Gaussian prior $\phi(x; \mu_{k-1}, \Sigma_{k-1})$ and a binary update with observation model $\Phi(y_k^b(b_k^T x + a_k))$, a site approximation is a Gaussian distribution $p^s(x) := \phi(x; \mu^s, \Sigma^s)$ such that the distribution (normalized by* *the constant β)*

$$p^G(x) = \beta \phi(x; \mu_{k-1}, \Sigma_{k-1}) \phi(x; \mu^s, \Sigma^s)$$

has the same mean and covariance matrix as the true posterior

$$p_{k|k}(x) = \frac{1}{Z_k} \Phi(y_k^b(b_k^T x + a_k)) \phi(x; \mu_{k-1}, \Sigma_{k-1}).$$

Site approximations are easily computed when we consider the natural parameters of the distribution. Suppose the prior distribution is $\phi(x; \Omega_{k-1}^{-1}\omega_{k-1}, \Omega_{k-1}^{-1})$, where $\Omega_{k-1} = \Sigma_{k-1}^{-1}$ and $\omega_{k-1} = \Omega_{k-1}\mu_{k-1}$ are the prior's information matrix and mean, respectively. Similarly, suppose the posterior Gaussian approximation is $\phi(x; \Omega_k^{-1}\omega_k, \Omega_k^{-1})$. Then the parameters of the site approximation $\phi(x; (\Omega_k^s)^{-1}\omega_k^s, (\Omega_k^s)^{-1})$ are computed as follows [18]:

$$\Omega_k^s = \Omega_k - \Omega_{k-1} \tag{3.19}$$

$$\omega_k^s = \omega_k - \omega_{k-1}. \tag{3.20}$$

The site approximation abstraction is useful as it allows us to reason about the "contribution" of each update. In particular, we can derive the following result.

Theorem 3. *Suppose the prior is $\phi(x; \Omega_k^{-1}\omega_k, \Omega_k^{-1})$ (where $\Omega_k = \Sigma_k^{-1}$ and $\omega_k = \Omega_k \mu_k$). After performing an update in the context-aware filter, the natural parameters of the site approximation are:*

$$\Omega_{k+1}^s = b_{k+1}\gamma_{k+1}^{-1} b_k^T \tag{3.21}$$

$$\omega_{k+1}^s = \Omega_{k+1}^s \mu_k + (I + L_{k+1}) b_{k+1} N_{k+1}^{-1} y_{k+1}^b, \tag{3.22}$$

where

$$N_{k+1} = b_{k+1}^T \Sigma_k b_{k+1} + \chi_{k+1}$$
$$L_{k+1} = b_{k+1} \gamma_{k+1}^{-1} b_{k+1}^T \Sigma_k.$$

Corollary 3. *Suppose the system has no dynamics (i.e., $A_k = I$, the identity matrix, and $Q = 0$). If (b_k, a_k) is persistently exciting, then the natural parameters of the site approximations converge to the Newton Method [36], i.e.,*

$$\Omega_{k+1}^s \xrightarrow{p} \psi_{k+1}^{''}(\mu_k) \tag{3.23}$$
$$\omega_{k+1}^s \xrightarrow{p} \Omega_{k+1}^s \mu_k - \psi_{k+1}^{'}(\mu_k), \tag{3.24}$$

where ψ_{k+1} is the negative log-likelihood of the measurement y_{k+1}^b, i.e.,

$$\psi_{k+1}(x) = -\ln(\Phi(y_{k+1}^b(b_{k+1}^T x + b_{k+1}))).$$

Remark. *Note that since $\Omega_{k+1}^s \mu_{k+1}^s = \omega_{k+1}^s$, we can conclude that $\mu_{k+1}^s \xrightarrow{p} \mu_k - [\psi_{k+1}^{''}(\mu_k)]^{-1} \psi_{k+1}^{'}(\mu_k)$, which has the exact form of the Newton Method.*

The significance of Corollary 3 is that since the Newton Method converges to the minimal (maximal) point of a log-convex (-concave) function, then the site approximations converge to the so called Canonical Gaussian Approximation (CGA) [60], i.e., the Gaussian distribution whose mean is the maximizer of the true observation model's probability distribution and whose covariance matrix is the Hessian at that maximum. Finally, it is known that CGA's converge almost surely to a large class of posterior distributions, e.g., as shown by the Bernstein-von Mises Theorem [28]. Thus, Corollary 3 presents strong evidence to believe that the context-aware filter does indeed converge to the mean of the true posterior distribution. Sections 3.7 and 3.8 present multiple evaluations in support of this claim as well.

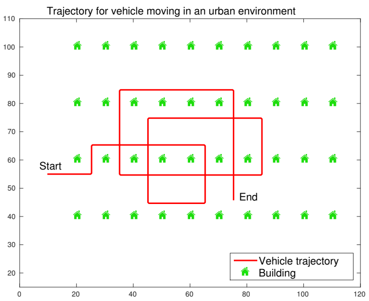


Figure 3.1: Entire LandShark trajectory.

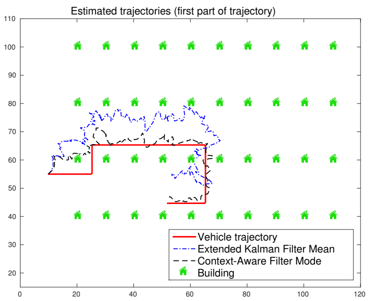


Figure 3.2: Estimated trajectories.

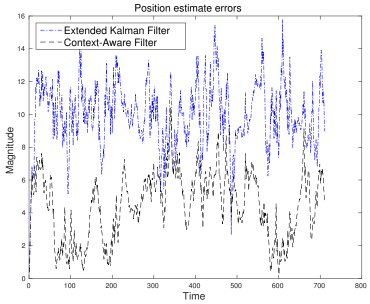


Figure 3.3: Position errors by each filter.

3.6 Evaluation of the Inverse-Exponential Context-Aware Filter: Localization Simulation

Having developed the theory of context-aware filtering in the previous sections of this chapter, in the remaining sections we provide several case-study evaluations in order to illustrate the usefulness of this approach. To evaluate the inverse-exponential context-aware filter, in this section we present a simulation of a robot localization scenario where context measurements can be used to improve state estimation. The sigmoid-based context-aware filter is evaluated, both in simulation and on real data, in the following sections.

In order to evaluate the inverse-exponential filter, we develop a case study using the LandShark robot. In this scenario, the LandShark is moving in an urban environment while trying to visit different waypoints as part of its mission. The vehicle has access to one continuous sensor, namely GPS, in order to perform localization; however, GPS measurements are often inaccurate in urban environments – in this case, they have a large variance and a bias to the North, thereby making localization challenging. In order to improve its state estimates, the LandShark also uses context measurements – it can recognize nearby buildings using image processing and model the corresponding binary measurements using the inverse-exponential functions as described above.

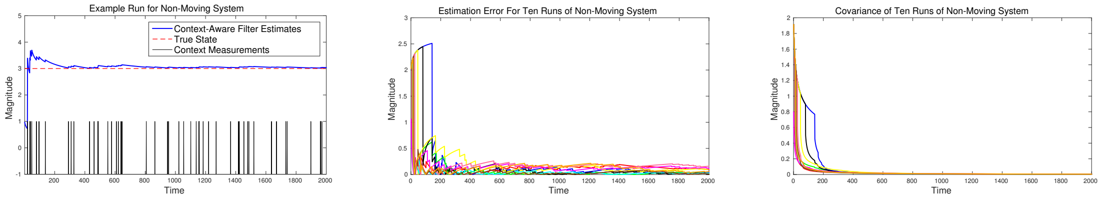
The entire trajectory driven by the LandShark, including the city's map, is presented in Figure 3.1. Note that the LandShark has a differential-drive model, i.e., each turn results in nonlinear state dynamics. Therefore, a linearization is necessary in these cases in order to apply the context-aware filter, similar to the one in a typical extended Kalman filter (EKF).

In order to evaluate the performance of GM-based filter, we compare the accuracy of its estimates with a regular EKF that only uses the continuous GPS measurements. The (first part of the) estimated trajectories by each filter are shown in Figure 3.2. For the EKF, the estimate is chosen to be the mean of the posterior Gaussian distribution at each time step; for the inverse-exponential filter, the estimate is the mode of the distribution (in this application the mode is selected as the element with highest weight in the mixture). As shown in Figure 3.2, the context-aware filter consistently outperforms the EKF – its estimates are closest to the actual trajectory and are more robust to the large variance in GPS measurements, whereas the EKF's estimates tend to change significantly when inaccurate measurements are received.

As further evaluation, Figure 3.3 shows the absolute errors incurred by each filter for the entire trajectory of the LandShark. The context-aware filter's errors are invariably lower than those incurred by the EKF and also do not exhibit much lower variability from one round to the next. All these results suggest that the context-aware filter with inverse-exponential functions can significantly improve state estimation and greatly outperforms continuous-sensor-based approaches, especially in scenarios with inaccurate and unreliable continuous sensors.

3.7 Evaluation of the Probit-Based Context-Aware Filter: Simulations

In this section, we provide three simulation evaluations in order to illustrate different properties of the probit-based context-aware filter. Real-data evaluation is provided



(a) Example run.

(b) Estimation error for ten runs.

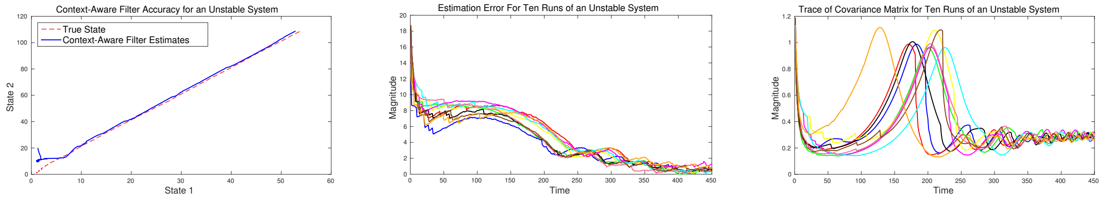
(c) Magnitude of variance for ten runs.

Figure 3.4: Illustration of the performance of the context-aware filter on a non-moving scalar system.

in Section 3.8.

3.7.1 System with No Dynamics

In the first simulation scenario, we evaluate the performance of the filter on a system with no dynamics, in order to illustrate the significance of Theorem 2. Figure 3.4 shows the filter's evaluation on a scalar system with a constant state $x_k = 3$ and with access to one context measurement with corresponding parameters $b_k = 1$ and $a_k = -5$. The initial condition is set to $\mu_0 = 1$, $\Sigma_0 = 2$. Figure 3.4c shows the evolution of the covariance for 10 runs of the system; as expected, the covariance converges to 0 for each one, thus ensuring the convergence of the filter overall. Figure 3.4b shows the estimation errors for the same 10 runs – the figure indicates that the estimates are close to the true state, although some estimates converge more slowly due to different random realizations of the measurements. Finally, Figure 3.4a shows the interesting toothed shape of the estimates for an example run, with discrete jumps as new context measurements are incorporated.



(a) Example run. Note that each axis represents one state of the system.

(b) Estimation error for ten runs.

(c) Trace of the covariance matrix for ten runs.

Figure 3.5: Illustration of the performance of the context-aware filter on an unstable system.

3.7.2 System with Unstable Dynamics

In the second simulation, we evaluate the performance of the context-aware filter on a system with unstable dynamics. The system dynamics are as follows:

$$x_{k+1} = \begin{bmatrix} 1.01 & 0 \\ 0 & 1.01 \end{bmatrix} x_k + \nu_k^p,$$

where $\nu_k^p \sim \mathcal{N}(0, 0.001I)$ and $x_0 = [1 \;\; 1]^T$.[6] 30 context measurements are received at each time, 15 with weights $b_{k,1} = [0 \;\; 1]^T$ and 15 with weights $b_{k,2} = [1 \;\; 0]^T$; the 15 offsets a_k are decreased linearly from 0 to -150 (i.e., they provide rough information as to whether each state is between 0 and 10, 10 and 20, etc.).

Figure 3.5 shows the results of the simulation. We observe similar trends as in Figure 3.4, i.e., the trace of the covariance matrix (Figure 3.5c) converges over time, and the filter's estimates seem to track the real system well after the initial period of uncertainty (Figure 3.5b). These results suggest that the context-aware filter does seem to converge over time (given certain observability-like conditions) and is likely asymptotically unbiased.

[6] Note that systems with larger-eigenvalue dynamics were tested as well with similar results; the system used in this section was chosen for visualization purposes.

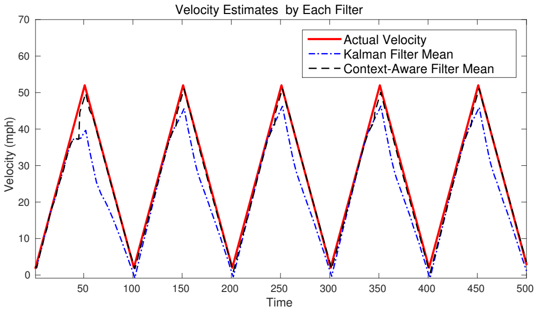


Figure 3.6: Velocity estimates by each filter.

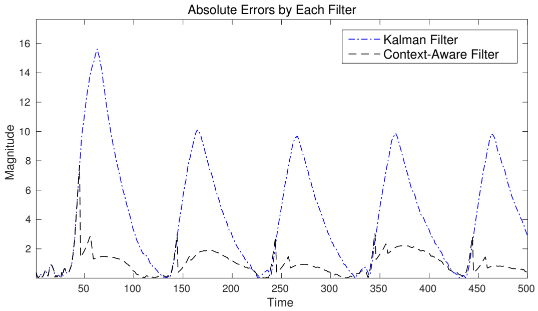


Figure 3.7: Absolute errors by each filter.

3.7.3 Velocity Estimation with Biased Measurements

Finally, we evaluate the sigmoid-based context-aware filter in a scenario with both plant (continuous) and context measurements; note that plant measurements are biased in order to illustrate the fact that context measurements can be used to improve estimation in scenarios where standard sensors are not sufficient. Once again, we use the LandShark robot as the experimental platform. In this case study, the LandShark is moving in a straight line in an urban environment, accelerating to a target velocity and then slowing down for intersections. The LandShark's goal is to estimate its velocity in order to avoid collisions at intersections while moving as quickly as possible. Once again, it has access to GPS measurements to estimate velocity; however, the GPS velocity measurements have a negative bias at high speeds, thus potentially causing the LandShark to apply higher inputs and reach a dangerously high speed. In order to improve estimation, the LandShark can also measure air resistance at the front of the vehicle; while resistance cannot be mapped to speed in a straightforward fashion, it possible to establish whether the vehicle is moving beyond a certain velocity threshold. This information can be converted into a binary context measurement indicating that the LandShark is approaching its target velocity and can be consequently modeled using a sigmoid function.

To evaluate the performance of the sigmoid-based filter, we compare it with a Kalman filter that is only using the continuous measurements (note that a classical

Kalman filter is sufficient in this case since the vehicle is moving in a straight line). Figure 3.6 shows the estimates produced by each filter, including the actual velocity. Once again, the context measurements provide essential information that allows the system to significantly improve its state estimates and overcome the GPS bias, especially when running at high speeds. In addition, Figure 3.7 provides the absolute errors of each filter; the Kalman filter's errors have a much larger variance and are invariably higher as well, similar to the localization scenario discussed above. Thus, this case study also supports the conclusion that the context-aware can be used to greatly improve estimation by incorporating binary context measurements.

3.8 Estimation of Blood Oxygen Concentration Using Context-Aware Filtering

While the previous case studies were simulated and focused on automotive CPS, in this section we investigate a medical CPS application of the sigmoid-based context-aware filter using real-patient data collected at the Children's Hospital of Philadelphia (CHOP). In particular, we address the problem of non-invasively estimating the concentration of oxygen (O_2) in the blood, one of the most closely monitored variables in operating rooms (ORs).

The motivation for this problem comes from the fact that modern ORs are equipped multiple devices that measure various vital signs and provide clinicians with ample information about the patient's state. Analyzing this information in real time can be challenging in a busy OR, especially when trends over time and correlations between variables need to be considered. The OR setting fits exactly in the framework of this thesis, namely the design and development of CPS that process multi-sensor information and provide safety analysis of the resulting system.

As mentioned above, the specific problem addressed in this section is the estimation of the O_2 concentration (also referred to as content) in the blood; the O_2

content has to be maintained within certain limits at all times because too low values can lead to organ failure and brain damage whereas too high concentration could be toxic. Therefore, one of clinicians' highest priorities is controlling the O_2 by keeping a stable and sufficient end-organ perfusion.

Similar to the previous case studies, the estimation of O_2 content is made challenging by imprecise measurements. The total concentration can currently be measured only in an invasive fashion, i.e., by drawing blood from the patient. As a result, clinicians use a non-invasive alternative, namely the hemoglobin-oxygen saturation (S_pO_2), which is measured by a pulse oximeter. While S_pO_2 is a good measure of the concentration in the location where it is measured (e.g., a finger tip), it is not a good indication of the O_2 content in other parts of the body as there may be differences in perfusion (e.g., as caused by a tourniquet on a limb). Furthermore, monitoring S_pO_2 forces clinicians to perform reactive control only – they take action when low S_pO_2 is observed, at which point the patient may already be in a critical state.

As a proactive way of controlling the O_2 concentration, clinicians also monitor the remaining O_2 content (i.e., non-hemoglobin-bound), namely the content of O_2 dissolved in arterial blood. Unlike S_pO_2, the partial pressure of dissolved oxygen in arterial blood (denoted by P_aO_2) can used as a predictive control indicator – P_aO_2 drops significantly before major decreases in the overall concentration are observed. At the same time, P_aO_2 is currently only measured by drawing blood from the patient, which is invasive and requires more time (on the order of several minutes), thus losing its predictive value.

To overcome this problem, in this thesis we address the problem of estimating P_aO_2 non-invasively and in real time. To do so, we employ other measurements available in real time in modern ORs, namely the fractions of O_2 and carbon dioxide (CO_2) in inhaled and exhaled air, respectively, the pressure and volume of inhaled air, and the respiratory rate. By using the correlation between P_aO_2 and these pulmonary variables, one can estimate P_aO_2 without having to draw blood from the patient. However, this approach introduces another challenge – models describing the circulation of O_2 in the blood and airways are imprecise and contain multiple parameters that vary widely across patients, e.g., metabolic rate, lung membrane thickness, arterial wall thickness. While it may be possible to learn some of these parameters given enough data, most of them are not identifiable using non-invasive measurements only. At the same time, certain binary correlations between variables can be established, e.g., when S_pO_2 is above a certain threshold, then the overall O_2 concentration must be above a certain threshold as well. Having obtained such binary measurements, we can now apply the context-aware filter to the O_2 content estimation problem.

3.8.1 Related Work

Related work in the MCPS domain can be broadly divided in three areas: verification, detection and estimation [125, 142, 149]. When precise models are available, it is possible to use formal methods in order to analyze the system and ensure it satisfies certain safety properties; several applications have been investigated in this domain, including the cardiac pacemaker [33, 154], the artificial pancreas [37, 122, 217] and the verification of the infusion pump [16, 157].

The most common approach to detection of adverse events in hospitals is the use of threshold alarms [117]. These systems work by tracking a single vital sign and raising an audible alarm when an upper or lower threshold is crossed [72]. Threshold alarms are popular because they are simple to implement and understand. However, multiple works have shown that single-variable-tracking alarms are severely limited because they tend to produce a large amount of false alarms, ranging from 57% to 99% depending on the application [32, 88, 206]. This in turn has led to the problem of alarm fatigue in caregivers who would sometimes ignore important alarms believing that they are false [59, 71, 187]. The main reason threshold alarms fail is that physiological models contain a lot of parameters that vary drastically across humans – in order to deal with this issue and to provide consistent and guaranteed performance regardless of the patient physiology, the parameter-invariant detector has been developed [177, 216] and successfully applied to three different detection scenarios, namely critical pulmonary shunt detection [107, 108], meal detection in Type I diabetics [50, 215] and hypovolemia detection [169, 170].

Estimation tends to be harder than detection because it requires knowledge of physiological models in order for the resulting estimates to be accurate. Physiological models are typically developed using compartments [53] – in this setting a compartment may represent an actual physical location, e.g., a lung, or may be an abstraction for a larger component, e.g., the transport of blood from the heart to the tissues. Example compartmental models include the cardiac [204] and insulin-glucose [133] systems. A fundamental challenge of compartmental modeling is the balance between physiological accuracy and model identifiability. While accurate models may better capture human physiology, it is harder to identify their parameters using standard system identification techniques [17, 105, 129]. On the other hand, parsimonious models can be identifiable through the training data, but their accuracy may be poor.

In cases where models may be difficult to develop or identify, it may be possible to use data-driven approaches such as machine learning [41, 54, 144, 160, 174, 175]. In order to perform well, however, machine learning requires rich training data with accurate annotations [29] which may not be available in most medical applications [71, 189]. Moreover, temporal reasoning over clinical data using data-driven techniques is still an open area of research [194, 197]. Thus, it is unlikely that a pure data-driven approach will perform well as an oxygen content estimator. Instead, as explained below, in this dissertation we approach the problem by building a crude physiological model and improving it by providing extra information through context measurements.

(a) A blood gas analyzer [3].

(b) A pulse oximeter [4].

(c) A standard anesthesia machine [5].

Figure 3.8: Measurement devices currently available to clinicians.

3.8.2 Problem Formulation

This subsection outlines the current approach to monitoring the O_2 concentration, notes its drawbacks, and formulates the problem addressed by the context-aware filter.

Currently, clinicians have only one available real-time measurement on the blood side, namely the hemoglobin-oxygen saturation in the peripheral capillaries (S_pO_2), measured by a pulse oximeter (Figure 3.8b) at an extremity (usually a finger tip). The saturation is a good measure of the O_2 concentration in the location it is measured because of the oxygen content equation [218]:

$$C_pO_2 = 1.34 S_pO_2 Hb + 0.003 P_pO_2, \tag{3.25}$$

where C_pO_2 is O_2 concentration in the peripheral capillaries, Hb is the amount of hemoglobin in g/dL, and P_pO_2 is the partial pressure of dissolved oxygen in the peripheral capillaries measured in $mmHg$. As can be observed in (3.25), O_2 appears in only two forms in the blood – it is either bound to hemoglobin or dissolved in the blood. Equation (3.25) shows that, for normal values of P_pO_2 around 80-200 $mmHg$ and of Hb around 12-17 g/dL [218], the majority of O_2 is bound to hemoglobin. Thus, S_pO_2 is a good measure of the O_2 concentration in the peripheral capillaries.

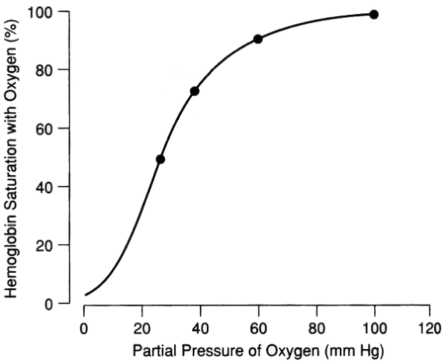


Figure 3.9: A typical hemoglobin dissociation curve for O_2. It shows the shape of the relationship between the partial pressure of dissolved O_2 and hemoglobin saturation. The curve is true for any physiological location, e.g., in the peripheral capillaries the horizontal axis label would be P_pO_2 and the vertical would be S_pO_2.

By assuming that C_pO_2 is just a delay of C_aO_2 (the O_2 concentration in the arteries), S_pO_2 can also be used as a good proxy for C_aO_2. The disadvantage of using only S_pO_2, however, is that it is usually at 100% in healthy humans – by the time it starts dropping, the O_2 content has already decreased; hence, monitoring the O_2 concentration through S_pO_2 is reactive in nature.

In contrast, monitoring the partial pressure of dissolved O_2 is proactive. In addition to (3.25), dissolved O_2 and hemoglobin-bound O_2 are related according to a well-studied hemoglobin dissociation curve [181]. Figure 3.9 shows an example dissociation curve. While the magnitude of the curve may vary across patients, the overall S-shape remains the same. Figure 3.9 shows that for large values of the partial pressure (top right corner), the saturation is close to 100%; at the same time, any noticeable decrease in saturation (and consequently the O_2 concentration) can be observed only after a large decrease in the partial pressure. Thus, monitoring the partial pressure of dissolved O_2 in arterial blood (P_aO_2) provides clinicians with a proactive way of addressing changes in O_2 content before they are reflected in changes in S_pO_2.

Estimating P_aO_2, however, is challenging because it cannot be measured non-invasively and in real time (it can only be measured by drawing blood and analyzing it in a blood-gas analyzer, shown in Figure 3.8a). Instead, we focus on other real-time measurements, available in modern ORs, as means to infer P_aO_2. At CHOP, the anesthesia machine (Figure 3.8c) provides several pulmonary measurements, namely the fractions of O_2 and CO_2 in inspired and expired air, the volume and pressure of inspired air, respiratory rate and others. At the same time, estimating P_aO_2 from these variables is not straightforward – while it is possible to model the relationship between P_aO_2 and the anesthesia machine measurements (e.g., using Fick's laws of diffusion), such models contain multiple parameters that vary widely across patients. Instead of learning these parameters for each patient (which is made challenging by the limited amount of available data), we aim to incorporate the pulmonary measurements by extracting context information from them, thereby improving the overall O_2 content estimates.

Problem 6. *The problem considered in this section is to develop an estimator for P_aO_2 and C_aO_2 by using the noninvasive real-time inputs (fraction of O_2 in inspired air, volume and pressure of inspired air, respiratory rate) and pulmonary measurements (partial pressure of CO_2 in exhaled air) available to clinicians.*

Remark: Our solution uses one blood-gas analysis in order to initialize the estimator.

3.8.3 Physiological Model

In order to develop an estimator for P_aO_2, one needs to first identify a model mapping the available measurements to P_aO_2, as well as formalize the dynamics of the variables in the human body. This subsection develops both of these models; a general-trends dynamics model is described first capturing the first-order effects of the circulation of O_2 around the body. Next, a measurement model is presented, containing both the available regular continuous measurements as well as context measurements derived from the real-time pulmonary measurements.

Table 3.1: Summary of cardiopulmonary partial pressures and blood concentrations. Partial pressures begin with the letter "P" whereas concentrations begin with "C".

Variable Names	Physiological Location
P_iO_2	Airways (inspiration)
P_AO_2	Alveoli
C_aO_2, P_aO_2	Arteries
C_pO_2, P_pO_2	Peripheral capillaries
C_vO_2, P_vO_2	Veins
C_dO_2, P_dO_2	Pulmonary capillaries
P_eO_2	Airways (expiration)

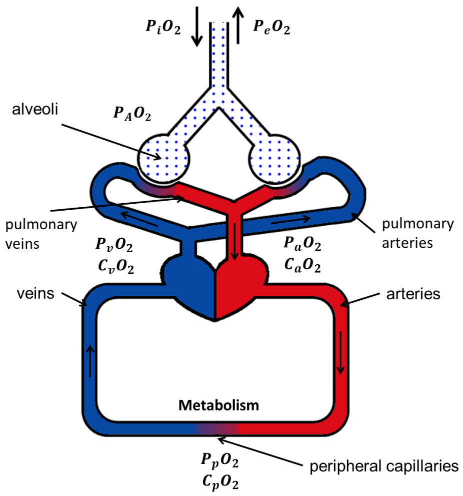


Figure 3.10: A simplified schematic model of O_2 variables in the respiratory and cardiovascular systems.[7]

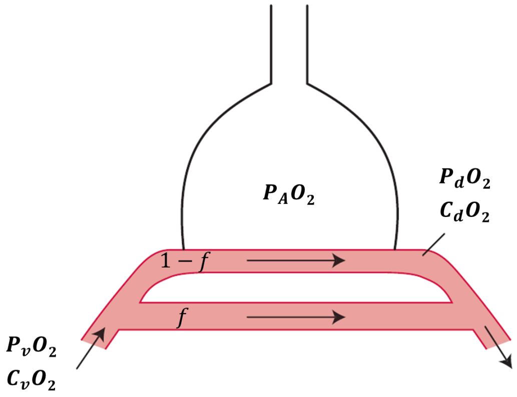


Figure 3.11: An illustration of shunted (bottom) vs. non-shunted (top) blood dynamics in the lung. O_2-rich non-shunted blood participates in diffusion and then mixes with CO_2-rich shunted blood.

3.8.4 Overview of Physiological Variables

Before presenting the actual models, we first provide an overview of the physiological variables used in this section. For reference, all variables are summarized in Table 3.1 and shown in Figure 3.10. In inspired air, the partial pressure of O_2 is denoted by P_iO_2. In the lungs, the air enters the alveoli where the partial pressure is denoted

[7] Note that, for better illustration, the figure shows the pulmonary veins merging before entering the heart, whereas in healthy humans they connect to the left atrium directly.

by P_AO_2. In the alveoli, diffusion occurs, and the gas enters the blood stream at the pulmonary capillaries where the partial pressure of O_2 is denoted by P_dO_2 and the total concentration is C_dO_2. Note that, as shown in Figure 3.11, some of the blood is shunted (e.g., due to blood draining directly into the cavity of the left ventricle through the thebesian veins [218]) and does not participate in diffusion. When the blood from the pulmonary veins enters the heart, it is pumped in the arteries where the partial pressure and concentration are denoted by P_aO_2 and C_aO_2, respectively. The arteries transport the blood to the peripheral capillaries (P_pO_2 and C_pO_2), where metabolism occurs and converts O_2 into CO_2. Finally, the veins (P_vO_2 and C_vO_2) transport the blood back to the lungs and the cardiovascular cycle repeats. The breathing cycle concludes with expiration, where the partial pressure of O_2 in expired air is denoted by P_eO_2.

Dynamics Model

Having introduced the variables and processes at a high level, we now formalize the model dynamics. While models of varying complexity exist in the literature, typically, as the model complexity increases, so does the number of unknown model parameters (e.g., lung capacity, metabolism) that vary across patients. Since these parameters are unidentifiable with current non-invasive measurements, the most popular approach is to use minimal models, i.e., models with a minimal number of parameters that still capture the first- or second-order dynamics of the system. Therefore, we develop a minimal dynamics model, building on results from the work of Kretschmer et al. [120] on estimating P_aO_2 and from our previous work [107] on detecting drops in the O_2 concentration. Our model is approximate in the sense that it captures general trends and relationships among the variables in order to reduce the number of unidentifiable parameters. We use population average values for the few remaining parameters and improve the fidelity of the model by incorporating binary context measurements.

We develop a discrete-time model for the O_2 concentration and later discuss how to convert from concentrations to partial pressures.[8] The relationship between the variables in the airways is governed by the alveolar air equation [79]:

$$P_AO_2(k) = F_iO_2(k)(P_{ATM} - P_{H_2O}) - \frac{P_ACO_2(k)(1 - F_iO_2(k)[1 - RQ])}{RQ}, \quad (3.26)$$

where F_iO_2 is the fraction of O_2 in inhaled air (it can be converted to P_iO_2 using the first term on the right-hand side), P_ACO_2 is the partial pressure of CO_2 in the alveoli, P_{ATM} and P_{H_2O} are the atmospheric and water vapor pressures (in $mmHg$), respectively, and RQ is the respiratory quotient. RQ is a measure of the ratio of O_2 and CO_2 used in metabolism and varies with the type of consumed food. Note that F_iO_2 is set by clinicians, so it can be considered as input, whereas P_ACO_2 is measured by end-tidal CO_2 ($EtCO_2$), i.e., the partial pressure of CO_2 at the end of the breath.[9]

When diffusion occurs, O_2 usually diffuses completely so that the partial pressures are the same:

$$P_dO_2(k) = P_AO_2(k). \quad (3.27)$$

After diffusion, O_2 is in the blood, so its concentration needs to be computed as well. To convert from partial pressure to concentration, one uses (3.25)[10] in combination with the O_2 dissociation curve (Figure 3.9) in order to compute the saturation corresponding to that partial pressure. Let us denote the dissociation curve by g, i.e., g is a function mapping partial pressures of dissolved O_2 to hemoglobin oxygen saturation. Thus, the O_2 concentration in the pulmonary capillaries after diffusion

[8] Our model is discrete-time because the available sensors (at CHOP) have a discrete sampling rate. It does not model the partial pressures of dissolved O_2 directly because the required relationships are nonlinear and would unnecessarily complicate the estimation task.

[9] Note that $EtCO_2$ might be smaller than P_ACO_2 due to dead space, i.e., the volume of air in the airways that is not in contact with blood. However, dead space is about 5% of tidal volume [58], hence it is not considered in this dissertation.

[10] Note that the O_2 content equation is true for any location in the body, i.e., one can replace C_pO_2 and P_pO_2 with C_dO_2 and P_dO_2.

can be expressed as:

$$C_dO_2(k) = 1.34Hb \cdot g(P_dO_2(k)) + 0.003P_dO_2(k). \tag{3.28}$$

Note that g varies greatly between patients. We show how to select g based on population averages below.

Continuing with the cardiovascular dynamics, the concentration in arterial blood, as shown in Figure 3.11, is the weighted average of the concentrations in shunted and non-shunted blood, according to the fraction f of shunted blood. Then

$$C_aO_2(k) = (1 - f)C_dO_2(k) + fC_vO_2(k), \tag{3.29}$$

where the shunted blood has the same O_2 concentration as venous blood.

The O_2 concentration in the peripheral capillaries is assumed to be the same as in the arteries [218], i.e., no reactions occur that change the gas concentrations:

$$C_pO_2(k) = C_aO_2(k). \tag{3.30}$$

Finally, the concentration in the veins is equal to that in the peripheral capillaries minus the effect of metabolism:

$$C_vO_2(k+1) = C_pO_2(k) - \mu, \tag{3.31}$$

where μ captures the patient-specific metabolic rate. Note that a delay is introduced in order to model the fact that it takes time for the blood to travel from the arteries to the veins.

The whole parameterized model can now be summarized in a typical state-space equation:

$$\begin{aligned} a_{k+1} &= (1-f)(d_{k+1}) + f(a_k - \mu) + v_{1,k} \\ e_{k+1} &= e_k + v_{2,k} \end{aligned} \tag{3.32}$$

where $a_k := C_aO_2(k)$, $d_k := C_dO_2(k)$, $e_k := P_ACO_2(k)$, $v_k := [v_{1,k}, v_{2,k}]^T$ is white Gaussian process noise. Modeling the dynamics of $P_ACO_2(k)$ more precisely is possible but introduces more parameters. A random walk model achieves two goals: 1) a linear model with few parameters is maintained and 2) P_ACO_2 is a system state and, hence, may vary less than the noisy $EtCO_2$ measurements.

Note that the model in (3.32) is close to linear in the ranges we are interested in. To see this, note that $F_iO_2(k)$, i.e., the fraction of O_2 in inhaled air, is 21% in breathing air and usually much higher during mechanical ventilation. This means that $P_AO_2(k)$, as computed in (3.26), is also very high (in the extreme case when $F_iO_2(k) = 100\%$, $P_AO_2(k) = 713\text{mmHg}$, with normal values for $P_{ATM} = 760\text{mmHg}, P_{H_2O} = 43\text{mmHg}$). This in turn means that $P_dO_2(k)$ is also high, i.e., in the top right corner of the dissociation curve in Figure 3.9. Therefore, $g(P_dO_2(k)) \approx 1$, i.e., (3.28) simplifies to:

$$C_dO_2(k) = 1.34Hb + 0.003P_dO_2(k). \tag{3.33}$$

Using (3.33) in (3.32), the new model becomes

$$\begin{aligned} a_{k+1} &= (1-f)(1.34Hb + 0.003(c_1u_k + c_{2,k}e_k)) + f(a_k - \mu) + v_{1,k} \\ e_{k+1} &= e_k + v_{2,k}, \end{aligned} \tag{3.34}$$

where $c_1 = (P_{ATM} - P_{H_2O})$, $u_k = F_iO_2(k)$ and $c_{2,k} = (1 - u_k[1 - RQ])/RQ$.

Thus, the above model is a linear time-varying system (note that the input u_k, which also appears in $c_{2,k}$, is multiplied by one of the states, e_k, but this does not introduce non-linearities because we are only considering the estimation problem and not the control problem). There are several parameters in the model; as argued above, these cannot be learned due to unobservability. Thus, we select population average values for the parameters (except for f) and argue that context measurements will correct model inaccuracies. More specifically, based on medical literature [218], these values were selected as: $Hb = 12$ g/dL, $P_{ATM} = 760$ mmHg, $P_{H_2O} = 47$ mmHg, $\mu = 5$ mL/dL, $RQ = 0.8$.

The parameter f, which represents the fraction of shunted blood, does not have typical ranges and can vary widely depending on the patient's condition (e.g., a pulmonary shunt leads to 50% shunted blood). Thus, we adopt an approach used in prior work [120] for the estimation of f. This requires an initializing measurement of P_aO_2 through blood-gas analysis. By obtaining this measurement, one can estimate C_aO_2 through (3.28), where a functional form for g is also assumed, as developed in [115]. Then, using (3.32) and assuming that $a_{k+1} = a_k = a$, one obtains the equation:

$$a = (1 - f)d + f(a - \mu), \tag{3.35}$$

where d is computed from (3.28). This equation can now be solved for f in order to obtain the fraction of shunted blood.

Measurement Model

As usual in the context-aware setting, the available measurements are split into continuous and context. The only available continuous measurement is

$$y_k = e_k + w_k, \tag{3.36}$$

where $y_k := EtCO_2(k)$ and w_k is white Gaussian measurement noise, independent of the process noise v_k.

In addition to the continuous measurement, the system has access to several context measurements. The first context measurement can be derived from the hemoglobin-oxygen saturation – when S_pO_2 is below 100%, this information can be used to upper-bound the O_2 concentration since the majority of O_2 is hemoglobin-bound (as shown in (3.25)). Note that we do not use S_pO_2 measurements directly in the model since mapping the saturation to the concentration of dissolved O_2 requires knowledge of the magnitude of the dissociation curve g. The alarm related to S_pO_2 measurements is raised when $S_pO_2(k)$ drops below 99%. According to (3.25), one can reasonably conclude that if $S_pO_2 < 99\%$, then $C_aO_2 < (1.34 * 0.99)Hb$. This naturally leads to a threshold alarm based on S_pO_2 and to the sigmoid-based context aware filter. The sigmoid parameters of the context detection function in (3.7) can be set to $v_i = [1 \;\; 0]^T, a_i = -(1.34 * 0.99)Hb$.

The second class of alarms consists of several alarms due to the more complicated nature of the signal. This class of alarms aims to use the three other inputs available to clinicians: tidal volume (V_t), respiratory rate (RR) and peak inspiratory pressure (PIP). Each of these inputs affects diffusion through Fick's law of diffusion, which can be stated as follows, adapted to this application [218]:

$$\dot{d}_k \propto cA(P_AO_2(k) - P_dO_2(k)), \tag{3.37}$$

where $d_k = C_dO_2(k)$ as before, c is a constant that captures the O_2 diffusive capacity and lung thickness, and A is the lung surface area. Equation (3.37) states that the number of diffused moles is directly proportional to the surface area and to the difference between the pressures in the lung and in the blood. Note that (3.37) cannot be solved because of the unknown initial condition and unknown parameters. However, one can compute the signal on the right hand side at each point in time; since it is proportional to O_2 diffusion, when the signal is higher, one would expect the O_2 concentration to increase as well.

To construct this signal, note that if we make the usual assumption that a lung is a sphere, then $A \propto V_t^{2/3}$. In addition, since a patient can take several breaths in between two measurements, the respiratory rate can be used as well in order to compute a "cumulative tidal volume" since the last measurement, i.e.,

$$\bar{V}(k) = \frac{t_S}{60} RR(k) V_t(k), \tag{3.38}$$

where t_S is the sampling time in seconds. Thus $A \propto \bar{V}^{2/3}$.

Furthermore, note that PIP is directly proportional to $P_A O_2$. Thus, one can adapt (3.26) to include PIP as effectively increasing atmospheric pressure:

$$P_A O_2(k) = F_i O_2(k)(P_{ATM} - P_{H_2O} + PIP(k)) - \frac{P_A CO_2(k)(1 - F_i O_2(k)[1 - RQ])}{RQ}. \tag{3.39}$$

The final piece of the "diffusion" signal is the initial value of $P_d O_2(k)$. Since the initial value is equal to the venous $P_v O_2(k)$, $P_d O_2(k)$ is directly proportional to $P_a O_2(k-1)$; therefore, we use the expected value of a_{k-1} to obtain an "expected" $P_a O_2(k)$. To obtain a rough estimate of the partial pressure, one needs to invert (3.25) and solve the following nonlinear equation (e.g., by using simplex methods):

$$\mathbb{E}[a_{k-1}] = 1.34 Hb \cdot g(\mathbb{E}[P_a O_2(k-1)]) + 0.003 \mathbb{E}[P_a O_2(k-1)], \tag{3.40}$$

where $\mathbb{E}$ denotes the expectation operator; note that a functional form of g must be assumed, e.g., as in [115]. Thus, the final constructed signal is:

$$s_k = \bar{V}(k)^{2/3} * (P_A O_2(k) - \mathbb{E}[P_a O_2(k-1)]). \tag{3.41}$$

In order to use s as a context measurement, one needs to identify changes in its baseline and raise alarms. To do this, an initial baseline of the signal is selected, and alarms are raised if the signal is too high or too low with respect to that baseline. In particular, suppose the first blood-gas measurement of $P_a O_2$ is received at time step q; then the value of s_q is selected as a baseline and alarms are raised at a later step k

if s_k is lower than $0.5s_q$ or $0.8s_q$ or if it is higher than s_q, $1.2s_q$, or $1.5s_q$. Therefore, similar to the first class of context measurements, one can use a sigmoid function to model these binary measurements.

To select the respective C_aO_2 thresholds, we note that since s_k is directly proportional to C_aO_2, a relative change in s_k should result in a similar relative change in $C_aO_2(k)$. Thus, we identify the baseline $C_aO_2(q)$ and set the thresholds accordingly. For example, if $s_k < 0.8s_q$, then an alarm is raised and the corresponding sigmoid parameters are $v_i = [1 \;\; 0]^T$, $a_i = -0.8C_aO_2(q)$. The other thresholds are derived similarly.

This fully specifies the context observation model and completes the full system model. The following subsection presents the case-study evaluation of this model and of the resulting context-aware filter.

3.8.5 Case Study

This section presents a case-study evaluation of the context-aware estimator for P_aO_2. We use real-patient data collected during lung lobectomy surgeries on children performed at CHOP. A lung lobectomy is the surgical removal of a lung lobe, often due to disease such as cancer or a cystic lung lesion; lobectomies often require one-lung ventilation (i.e., the endotracheal tube is inserted down a mainstem bronchus, so the patient breathes with one lung only) in order to keep the perioperative lung still. In children, one lung is often not enough to provide sufficient O_2 to the body, hence the O_2 concentration tends to decrease.

For evaluation purposes, we use the blood-gas samples taken during these cases and compare them with our estimates. As mentioned earlier, clinicians do not usually draw blood unless they suspect a problem, hence there are at most several measurements per case, while most cases do not have any. After removing all cases with less than two measurements (recall that one is necessary for the algorithm initialization), we retain 51 cases overall. In each case, we initialize the context-aware filter with

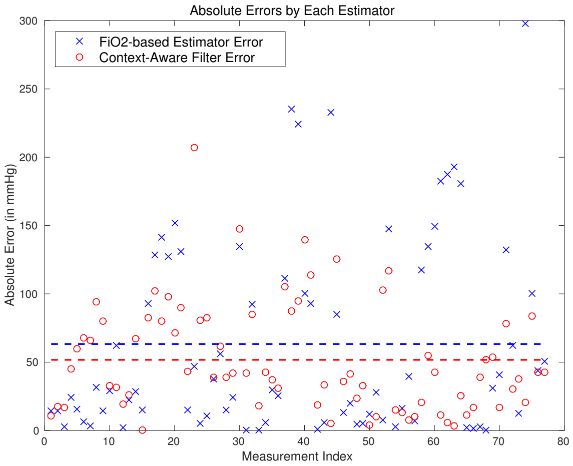
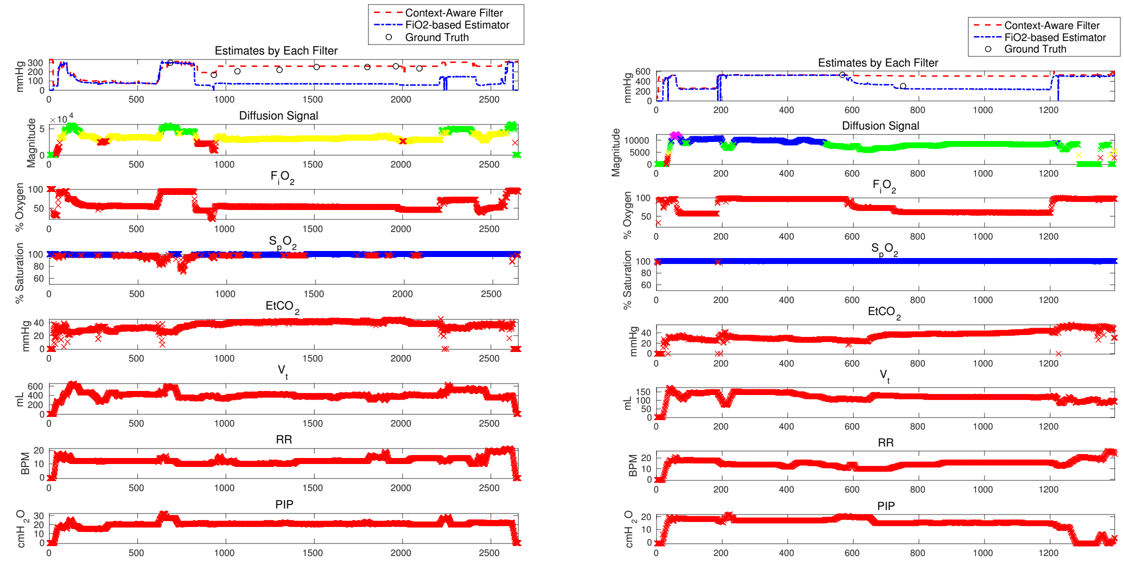


Figure 3.12: Absolute errors for each of the two compared P_aO_2 estimators, the context-aware filter and the F_iO_2-based estimator. Red dashed line shows the average error of the context-aware filter, whereas blue dashed line indicates the average error of the F_iO_2-based estimator.

the first blood-gas measurement and evaluate it on the remaining ones. In addition, as described in the previous section, the diffusion signal baseline (used to define context measurement thresholds) is also computed at the time of the first blood-gas measurement. Finally, note that the available blood-gas measurements only contain P_aO_2 measurements, hence **only P_aO_2 estimates are evaluated**.

Figure 3.12 presents the absolute errors of the context-aware filter, with all measurements from all patients stacked together. For better evaluation, we compare the filter with a P_aO_2 estimation algorithm developed in previous work that uses a similar model and also requires one blood-gas measurement for initialization [120]; this algorithm is named here “F_iO_2-based estimator”. As can be seen in the Figure, the context-aware filter eliminates all of the F_iO_2-based estimator outliers except for one (discussed below). In addition, the context-aware filter achieves a lower average error overall, 51.7 mmHg, than the F_iO_2-based estimator's average error, 63.3 mmHg. To put the error in perspective, note that P_aO_2 measurements are usually in the 200-400 mmHg range (due to F_iO_2 being usually close to 100%), with the exception of a few cases with infants where it is in the 100-200 mmHg range. With



(a) Example case with good estimation by the context-aware filter.

(b) Example case with bad estimation by the context-aware filter.

Figure 3.13: Example cases for different scenarios. Red S_pO_2 data points indicate low-S_pO_2 alarms; blue S_pO_2 data points indicate no S_pO_2 alarms. Diffusion signal: red data points indicate $0.5s_q$ alarms; yellow data points indicate $0.8s_q$ alarms; green data points indicate no alarms; blue data points indicate $1.2s_q$ alarms; magenta data points indicate $1.5s_q$ alarms (recall s_q is the diffusion signal at the initialization point, i.e., first blood-gas analysis).

this in mind, errors of 100 mmHg are still significant; yet, the reasonably uniform distribution of the errors suggests that the context-aware filter is not greatly affected by inter-patient variability and is thus a reasonable choice of estimator, once a more accurate model and more precise context measurements are obtained.

To further analyze the performance of the context-aware filter, we analyze two cases, one with very good performance and one with bad performance. Figure 3.13a presents an example case where context measurements bring a significant improvement.[11] It shows the estimates of each of the two estimators, together with the blood-gas samples, as well as all other measurements and inputs used in the filters. Note that after the initializing blood-gas measurement, clinicians reduce F_iO_2

[11] Note that the estimates prior to the first blood-gas sample are not used for evaluation but are included for completeness.

(around time step 800), probably content with the patient's current condition. Yet, other inputs (V_t, RR, PIP) do not change greatly, indicating that the patient's O_2 concentration should not decrease significantly. This is confirmed by the diffusion signal, which only decreases by about 20%; thus the $0.8s_q$ alarm is raised but the $0.5s_q$ alarm remains silent, which causes the filter to set the estimate somewhere in between. In contrast, the F_iO_2-based estimator is greatly affected by the reduced F_iO_2.

As an example bad-performance case, we consider the outlier in Figure 3.12 for the context-aware filter. Note that, once again, the context-aware filter is not greatly affected by the decreased F_iO_2. In this case, the problem is that the diffusion signal is actually too low at the initialization stage (around step 580), so high-signal alarms are raised later. A possible explanation for the bad performance of the filter in this case is a wrong timestamp of the first blood-gas sample; these timestamps are entered manually and are prone to significant errors, as explored in prior work [189]. In particular, note that tidal volume and respiratory rate are steadily decreasing from around step 420 onwards; thus it is not unlikely that the blood-gas sample was obtained at that time as well. As is apparent from the diffusion signal, if the baseline is set around step 420, no high-signal alarms would be raised later. Finally, note that estimation is made harder by the lack of low-S_pO_2 alarms.

Based on these results, we conclude that the context-aware filter is a promising direction for future research in the MCPS area. By incorporating auxiliary information, it is able to correct some of the deficiencies of imprecise models and results in better estimation overall, even when the variables in question are unobservable.

Chapter 4

Safety Detection Using Sensor Fusion

In this chapter, we consider the main problem of this dissertation, namely safety detection. As noted in Chapter 1, modern systems can fail and be attacked in arbitrary ways; thus, it is difficult to justify any assumptions about the timing or types of faults/attacks. At the same time, since we are primarily interested in providing accurate information to the controller (in the form of estimation and detection), we focus on attacks and faults in sensors only and assume that other components (e.g., actuators) behave as modeled.

We address the problem of safety detection in the presence of arbitrary sensor attacks and faults by using the inherent redundancy in modern CPS. As shown in Chapter 1, modern CPS have access to multiple sensors that can be used to provide redundant information (e.g., several sensors can be used to estimate velocity in the LandShark robot). Using redundancy allows us to develop safety detection techniques without making assumptions about how and when sensors might fail/be attacked; as argued in Chapters 1 and 2, such assumptions are made in most related work on detection/estimation, which makes those approaches not suitable for the problem considered in this dissertation.

Sensor redundancy has been explored in depth in the area of sensor fusion where sensors are generally considered to measure the same variable but through different means and with different accuracies [109, 136]. One of the first works in this field [136] 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 [136] relaxes the worst-case guarantees in favor of obtaining more precise fused measurements through weighted majority voting [38]. 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 [222]. Finally, sensors can be assumed to not only provide intervals but also multidimensional rectangles and balls [51] and more general sets as well [139, 140].

Sensor redundancy has also been applied to multiple fault detection and isolation problems where relations between sensor measurements can be derived [211]. Similarly, it might also be possible to combine the measurements and draw conclusions using a voting [45, 114] or a fuzzy voting scheme [30].

The concept of sensor redundancy has also been used in attack resilience research as well. In particular, Fawzi et al. [78] provide worst-case state estimation analysis depending on the number of attacked sensors; more specifically, they derive sufficient conditions on the maximum number of attacked sensors the system can tolerate, i.e., conditions under which the system can recover its initial state. This result was extended for the purposes of resilient state estimation where the authors considered bounded process and measurement noise as well [158, 159].

In this dissertation, we provide several contributions over related works. First of all, we modify the sensor fusion framework in order to handle multidimensional measurements. In particular, we model each sensor as providing a multidimensional polyhedron (constructed around the physical sensor's measurement). Based on the assumption that at most half of all sensors are attacked/faulty, a bounded fusion polyhedron is derived that is guaranteed to contain the true value. The fusion polyhedron is then used for safety detection – if it does not contain any unsafe states, the system is considered safe.

In addition, we develop a few algorithms for reducing the size of the fusion polyhedron (thereby improving the guarantees of sensor fusion). In the first, we incorporate historical measurements in order to improve the output of sensor fusion. More specifically, we develop several approaches for mapping historical measurements to the current time and compare them in terms of the volume of the resulting fusion polyhedron. Different optimal approaches are presented depending on the assumptions on attacked sensors (e.g., the same sensors are attacked at all times).

In the second approach, we revisit the overall system architecture and note that in many modern CPS, nodes communicate over a shared bus (e.g., a CAN bus in automotive CPS); thus, all sensor measurements can be observed by all other system components, including attacked ones. In addition, these systems are often implemented in a time-triggered fashion where at every round of execution, each sensor transmits its measurement during its allocated time slot, according to a predefined schedule [184, 199]. This in turn means that, depending on the schedule, the attacker can consider other (correct) sensor measurements before sending his own in an attempt to increase the uncertainty of the sensor fusion output while remaining undetected. Therefore, we consider different communication schedules (based on sensors' precisions) and investigate how they affect the attacker's impact on the output of sensor fusion. We provide both theoretical and experimental evidence to show that systems with similar architectures to the one considered in this work should implement the *Ascending* schedule, which orders sensors according to their precision starting from the most precise.

Finally, we note that the above algorithms assume that less than half of all sensors are faulty or attacked at any given time. However, sensors often experience transient faults that recover on their own (e.g., GPS losing connection in a tunnel)

– since transient faults are a normal part of system operation, controllers should be designed to be robust and achieve a guaranteed level of performance regardless of the manifestation of these faults. On the other hand, sensor fusion loses its worst-case guarantees if all sensors are allowed to be faulty at a given time. Thus, we develop a transient fault model for each sensor and a corresponding sensor fusion algorithm that is still guaranteed to contain the true state (and can be used for safety analysis) even in the presence of transient sensor faults.

All of the above approaches are evaluated both in simulation and in experiments using the LandShark robot. Thus, we believe that this is a powerful framework that can be used to improve the resilience of any modern CPS that have access to redundant information.

4.1 System Model and Problem Formulation

This section formalizes the sensor fusion framewrok as well as the attack models used in rest of this chapter. The problem formulations considered in this chapter are stated as well.

4.1.1 System Model

We begin by noting that many of the techniques used in sensor fusion are independent of system dynamics (i.e., they are applied at every time step and provide guarantees even if the dynamics are unknown). That is why we do not specify a dynamics model at this point and leave the dynamics model in its most general form, i.e.,

$$x_{k+1} = f(x_k, u_k) + \nu_k^p. \tag{4.1}$$

At the same time, some of the following sections are developed with specific dynamics models in mind – the corresponding assumptions are always explicitly noted in their respective sections.

The sensor model, on the other hand, is markedly different from the one in Chapter 3 – while in Chapter 3 we used a probabilistic model, here we adopt an abstract sensor model (also known as a set membership model). The reason for this is that although probabilistic models are well suited for describing a system's expected operation and expected state estimation given the measurements, their safety detection performance may suffer when the wrong noise distributions are selected. Under the abstract model, on the other hand, a set is constructed around each sensor's measurement containing all possible values for the true state, where the size of the set depends on the sensor's accuracy. By tracking these sets over time, one may be able to draw conclusions about the system's safety even in the worst case, e.g., if none of the received "measurements" contain unsafe states, then the system must be safe. Thus, the abstract model does not require any assumptions on the process or measurement noise distributions and is naturally suited for safety and security analysis.

Another modification to the sensor model is that in this chapter we abstract away the functional relation between the state and the measurements. More specifically, we assume that all sensors measure the state directly despite the fact that the actual measurements may be some non-linear functions of the state. This assumption allows us to consider sensors as truly providing redundant information and to directly compare their "measurements". Note that while this assumption may not hold in certain systems (e.g., in medical scenarios it is difficult to convert most available measurements to other available measurements), it is a reasonable assumption in many other cases where the same variable may be estimated through several sensors (e.g., speed can be estimated using multiple sensors on the LandShark as shown in Figure 1.2a). Naturally, these different sensors will have varying accuracy depending on the estimation technique that is used; yet, by leveraging the redundant information that they provide, the system should be able to detect when it is unsafe even in the presence of attacks/faults in some of the sensors.

We now formalize the above notions by using the abstract sensor framework, as noted above. Thus, each sensor i provides a direct measurement of the state at time k of the form

$$y_{i,k} = x_k + \nu^m_{i,k}, \tag{4.2}$$

where $\nu^m_{i,k}$ is bounded measurement noise. Using the bounds on ν^m, one may then construct the set of all possible values for x_k given $y_{i,k}$. These bounds can be obtained by using sensor specifications and manufacturer guarantees or they can also be learned from data by observing the system's operation and the largest deviations of the measurements from the true states.

An intuitive approach to specifying the bounds on ν^m is to select bounds in each dimension independently, i.e., form an d-rectangle around the measurement. However, since most modern sensors employ internal filtering techniques (e.g., Kalman filters in GPS) these bounds are not always as simple as d-rectangles; furthermore, some camera-based velocity and position estimators used in urban robotics applications, for example, guarantee different position precisions at different velocities. Therefore, we use a more expressive notion than d-rectangles, namely d-dimensional polyhedra.[1] Thus, each abstract sensor i can now be considered as providing an d-dimensional polyhedron $P_{i,k}$ (constructed around the actual measurement $y_{i,k}$) of the form

$$P_{i,k} = \{y_{i,k} + z \in \mathbb{R}^d \mid B_i z \leq b_i\}, \tag{4.3}$$

where $B_i \in \mathbb{R}^{q \times d}$ and $b_i \in \mathbb{R}^q$ (for some q) are parameters that are determined by the accuracy of sensor i.

By construction, each polyhedron $P_{i,k}$ in (4.3) is guaranteed to contain the true state under nominal conditions. At the same time, sensors often experience transient

[1] Note that in some areas the term "polyhedron" is used to refer to three-dimensional objects only. In this work, polyhedra can have arbitrary dimensions; in some areas, a "convex polytope" is another synonym for "polyhedron" as used in this thesis.

faults, e.g., a camera might be affected by the sun or by temporary obstructions. Thus, we distinguish between a correct and a faulty measurement depending on whether a polyhedron contains the true value.

Definition. *A measurement is said to be* correct *if the corresponding polyhedron contains the true state times and* faulty*, otherwise.*

4.1.2 Attack Model

In addition to sometimes being faulty, a sensor can also be attacked. Note that no assumptions are made on attacked sensors – once a sensor is under attack, the attacker can send any measurements on behalf of that sensor. The only assumptions we make are on the number of attacked sensors – we distinguish between two quantities, namely the real number of attacked sensors, denoted by f_a, as well as the assumed upper bound by the system on the number of attacked sensors, denoted by f.

Assumption. *We assume that the (assumed) upper bound on the number of attacked sensors, f, is always larger than the actual number of attacked sensors, and that the number of attacked sensors, f_a, is less than half of all sensors, i.e.,*

$$f_a \leq f \leq \lceil n/2 \rceil - 1, \tag{4.4}$$

where n is the total number of sensors.

This assumption ensures that the problem is decidable – if it does not hold, then the system cannot provide any bounds on the true state.

4.1.3 Problem Statement

Given the system and attack models defined above, we can now state the safety detection problem considered in this dissertation. Note that in the sensor fusion framework we perform safety detection by checking whether the fusion polyhedron (which is guaranteed to contain the true state) contains any unsafe states. Thus, the problem of sensor fusion is to obtain a minimal-in-volume fusion polyhedron that is guaranteed to contain the true state.

Note that we first address the nominal sensor fusion problem where sensors are assumed to be always correct unless they are attacked. In this setting, we address three problems: 1) the problem of obtaining a minimal fusion polyhedron in a single time step; 2) the problem of obtaining a minimal fusion polyhedron when historical measurements are used as well; 3) the problem of analyzing different schedules of measurement transmissions in order to minimize the attacker's information and impact on the size of the fusion polyhedron.[2] These three problems are stated below.

Problem. *The first problem in the sensor fusion framework is how to obtain a fusion polyhedron in a single time step, i.e., a minimal-volume polyhedron that is guaranteed to contain the true state.*

Problem. *The second problem is to incorporate historical measurements in the sensor fusion algorithm in order to further reduce the volume of the fusion polyhedron while preserving the guarantee that it contains the true state.*

Problem. *The third problem is to consider different schedules of measurement transmissions in order to limit the attacker's impact on the size of the fusion polyhedron.*

Note that the above problems are all addressed in the nominal case, i.e., when non-attacked sensors always provide correct measurements. However, if sensors are allowed to temporarily provide faulty measurements as well, then the fusion polyhedron would not be guaranteed to contain the true state at all times since it is possible that all sensors might provide faulty measurements at the same time. Thus, we also consider the problem of sensor fusion in the presence of transient sensor faults.

[2] Note that the third problem is only addressed in the one-dimensional case as shown in Section 4.6.

Problem. *The problem of sensor fusion in the presence of transient sensor faults is to produce a bounded fusion polyhedron that is guaranteed to contain the true state at each time step despite the fact that unattacked sensors might experience transient faults (choosing an appropriate model for transient faults is also a contribution of this dissertation).*

4.2 Sensor Fusion in One Dimension

We begin our discussion of sensor fusion with the special one-dimensional case. In this case each sensor's polyhedron reduces to an interval. As discussed in the introduction of this chapter, sensor fusion with intervals has been studied extensively in the related literature. In this section, we briefly describe the classical sensor fusion algorithm in one dimension [136].

The inputs to the one-dimensional sensor fusion algorithm are n real intervals, and a number f that denotes an upper bound on the number of attacked[3] intervals the system might have. The fusion interval is then computed as follows: its lower bound is the smallest point contained in at least $n-f$ intervals and the upper bound is the largest such point. Intuitively, the algorithm works conservatively: since at least $n - f$ intervals are correct, any point that is contained in $n - f$ intervals may be the true value, and hence it is included in the fusion interval.

The algorithm is illustrated in Figure 4.1. When $f = 0$ and the system is confident that every interval is correct, the fusion interval is just the intersection of all intervals. When at most one sensor can be attacked ($f = 1$), the fusion interval is the convex hull of all points contained in at least four intervals. Similarly, when $f = 2$ the fusion interval contains the convex hull of all points that lie in at least three intervals. As shown in Figure 4.1, as f increases so does the uncertainty represented as the size of the fusion interval. In particular, for $f = n - 1$ the fusion interval is the convex

[3] Note that the original sensor fusion work was developed with faulty, not attacked, sensors in mind [136]. We modify the paper's language to refer to attacked sensors.

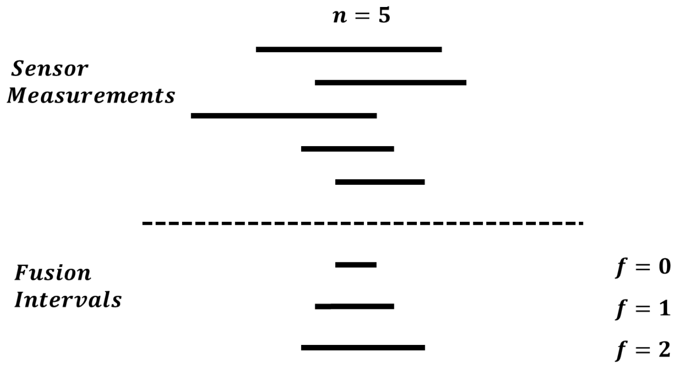


Figure 4.1: Fusion interval for three values of f. Dashed horizontal line separates sensor intervals from fusion intervals in all figures.

hull of the union of all intervals.

Three important results of this work are worth noting. If $f \leq \lceil n/3 \rceil - 1$, then the width of the fusion interval is bounded above by the width of some correct interval. Additionally, if $f \leq \lceil n/2 \rceil - 1$, the width of the fusion interval is bounded above by the width of some interval (not necessarily correct). Finally, if $f \geq \lceil n/2 \rceil$, then the fusion interval can be arbitrarily large. Thus, as noted in Section 4.1.2, we assume that f is always at least as large as the true number of attacked sensors, f_a, and always less than half of all sensors, i.e., $f_a \leq f \leq \lceil n/2 \rceil - 1$, causing the fusion interval to be bounded.

4.3 Notation

Before we address the problem of multidimensional sensor fusion, we introduce some notation that is used throughout this chapter. Let $\mathcal{N}_k$ denote all n polyhedra at time k. We use $S_{\mathcal{N}_k,f}$ to denote the fusion polyhedron given the set $\mathcal{N}_k$ and a fixed f. Let $|P|$ denote the volume of polyhedron P; in particular, $|S_{\mathcal{N}_k,f}|$ is the volume of the fusion polyhedron. We use $\mathcal{C}_k$ to denote the (unknown) set of all correct polyhedra. Finally, we use $s_1, \ldots, s_n$ to denote the sensors themselves (not their measurements).

Algorithm 1 Sensor Fusion Algorithm

```
Input: An array of polyhedra $P$ of size $n$ and an upper bound on the number of
    corrupted polyhedra $f$
$C \leftarrow combinations\_n\_choose\_n\_minus\_f(P)$
$R_{\mathcal{N}_k,f} \leftarrow \emptyset$
for each $K$ in $C$ do
    add($R_{\mathcal{N}_k,f}$, intersection($K$))
end for
return conv($R_{\mathcal{N}_k,f}$)
```

4.4 Multidimensional Sensor Fusion

The sensor fusion alorithm in the multidimensional case uses the same intuition as the one-dimensional case, with the main difference being that different tools are required to argue over general polyhedra as opposed to one-dimensional intervals. The sensor fusion algorithm is described in Algorithm 1. It is based on the algorithm for d-rectangles described by Chew and Marzullo [51]. It computes the fusion polyhedron by finding all regions contained in $n - f$ polyhedra, denoted by $R_{\mathcal{N}_k,f}$, and then taking their convex hull in order to return a polyhedron, i.e.,

$$S_{\mathcal{N}_k,f} = \texttt{conv}(R_{\mathcal{N}_k,f}), \tag{4.5}$$

where $\texttt{conv}(\cdot)$ denotes the convex hull. Intuitively, the algorithm works in the same conservative fashion as the one-dimensional case – since there are at least $n - f$ correct polyhedra, any point that is contained in $n - f$ polyhedra may be the true state, and thus it is included in the fusion polyhedron; the convex hull is computed since the output should be in the same format as the inputs (i.e., a polyhedron).

The algorithm is illustrated in Figure 4.2. The system consists of three sensors, hence three polyhedra are obtained, and is assumed to have at most one attacked sensor. Therefore, all regions contained in at least two polyhedra are found, and their convex hull is the fusion polyhedron (shaded).

Proposition 6. *The fusion polyhedron computed by Algorithm 1 will always contain*

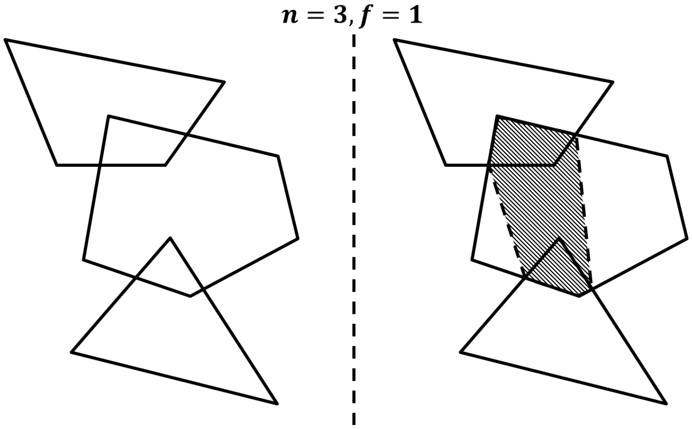


Figure 4.2: An illustration of the proposed sensor fusion algorithm.

the true state.

Proposition 7. *The fusion polyhedron computed by Algorithm 1 is the smallest convex set that is guaranteed to contain the true state.*

Having shown the desired properties of the proposed algorithm, we comment on its complexity. There are two subprocedures with exponential complexity. First, finding all combinations of $n - f$ polyhedra is exponential in the number of polyhedra. Second, computing the convex hull of a set of polyhedra requires finding their vertices; this problem, known in the literature as vertex enumeration, is not known to have a polynomial algorithm in the number of hyperplanes defining the polyhedra (hence in their dimension) [20].

We now provide some bounds on the volume the fusion polyhedron. To prove the first bound, for completeness we first provide the following lemma that will be useful in showing the final result.

Lemma 2. *The vertices of the convex hull of a set of polyhedra are a subset of the union of the vertices of the polyhedra.*

Before formulating the theorem, we introduce the following notation. Let $\min_p \mathcal{B}$ denote the p^{th} smallest number in the set of real numbers $\mathcal{B}$ with size $r = |\mathcal{B}|$.

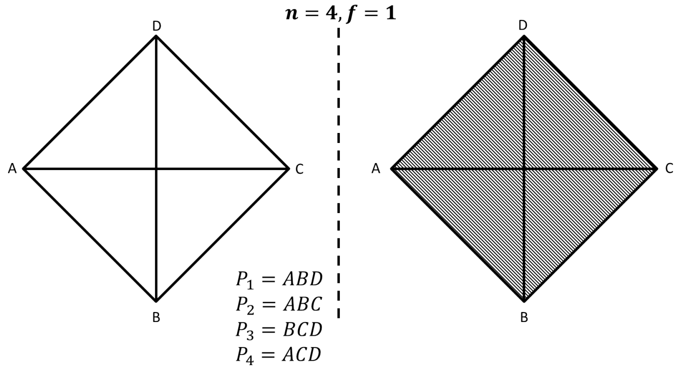


Figure 4.3: An example showing that the bound specified in Theorem 4 is tight.

Similarly, we use $\max_p \mathcal{B}$ to denote the p^{th} largest number in $\mathcal{B}$. We note that $\min_p \mathcal{B} = \max_{r-p+1} \mathcal{B}$ (e.g., if $\mathcal{B} = \{14, 15, 16\}, \min_1 \mathcal{B} = 14 = \max_3 \mathcal{B}$). Finally, let v_P be the number of vertices in the fusion polyhedron.

Theorem 4. *If $f < n/v_P$ then*

$$|S_{\mathcal{N}_k,f}| \leq \min_{fv_P+1}\{|P| : P \in \mathcal{N}_k\}.$$

Theorem 4 suggests that if $f < n/v_P$ then the volume of the fusion polyhedron is bounded by the volume of some polyhedron. We note that this condition may not always hold as the number of vertices of the fusion polyhedron may be the sum of the number of vertices of the original polyhedra. However, the condition is tight in the sense that if it does not hold, then the volume of the fusion polyhedron may be larger than the volume of any of the individual polyhedra. This is illustrated in Figure 4.3. In this case, each polyhedron (P_1, P_2, P_3 or P_4) is a triangle that is a half of the big square, so $n = 4$, and $f = 1 = n/v_P$. Hence the fusion polyhedron, i.e., square, is larger in area than any of the triangles. In cases like this one, we resort to the following bound.

Theorem 5. *If $f < \lceil n/2 \rceil$, then $|S_{\mathcal{N}_k,f}|$ is bounded by the volume of $\boldsymbol{conv}(\mathcal{C}_k)$ (i.e., the convex hull of all correct polyhedra).*

In conclusion, three different upper bounds on the volume of the fusion polyhedron exist based on different values of f. If $f \geq \lceil n/2 \rceil$, then the fusion polyhedron can be arbitrarily large. This is due to the fact that there are now enough corrupted sensors to include points not contained in any correct polyhedra in the fusion polyhedron (as opposed to Theorem 5). On the other hand, if $f \leq \lceil n/2 \rceil - 1$, then $|S_{\mathcal{N}_k,f}| \leq |\mathtt{conv}(\mathcal{C}_k)|$. In addition, if $f < n/v_P$, then the volume of $S_{\mathcal{N}_k,f}$ is bounded from above by the volume of some polyhedron. Note that either of the last two bounds may be tighter than the other depending on the scenario.

4.5 Sensor Fusion Using Historical Measurements

Having developed a sensor fusion algorithm that produces a minimal fusion polyhedron from n polyhedra in a given time step, we now consider the problem of incorporating knowledge of system dynamics to reduce the volume of the fusion polyhedron by using measurement history. In this section, we assume that state dynamics have the following linear form:

$$x_{k+1} = A_k x_k + \nu_k^p, \tag{4.6}$$

where $x_k \in \mathbb{R}^d$ is the state as before, $A_k \in \mathbb{R}^{d \times d}$ is the transition matrix and $\nu_k^p \in \mathbb{R}^d$ is bounded noise such that $\|\nu_k^p\| \leq M$, where $\|\cdot\|$ denotes the L_∞ norm, and M is a constant.

Note that in this setting we are still assuming that non-attacked sensors provide correct measurements at all times. We relax this assumption in Section 4.7.

In order to use historical measurements, one needs to first of all develop a technique for mapping previous measurements to the current time using the dynamics model (similar to the prediction stage of the Kalman filter). For each polyhedron $P_{i,k}$, this can be done using the following map m:

$$m(P_{i,k}) = \{z \in \mathbb{R}^d \mid z = A_k p + q, \forall p \in P_{i,k}, \|q\| \leq M\}.$$

Thus, $m(P_{i,k})$ is once again a polyhedron (due to the linear mapping and the bounds on ν_k^p) that describes the prediction of $P_{i,k}$ one step in the future.

For simplicity, we also introduce the notation $\cap_p$, referred to as *pairwise intersection*. In particular, if $m(\mathcal{N}_k)$ is the mapping of all n polyhedra to the next time step, i.e.,

$$m(\mathcal{N}_k) = \{m(P_{i,k}), i = 1, \dots, n\},$$

then let $m(\mathcal{N}_k) \cap_p \mathcal{N}_{k+1}$ denote the intersection of each sensor s_i's measurement in time $k+1$ with the mapping of s_i's measurement from time k, i.e.,

$$m(\mathcal{N}_k) \cap_p \mathcal{N}_{k+1} = \{P_i' \mid P_i' = m(P_{i,k}) \cap P_{i,k+1}, i = 1, \dots, n\}.$$

Note that this set again contains n polyhedra, some of which may be empty.

It is worth noting here that our assumptions impose a restriction on the number of ways in which history can be used. In particular, we only assume an upper bound on the number of attacked sensors; thus, it is not possible to map subsets of the polyhedra while guaranteeing that the fusion polyhedron contains the true value. In other words, such mappings would require additional assumptions on the number of corrupted sensors in certain subsets of $\mathcal{N}_k$; hence, all permitted actions in this work are:

1. computing fusion polyhedra for all n polyhedra in a given time step;

2. mapping this fusion polyhedron to the next time step;

3. mapping all polyhedra to the next time step, thus doubling both n and f.

Based on these permitted actions, we can now enumerate different ways of historical measurements. While it is challenging to exhaustively list all possibilities, following are the ways of using past measurements considered in this work:

1. *map_n*: In this approach we map all polyhedra in $\mathcal{N}_k$ to time $k+1$, and obtain a total of $2n$ polyhedra in time $k+1$. We then compute their fusion polyhedron

with $2f$ as the bound on the number of corrupted polyhedra. This is illustrated in Figure 4.4a. Formally the fusion polyhedron can be described as

$$S_{m(\mathcal{N}_k)\cup\mathcal{N}_{k+1},2f}.$$

2. *map_S_and_intersect*: This algorithm computes the fusion polyhedron at time k, maps it to time $k+1$, and then intersects it with the fusion polyhedron at time $k+1$, as illustrated in Figure 4.4b. Formally we specify this as

$$m(S_{\mathcal{N}_k,f}) \cap S_{\mathcal{N}_{k+1},f}.$$

3. *map_S_and_fuse*: Here the fusion polyhedron from time k is mapped to time $k+1$, thus obtaining a total of $n+1$ polyhedra at time $k+1$, as presented in Figure 4.4c. Note that f is still the same because $S_{\mathcal{N}_k,f}$ is guaranteed to contain the true value by Proposition 6. Formally this is captured by

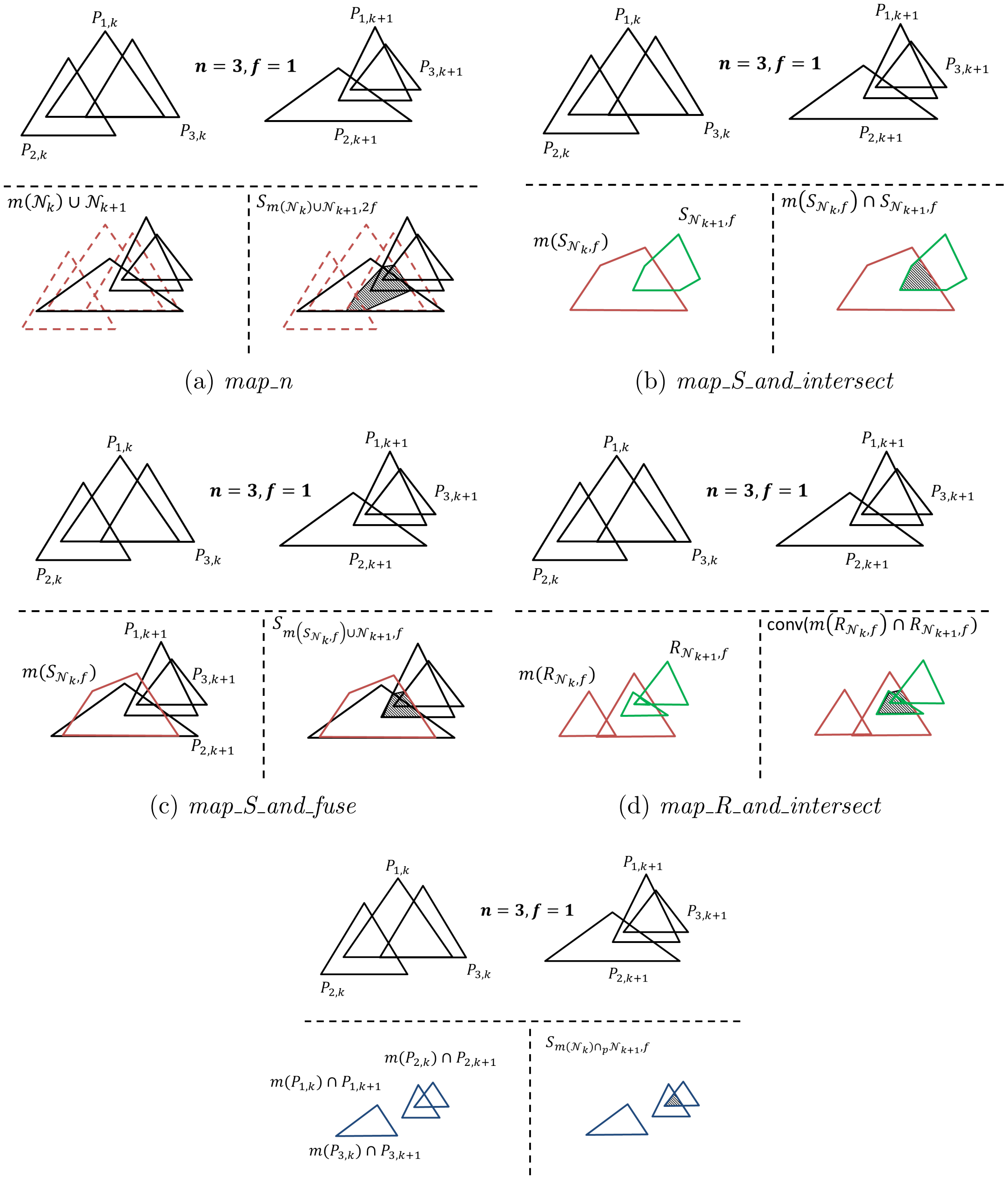
$$S_{m(S_{\mathcal{N}_k,f})\cup\mathcal{N}_{k+1},f}.$$

4. *map_R_and_intersect*: This is similar to *map_S_and_intersect*, but instead we map $R_{\mathcal{N}_k,f}$ to time $k+1$, intersect with $R_{\mathcal{N}_{k+1},f}$, and compute the convex hull as illustrated in Figure 4.4d. Formally we describe this as

$$\texttt{conv}(m(R_{\mathcal{N}_k,f}) \cap R_{\mathcal{N}_{k+1},f}).$$

5. *pairwise_intersect*: This algorithm performs pairwise intersection as shown in Figure 4.4e. Formally we capture this as

$$S_{m(\mathcal{N}_k)\cap_p\mathcal{N}_k,f}.$$


$P_{1,k}$
$P_{2,k}$
$P_{3,k}$
$n = 3, f = 1$
$P_{1,k+1}$
$P_{2,k+1}$
$P_{3,k+1}$
$m(\mathcal{N}_k) \cup \mathcal{N}_{k+1}$
$S_{m(\mathcal{N}_k)\cup\mathcal{N}_{k+1},2f}$
$S_{\mathcal{N}_{k+1},f}$
$m(S_{\mathcal{N}_k,f})$
$m(S_{\mathcal{N}_k,f}) \cap S_{\mathcal{N}_{k+1},f}$
$S_{m(S_{\mathcal{N}_k,f})\cup\mathcal{N}_{k+1},f}$
$R_{\mathcal{N}_{k+1},f}$
$m(R_{\mathcal{N}_k,f})$
$\mathrm{conv}(m(R_{\mathcal{N}_k,f}) \cap R_{\mathcal{N}_{k+1},f})$
$m(P_{2,k}) \cap P_{2,k+1}$
$m(P_{1,k}) \cap P_{1,k+1}$
$m(P_{3,k}) \cap P_{3,k+1}$
$S_{m(\mathcal{N}_k)\cap_p\mathcal{N}_{k+1},f}$

(a) *map_n*

(b) *map_S_and_intersect*

(c) *map_S_and_fuse*

(d) *map_R_and_intersect*

(e) *pairwise_intersect*

Figure 4.4: Illustrations of the different methods of using history. For simplicity $A_k = I$, the identity matrix, and $\nu_k^p = 0$.

The obvious way to compare these algorithms is through the volume of the fusion polyhedra. We provide below a series of results that relate the sizes of the fusion polyhedra for the aforementioned methods of incorporating measurement history. Note that all methods are compared over two time steps only – one can use induction to show the same results hold over the entire timeframe of system operation.

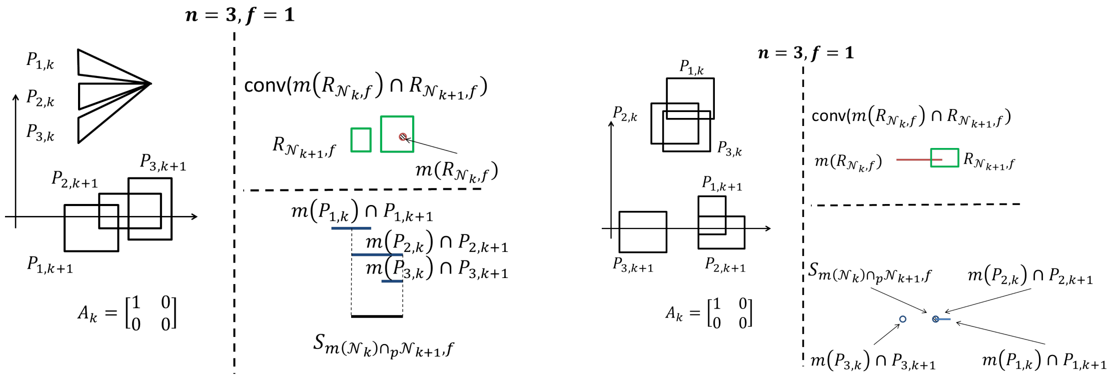
Theorem 6. *The region obtained using map_R_and_intersect is a subset of the region derived by map_n.*

Theorem 7. *The polyhedron derived by map_R_and_intersect is a subset of the polyhedron obtained by map_S_and_intersect.*

Theorem 8. *The polyhedron obtained by map_R_and_intersect is a subset of the polyhedron derived using map_S_and_fuse.*

Theorems 6, 7 and 8 suggest that *map_R_and_intersect* is the best of the first four methods enumerated above as can also be seen in Figure 4.4. This intuitively makes sense since it is only keeping enough information from previous measurements to guarantee that the true value is preserved. In particular, it is not computing the convex hull at time k as *map_S_and_intersect* and *map_S_and_fuse* do (and potentially introduce additional points to the fused region), nor is it mapping potentially corrupted polyhedra as does *map_n*.

We note, however, that without additional assumptions about the rank of A_k, *map_R_and_intersect* and *pairwise_intersect* are not subsets of each other. Counterexamples are presented in Figure 4.5. In Figure 4.5a, $R_{\mathcal{N}_k,f}$ is a single point that is projected onto the x axis. Hence *map_R_and_intersect* is a subset of *pairwise_intersect*, which produces an interval of points. Conversely, Figure 4.5b shows an example where *pairwise_intersect* is a point, and *map_R_and_intersect* is an interval containing that point. It is worth noting, however, that regardless of which of the two approaches is used, *pairwise_intersect* can be used as a preliminary step to detect attacked sensors – if the two polyhedra of a certain sensor have an empty intersection, then the sensor must be attacked in one of the rounds; thus, it can be discarded from both, effectively reducing n and f by one.



$n = 3, f = 1$
$P_{1,k}$
$P_{2,k}$
$P_{3,k}$
$P_{2,k+1}$
$P_{3,k+1}$
$P_{1,k+1}$
$A_k = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$
conv$(m(R_{\mathcal{N}_k,f}) \cap R_{\mathcal{N}_{k+1},f})$
$R_{\mathcal{N}_{k+1},f}$
$m(R_{\mathcal{N}_k,f})$
$m(P_{1,k}) \cap P_{1,k+1}$
$m(P_{2,k}) \cap P_{2,k+1}$
$m(P_{3,k}) \cap P_{3,k+1}$
$S_{m(\mathcal{N}_k)\cap_p \mathcal{N}_{k+1},f}$
$n = 3, f = 1$
$P_{1,k}$
$P_{2,k}$
$P_{3,k}$
$P_{1,k+1}$
$P_{3,k+1}$
$P_{2,k+1}$
$A_k = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$
conv$(m(R_{\mathcal{N}_k,f}) \cap R_{\mathcal{N}_{k+1},f})$
$m(R_{\mathcal{N}_k,f})$
$R_{\mathcal{N}_{k+1},f}$
$S_{m(\mathcal{N}_k)\cap_p \mathcal{N}_{k+1},f}$
$m(P_{2,k}) \cap P_{2,k+1}$
$m(P_{3,k}) \cap P_{3,k+1}$
$m(P_{1,k}) \cap P_{1,k+1}$

(a) *map_R_and_intersect* is not a subset of *pairwise_intersect*.

(b) *pairwise_intersect* is not a subset of *map_R_and_intersect*.

Figure 4.5: Examples showing that, in general, polyhedra obtained using *map_R_and_intersect* and *pairwise_intersect* are not subsets of each other if A_k is not full rank.

Finally, we note that if A_k is a full rank matrix and $\nu_k^p = 0$, then *pairwise_intersect* is the best of all five methods, as shown in the following theorem.[4]

Theorem 9. *If A_k is full rank and $\nu_k^p = 0$, the polyhedron obtained by pairwise_intersect is a subset of the polyhedron derived using map_R_and_intersect.*

Therefore, we argue that systems that incorporate past measurements in their sensor fusion algorithms should use the *pairwise_intersect* method. We now show that it satisfies the worst-case requirements, same as the no-history case. In addition, we show that adding historical measurements is always beneficial for the system.

Proposition 8. *The fusion polyhedron computed using* pairwise_intersect *will always contain the true state.*

Proposition 9. *The fusion polyhedron computed using* pairwise_intersect *is never larger than the fusion polyhedron computed without using history.*

[4] A similar counter-example to Figure 4.5 can be found in the case when $\nu_k^p \neq 0$.

Note that *pairwise_intersect* and *map_R_and_intersect* do not add significant computational complexity to the no-history sensor fusion algorithm described in Section 4.4. While they still suffer from the exponential procedure of computing the fusion polyhedron at each time, each of the two methods requires storing at most n polyhedra to represent historical measurements – intuitively they are the "intersection" of all past measurements. Thus, implementing any of these methods will not add substantial computational or memory cost for the system. The algorithm's implementation is discussed in greater detail in the evaluation section.

4.5.1 Evaluation

Given our results in Section 4.5, we argue that systems with linear dynamics should use the *pairwise_intersect* method. This section provides an algorithm that implements this method and a case study to illustrate its usefulness.

Implementation

The implementation is shown in Algorithm 2. In essence, at each point in time n polyhedra (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 polyhedron, the algorithm also needs very little memory – the required memory is linear in the number of sensors irrespective of how long the system runs.

An important detail that is hidden inside the `pair_inter` function is how attacked sensors are dealt with. If a sensor s_i's two polyhedra have an empty intersection then that sensor must be attacked. In this case, both polyhedra are discarded and n and f are reduced by one. Furthermore, the system has the option of discarding all future measurements provided by the sensor s_i; alternatively, the system may update *past_meas* with s_i's measurement in the next round. Which choice is made depends on the system's trust in the sensor – if it is believed to be continuously under attack,

Algorithm 2 Implementation of the *pairwise_intersect* algorithm

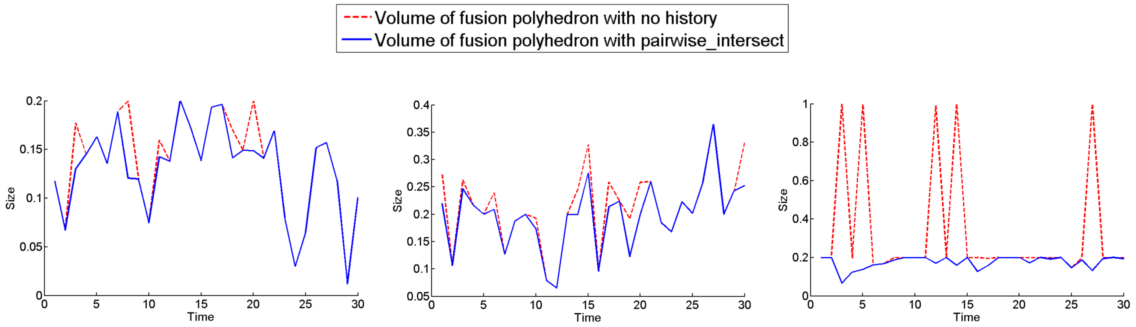
```
Input: $f$, the number of attacked sensors
$past\_meas \gets \emptyset$
for each step k do
    $cur\_meas \gets$ get_meas($k$)
    if $past\_meas == \emptyset$ then
        $past\_meas \gets cur\_meas$
    else
        $past\_meas =$ pair_inter($cur\_meas, past\_meas$)
    end if
    $S \gets$ fuse_polyhedra($past\_meas, f$)
    send_polyhedron_to_controller($S$)
end for
```

then discarding all or some of its future measurements is the better option. However, if it is attacked only in certain conditions (e.g., on certain territory), then its future measurements should be kept and incorporated in the algorithm. Quantification of sensor trust, however, is not within the scope of this paper, hence we take this choice as a design-time decision (input) and leave its analysis for future work.

Case Study

To show the effectiveness of the *pairwise_intersect* approach, we use the sensors available to the LandShark. In this section, we use four of the LandShark's sensors that can be used to estimate velocity – GPS, camera and two encoders. In addition, GPS and the camera can be used to estimate the vehicle's position. Therefore, the encoders provide the controller with interval estimations of the vehicle's velocity only, whereas GPS and the camera send two-dimensional polyhedra as estimates of the velocity and position.[5] The sizes of the encoders' intervals were obtained based on the manufacturer's specification, whereas the shapes and sizes of GPS and camera's polyhedra were determined empirically – the LandShark was driven in the open, and largest deviations from the true values (as measured by a high-precision laser

[5] For this case study we only require one-dimensional position as will become clear in the next paragraph. However, our approach could easily be applied to multidimensional measurements.



(a) GPS under attack.

(b) Camera under attack.

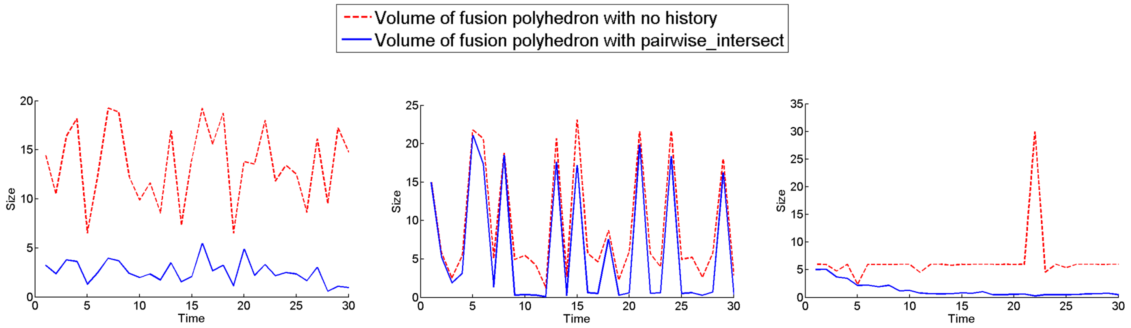
(c) Encoder under attack.

Figure 4.6: Sizes of velocity (ONLY) fusion intervals for each of the three simulated scenarios; Dashed line – volume of the fusion polyhedra when measurement history is not considered, Solid line – volume of the fusion polyhedra obtained using *pairwise_intersect*.

tachometer) were collected.

Given this information, the following three scenarios were simulated. The LandShark is moving in a straight line at a constant speed of 10 mph. In each scenario, a different sensor was attacked such that a constant offset of 1 mph was introduced to the sensor's speed estimate. The sensors' speed tolerances were as follows: 0.2 mph for the encoders, 1 mph for GPS and 2 mph for the camera. GPS's tolerance for position was 30 feet, whereas the camera's tolerance varies with speed (hence its polyhedron is a trapezoid) and was 100 feet at 10 mph. At each point in time, we compute the fusion polyhedron in two ways – using only current measurements and using the *pairwise_intersect* method. Finally, we record the differences and the improvement achieved by using history.

To illustrate consistence with earlier one-dimensional works (e.g., [136]), for each of the three scenarios we first computed the size of the fusion interval in one dimension. Figure 4.6 presents the results. For each scenario, the size of the fusion interval was never larger when using *pairwise_intersect*, while the gain was significant at certain times. This is particularly apparent when the encoder was under attack. The reason for this, as we explain in Section 4.6, is that it is in general beneficial for the attacker to corrupt the most precise sensors.



(a) GPS under attack.

(b) Camera under attack.

(c) Encoder under attack.

Figure 4.7: Sizes of fusion polyhedra of velocity and position for each of the three scenarios simulated; Dashed line – volume of the fusion polyhedra when measurement history is not considered, Solid line – volume of the fusion polyhedra obtained using *pairwise_intersect*.

Figure 4.7 presents the results when two-dimensional polyhedra are considered. Note that in this case there are only two sensors estimating the robot's position – when one is attacked, the size of the fusion polyhedron can grow dramatically. Consequently, *pairwise_intersect* is greatly beneficial for the system as it identifies the attacked sensors and discards their measurements when their polyhedra do not intersect. It is worth noting here that in all simulated scenarios if a sensor is found corrupted in any step we do not disregard its measurement in the next step. Note also that the volumes in Figure 4.7 are much larger than those in Figure 4.6 – this is a result of the fact that position tolerances are measured in feet and are larger than 10. (i.e., 30 feet for GPS). Finally, as consistent with Proposition 8, all fusion polyhedra contained the actual value of velocity (i.e., 10 mph).

4.6 Attack-Resilient Sensor Transmission Scheduling

In this section we present another approach for improving the performance of sensor fusion, namely the introduction of a sensor transmission schedule in order to limit the attacker's information. As described in the introduction of this chapter and as illustrated in Figure 1.1, many modern CPS use a shared bus for communication between different components (e.g., a CAN bus in automotive CPS). This allows the attacker to inspect other sensors' measurements before sending the spoofed measurements in order to maximize the impact on sensor fusion. At the same time, modern CPS usually operate in a time-triggered fashion such that each sensor transmits each measurement during a pre-allocated time slot; this effectively creates a schedule of sending measurements at each round. Thus, in this section we analyze how different schedules (based on sensor precisions) affect the attacker's impact (for different attack strategies) and compare these schedules in terms of the size of the resulting fusion interval.

Note that only the one-dimensional case is considered in this section since the concept of sensor precision (which is crucial when analyzing transmission schedules) does not extend to multiple dimensions in an obvious fashion (e.g., a sensor might be very precise in one dimension and very imprecise in another). In fact, the analysis presented in this section does not hold in multiple dimensions if sensor precision is measured by the volume of the fusion polyhedron. Thus, the multidimensional scheduling analysis for systems with different precision metrics (and possibly different measurement sets such as balls instead of polyhedra) is left for future work as well.

4.6.1 System Model

As noted above, in this section we assume a single-state system. We assume a linear bounded-noise system (similar to the one in Section 4.5):

$$x_{k+1} = a_k x_k + \nu_k^p, \tag{4.7}$$

where $x_k \in \mathbb{R}$ is the state, $a_k \in \mathbb{R}$ is the transition "matrix" and $\nu_k^p \in \mathbb{R}$ is bounded noise such that $|\nu_k^p| \leq M$, and M is a constant.

The observation model is once again grounded in the abstract sensor framework

– each sensor s_i provides a direct measurement of the state at time k of the form

$$y_{i,k} = x_k + \nu^m_{i,k}, \tag{4.8}$$

which is then converted to an interval, denoted by $I_{i,k}$ (note that we use the notation $I_{i,k}$ instead of $P_{i,k}$ in order to highlight the fact that each measurement is now an interval instead of a polyhedron). Note that, once again, in this setting non-attacked sensors are assumed to provide intervals that contain the true state.[6]

Finally, note that the time-triggered design of the system ensures that each sensor (attacked or not) sends its measurement during its allotted time slot at each time step.

4.6.2 Attack Model

This section focuses solely on stealthy attacks that are designed to disrupt system performance while remaining undetected. We define "disrupt system performance" as maximizing the size of the fusion interval – since larger fusion intervals mean higher uncertainty (potentially followed by more frequent emergency responses, such as system shutdowns, due to safety concerns), such an attack might have a severe effect on system performance. Thus, the attacker's goal is to maximize the size of the fusion interval while remaining undetected; the detection algorithm used by the system is a conservative attack detection algorithm, in which a sensor is declared attacked if its interval does not intersect the fusion interval, in which case it is guaranteed not to contain the true state (Chapter 6 presents a more sophisticated detection approach that also considers transient sensor faults).

Note that, similar to the rest of this chapter, we assume that the number of attacked sensors, f_a, is less than half of the total number of sensors.

[6] Extending the analysis presented in this section to the case with transient faults is left for future work. One of the main challenges with such an extension would be formulating a reasonable attack strategy (that also considers transient faults), which is one of the main contributions of this section.

Assumption. *We assume that the (assumed) upper bound on the number of attacked sensors is always larger than the actual number of attacked sensors, and that the number of attacked sensors f_a is less than half of all sensors, i.e.,*

$$f_a \leq f \leq \lceil n/2 \rceil - 1, \tag{4.9}$$

where n is the total number of sensors.

4.6.3 Problem Statement

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 problem considered in this section is the following:

Problem. *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 and over time? Find the schedule that minimizes this impact.*

4.6.4 Notation

The notation used in this section is the same as before, with some additions. In particular, let $\mathcal{N}_k$ denote all n intervals at time k, and let $S_{\mathcal{N}_k,f}$ denote the fusion interval given the set $\mathcal{N}_k$ and a fixed f, as before. The main difference is that instead of $P_{i,k}$ we now use $I_{i,k}$ to denote sensor i's interval at time k. Finally, let $l_{i,k}$ and $u_{i,k}$ be the lower and upper bound of sensor i's interval, respectively, such that $|I_{i,k}| = u_{i,k} - l_{i,k}$. Note that all time indices are dropped in Section 4.6.5, where only one round is considered in isolation.

4.6.5 Attack Strategy and Worst-Case Analysis

Note that the attack strategy, as stated in Section 4.6.2, is not fully specified. In particular, while it may be easy to see what is the best strategy from the attacker's point of view when the attacked sensors are last in the transmission schedule, selecting the best placement for the attacked intervals from other slots in the schedule is not trivial. Therefore, 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 subsection 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 4.6.6. 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 4.6.7.

As described in Section 4.6.2, the attacker has a goal, maximize the size of the fusion interval, and constraints, stay undetected. We now formalize the two, beginning with the latter.

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 section is to check whether each interval has a nonempty intersection with the fusion interval;[7] since the fusion interval is guaran-

[7] In Section 4.6.7, we use historical measurements to improve the system's detection capabilities.

teed to contain the true state, 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 state. We denote this intersection by Δ. Therefore, in passive mode the attacker must include Δ in his interval (any point that is not contained may be the true state) and has no restrictions on how to place the interval around Δ (if the interval is larger than Δ[8]).

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 Δ 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.

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 $a_1, \dots, a_{f_a}$

[8] Note that it cannot be smaller than Δ since Δ includes the intersection of all measurements of the corrupted sensors.

represent the attacked intervals:

$$\begin{aligned} \max_{a_1,\dots,a_{fa}} \quad & |S_{\mathcal{N},f}| \\ \text{s.t. } & S_{\mathcal{N},f} \cap a_i \neq \emptyset, \quad i = 1, \dots, f_a. \end{aligned} \tag{4.10}$$

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. *The attack strategy obtained as a solution to the optimization problem* (4.10) *(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 (4.10) 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 4.8, 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 I_1 and his own sensor's measurement but not of I_2. 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 I_1 ($a_1(1)$ in the figure) then I_2 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 possible placement of I_2.

While the attacker may be able to choose which sensors to attack, as argued in Chapter 1, certain sensors may not be compromised without detection or at all, with the resources available to the attacker. Thus, the attacker may not always ensure that his sensors would be last in the transmission schedule. Consequently, in cases such as the one in Figure 4.8, a reasonable strategy for the attacker is to maximize

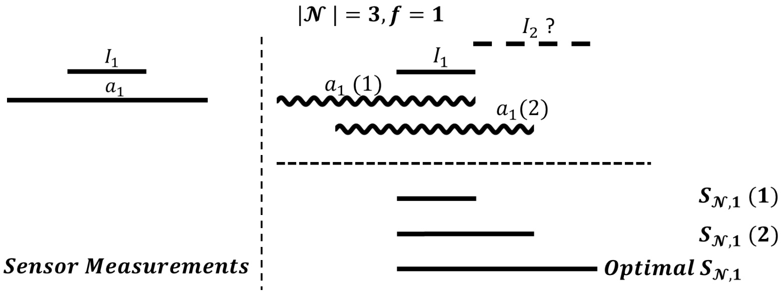


Figure 4.8: An example showing that if attacker (sinusoid) has not seen all intervals then he has no strategy that guarantees the fusion interval is maximized.

the expected size of the fusion interval. The expectation is computed over all possible placements of the **unseen** correct and compromised intervals.[9] Formally, for each compromised interval a_m (where m is an index in $\{1, \dots, f_a\}$) the attack strategy can be described with the following optimization problem

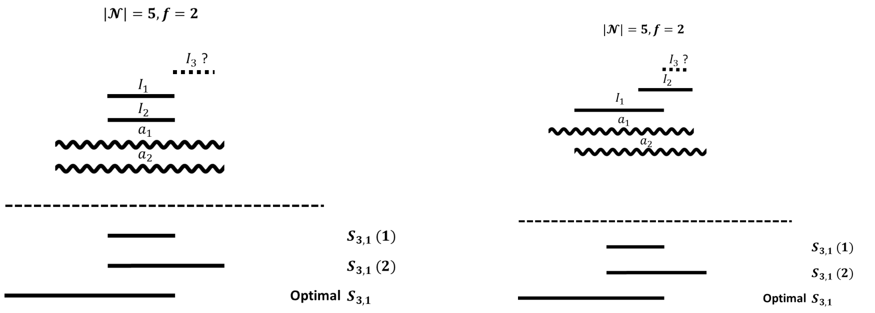
$$\begin{aligned} &\max_{a_m,\dots,a_{fa}} \ \mathop{\mathbb{E}}_{\mathcal{C}_m^R} |S_{\mathcal{N},f}| \\ &\text{s.t. } S_{\mathcal{N},f} \cap a_i \neq \emptyset \ \ i = m, \dots, f_a, \end{aligned} \tag{4.11}$$

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

As shown in Figure 4.8, 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 (4.11) 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

[9] To compute the expectation, the attacker is implicitly assuming intervals are uniformly distributed around Δ. If additional information is available about the distribution of sensor measurements, it can be incorporated in the optimization problem (4.11).



(a) Attacker has seen I_1 and I_2, while the unseen I_3 is small enough.

(b) Attacker has seen I_1 and I_2, while the unseen I_3 is small enough.

Figure 4.9: Examples of the two cases of Theorem 10. Attacked intervals are indicated by sinusoids.

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 10. *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) $\forall I_i, I_j \in \mathcal{C}^S, l_i = l_j, u_i = u_j$ and $\forall I_l \in \mathcal{C}^R$, $|I_l| \leq (|a_{min}| - |S_{\mathcal{C}^S \cup \Delta,0}|)/2$

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

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. Both cases are illustrated in Figure 4.9.*

Worst-Case Analysis

Given the attack strategy described above, 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 11. *Let I_{c_1} and I_{c_2} be the two largest-width correct sensors. Then $|S_{\mathcal{N},f}| \leq |I_{c_1}| + |I_{c_2}|$.*

Theorem 11 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.

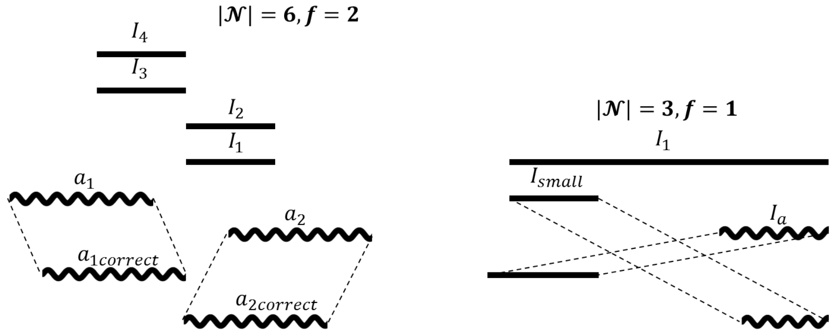
To formulate the theorems, we use the following notation. Let $\mathcal{L}$ be the set of predefined lengths of all intervals. We use S_{na} to denote the worst-case (largest width) fusion interval when no sensor is attacked. Similarly, let $S_{\mathcal{F}}$ be the worst-case fusion interval for a fixed 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 12. *If the f_a largest intervals are under attack, then $|S_{na}| = |S_{\mathcal{F}}|$.*

The theorem is illustrated in Figure 4.10a. 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 13. *$|S^{wc}_{f_a}|$ is achievable if the f_a smallest intervals are under attack.*

Figure 4.10b illustrates the theorem. The worst-case for the setup can be achieved when either I_a or I_{small} is attacked.



(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.

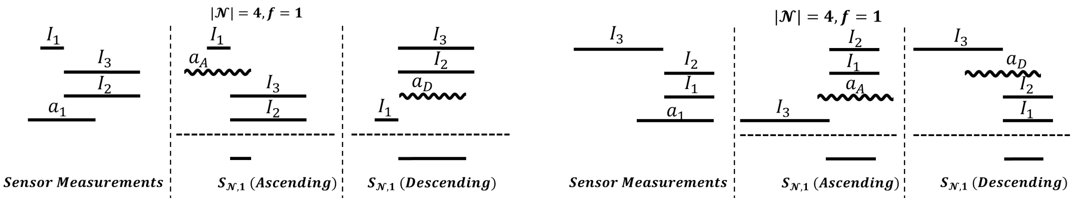
Figure 4.10: Illustrations of Theorems 12 and 13.

A few conclusions can be drawn from the worst-case results shown in this subsection. First of all, from Theorem 11, 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 12 and 13, 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.

4.6.6 One-Round Schedule Comparison and Analysis

In this subsection, we analyze the schedule design for communication over the shared bus in Figure 1.1. It builds on the analysis in Section 4.6.5 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. Thus, any investigated


$|\mathcal{N}| = 4, f = 1$
I_1
I_3
I_2
a_1
a_A
I_1
I_3
I_2
I_3
I_2
a_D
I_1
Sensor Measurements
$S_{\mathcal{N},1}$ (Ascending)
$S_{\mathcal{N},1}$ (Descending)
$|\mathcal{N}| = 4, f = 1$
I_3
I_2
I_1
a_1
I_2
I_1
a_A
I_3
I_3
a_D
I_2
I_1
Sensor Measurements
$S_{\mathcal{N},1}$ (Ascending)
$S_{\mathcal{N},1}$ (Descending)

(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 4.11: 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.

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.

We first note that neither schedule is better than the other in all scenarios. Figure 4.11 shows two examples in which different schedules are better, i.e., they produce smaller fusion intervals. In Figure 4.11a the fusion interval obtained with the Descending schedule is larger because the attacker is aware of the position of the largest interval. Figure 4.11b, 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 I_1 and I_2. Hence, if he is aware of I_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 I_1 and I_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 state (e.g., uniform over the 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,[10] 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[11] 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 sensor fusion algorithm.

Table 4.1 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 mixture of very precise sensors and very imprecise sensors. Hence, setups in Table 4.1 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

[10] 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.

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

Table 4.1: Comparison of the two sensor communication schedules. Subscript U denotes the uniform distribution, and N denotes the normal distribution.

	$\mathbb{E}_U \lvert S_{\mathcal{N},f} \rvert$ *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

system to compare impacts of communication schedules (based on sensors' precisions when no other information is available a priori) on the performance of attack-resilient sensor fusion.

To conclude this subsection, 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

Table 4.2: Comparison of the two attack strategies when Ascending schedule is used – AS_1 is the expectation optimization strategy; AS_2 is the constant offset strategy.

	$\mathbb{E}\,\lvert S_{\mathcal{N},f}\rvert$ *Asc.*, AS_1	$\mathbb{E}\,\lvert S_{\mathcal{N},f}\rvert$ *Asc.*, AS_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

the expected size of the fusion interval for each strategy for different setups. The results are shown in Table 4.2, 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.

4.6.7 Schedule Comparison Over Time

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.

In order to use historical measurements, we reintroduce some of the notation and terminology from Section 4.5. In particular, recall that we use the function m to

Table 4.3: Comparison of the two sensor communication schedules when historical measurements are used. Subscript U denotes the uniform distribution, and N denotes the normal distribution.

	$\mathbb{E}_U \lvert S_{p\_i} \rvert$ *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

map previous intervals to the current time:

$$m(I_{i,k}) = \{z \in \mathbb{R} \mid z = a_k p + q, \forall p \in I_{i,k}, |q| \leq M\}. \tag{4.12}$$

Furthermore, we use the *pairwise_intersect* method to map previous measurements to the current time; the fusion interval obtained from *pairwise_intersect* is then used in the following comparisons. We 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 4.1. The system dynamics were assumed to be $x_{k+1} = x_k + \nu_k^p$, with $|\nu_k^p| \leq 1$. Table 4.3 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 4.1, by adding history the system can further reduce the expected sizes for all setups, even when the attacker also has access to historical measurements.

4.6.8 Evaluation

To evaluate the sensor transmission scheduling technique proposed in this section, we illustrate how it can be implemented on an unmanned ground vehicle. We provide both simulation and experimental results using the LandShark.

Simulations

For our simulations, we used four of the LandShark's velocity sensors, namely the two wheel encoders, the GPS and the 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.

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. 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

Table 4.4: 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 9.5 mph	0%	16.8%	5.61%

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 $x_{k+1} = x_k + \nu_k^p$, with $|\nu_k^p| \leq 10$. The results are shown in Table 4.4. As can be seen, no critical violations were recorded under the Ascending Schedule, whereas the Descending and Random schedules both produced some.[12] In addition, adding history has greatly reduced the number of violations, both for the Descending and the Random schedules.

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 4.6.6, attack strategies can only be compared when the

[12] 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.

Table 4.5: 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

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 4.12 presents the results of the experiments.[13] 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 4.12 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 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 4.5 summarizes the results by providing the average size of the fusion interval for each scenario.

[13] A video with the experiments is available at *https://www.youtube.com/watch?v=C8jvo3xe5XU*.

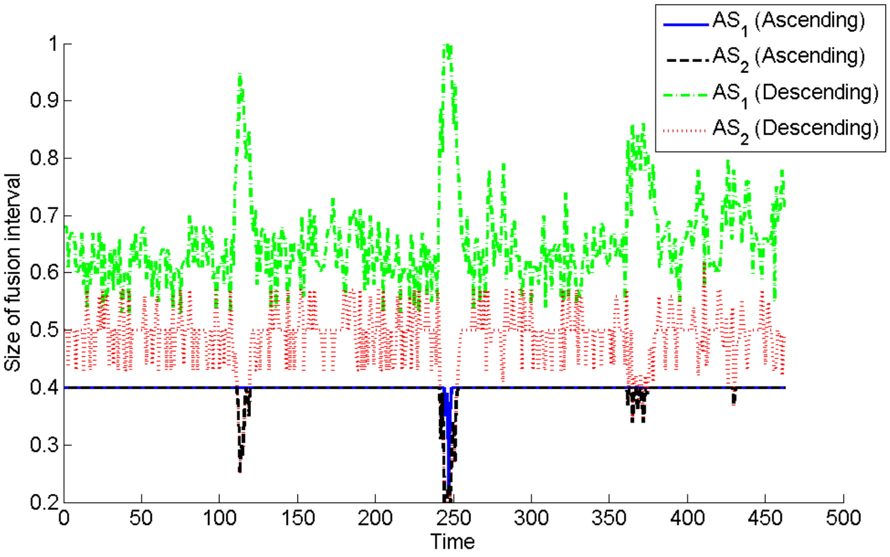


Figure 4.12: Comparison of the sizes of the fusion intervals as obtained with the two attack strategies, optimization (AS_1) and offset (AS_2), and two schedules.

4.7 Sensor Fusion in the Presence of Transient Faults

Having developed in the previos sections the sensor fusion framework for nominal systems, in this chapter we note that sensors sometimes experience faults during their operation and do not conform to the nominal observation model. Thus, the classical sensor fusion approach developed in the previous sections loses its worst-case guarantees since it might be possible for all sensors to experience transient faults at the same time. Therefore, in this section we develop a modified sensor fusion algorithm whose output is still guaranteed to contain the true value even in the case of transient faults.

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

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

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. We analyze both kinds of attacks and guard against their possible effects: in Chapter 6, we propose 1) a detector for the more dangerous, but easier to detect, kind of attacks, whereas in this section we develop 2) a modified sensor fusion algorithm whose output is guaranteed to contain the true state regardless of the manifestation of transient faults (or attacks that appear as transient faults).

In order to formalize the notion of a transient fault, we make use of sensor transient fault models (TFMs) that are now being provided by some manufacturers [82]. Such a model consists of three dimensions: (1) polyhedron size, (2) window size, and (3) number of allowed faulty measurements per window. At the same time, such specifications are not always available, so one of the contributions of this section is a method for selecting the three parameters based on observed training data. We illustrate this with a real-data case study using the LandShark.

Once a TFM is developed for each sensor, we propose a modified sensor fusion algorithm whose ouput, the filtered fusion polyhedron, is guaranteed to contain the true state even in the presence of transient fautls (or attacks that manifest as transient faults). The performance of the modified sensor fusion algorithm is evaluated using real data collected from the LandShark robot.

4.7.1 Problem Statement

In this section, we formalize the problem of sensor fusion in the presence of transient faults and emphasize the differences from the no-fault case.

System Model

Similar to the classical sensor fusion alorithm in Section 4.4, we note that the techniques developed here are independent of system dynamics, hence no assumptions on dynamics are made:

$$x_{k+1} = f(x_k, u_k) + \nu_k^p. \tag{4.13}$$

The nominal sensor model is also the same as in Section 4.4, i.e., each sensor i provides a direct measurement of the state at time k of the form

$$y_{i,k} = x_k + v_{i,k}, \tag{4.14}$$

which is then converted to a polyhedron $P_{i,k}$ such that

$$P_{i,k} = \{y_{i,k} + z \in \mathbb{R}^d \mid B_i z \leq b_i\}. \tag{4.15}$$

The main difference between the two models is that non-attacked sensors can now provide faulty measurements as well, i.e., they can experience transient faults.

Definition (Faulty measurement). *A sensor s_i provides a* faulty measurement $P_{i,k}$

at time k, if the true state is not included in the polyhedron, i.e.,

$$x_k \notin P_{i,k}.$$

The measurement is considered correct *otherwise.*

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 [82], we introduce the notion of a sensor's transient fault model (TFM). A TFM for a sensor s_i is a triple $(\mathcal{E}_i, e_i, w_i)$, where $\mathcal{E}_i$ represents the linear inequalities specifying the size of the polyhedron (i.e., the values B_i and b_i in (6.3)) 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. To relate the TFM to the original sensor fusion framework, in the conservative case the error bounds $\mathcal{E}_i$ would be specified large enough so that no faults are ever observed, i.e., $e_i = 0$ for any w_i. The TFM formulation, on the other hand, allows more flexibility by allowing tighter error bounds at the expense of declaring some sensor measurements "faults".

With this in mind, if s_i complies with its TFM, then any faulty measurements are deemed transient faults. Otherwise, it is *non-transiently faulty.*

Definition (Non-transiently faulty sensor)**.** *A sensor s_i is* non-transiently faulty *at time k if it has produced more than e_i faulty measurements in the last window of w_i measurements, i.e.,*

$$\left(\sum_{k'=k-w_i+1}^{k} F(i,k') \right) > e_i, \tag{4.16}$$

where $F(i,k) = 1$ if s_i outputs a faulty measurement at time k, and $F(i,t) = 0$, otherwise.

Attack Model

Note that formalizing attacks in a way that would distinguish them from faults is challenging. The reason for this is that 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. Thus, in order to scope the problem, we split our approach in two: 1) we develop in Chapter 6 a detector for attacks that manifest as the most disruptive kind of faults, namely non-transient faults; 2) for attacks that manifest as transient faults, we develop the modified sensor fusion algorithm presented in this section.

Thus, in this section we assume that all attacks that manifest as non-transient faults have been detected and discarded. The remaining attacked sensors are therefore assumed to comply to their transient fault models, i.e., all sensors are assumed to produce only a bounded number of faulty measurements. At the same, no assumptions about each individual faulty measurement is made (e.g., when it might occur or what the measurement might be).

Problem Statements

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

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

Once TFMs are available, we consider the problem of finding a bounded fusion polyhedron that is guaranteed to contain the true value at each time.

Problem. *Given a system with n sensors and a transient fault model $(\mathcal{E}_i, e_i, w_i)$ for each sensor, develop a sensor fusion algorithm that produces a fusion polyhedron that is guaranteed to contain the true value at each time.*

4.7.2 Transient Fault Model Parameter Selection

Before presenting the sensor fusion algorithm in the presence of transient faults, in this section we first outline 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 [82]. However, when the TFM of a sensor is not provided, it is necessary to identify the TFM parameters from empirical data. Note that we focus on the one-dimensional case only (i.e., each sensor provides an interval); the approach could be straightforwardly extended to d-rectangles by repeating the same procedure for all dimensions. At the same time, developing such parameters for multidimensional polyhedra is more challenging and is left for future work.

Due to the fact that it is used for worst-case analysis, the abstract model we obtain for each sensor must ensure that the sensor's interval contain the ground truth except in the case of a faulty measurement. In contrast, probabilistic sensor models construct a probability distribution of the sensor's possible measurements and are not naturally suited for worst-case analysis.[14] Thus, statistical approaches to parameter selection (e.g., the best-fit Poisson process) are unsuitable because they estimate parameters to maximally explain the data, without providing worst-case bounds. Therefore, we provide a new method for selecting the TFM parameters from empirical data. It is important to note that while the training data is assumed to contain no attacks, no assumptions are made about the presence of faulty measurements.

To empirically identify the TFM parameters, we apply the following procedure. First, we gather sensor measurements with known true state 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) (to simplify notation, the set of error bounds $\mathcal{E}$ is now

[14] Note that it might be possible to use probabilistic models for worst-case analysis by constructing bounds around all measurements with non-zero probability of occurring. This approach, however, reduces to the abstract model, hence we do not consider it in this dissertation.

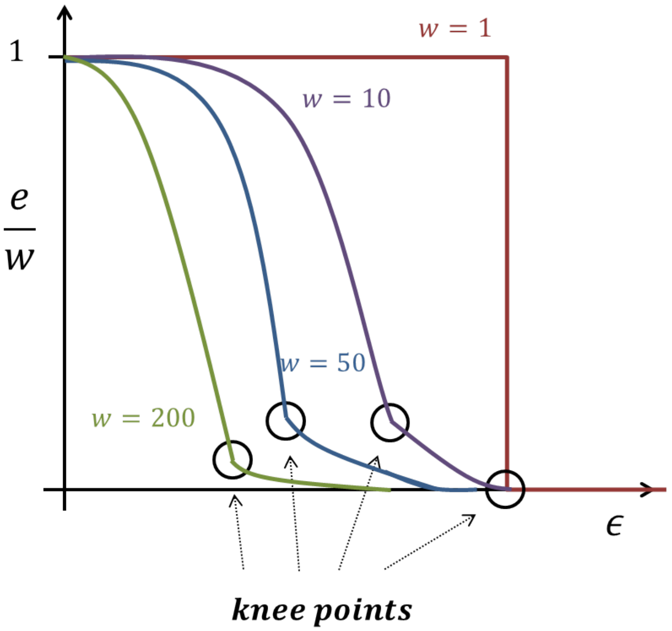


Figure 4.13: Sample plots of the proportion of faults in a window (e/w) against the error bound (ϵ).

merged into a single parameter ϵ, which denotes the size of the sensor's interval) 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 (Figure 4.13 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 increase. 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; at the same time, larger window sizes might result in greater uncertainty (i.e., larger fusion intervals) as historical measurements need to be mapped to the present time using system dynamics (with corresponding process noise). Section 4.7.4 provides a real-data evaluation of the process of obtaining a sensor's TFM.

4.7.3 The Filtered Fusion Polyhedron: A Modified Sensor Fusion Algorithm

In this subsection we describe an algorithm that outputs a *filtered fusion polyhedron* that is guaranteed to contain the true state and is bounded in size. The filtered fusion polyhedron can be thought of as the system's conservative, but correct, guess of its current state – since it does not trust its last fusion polyhedron, it examines the historical fusion polyhedra to improve this estimate.

We begin the analysis 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[15] provide faulty measurements in a single round or that all are correct in a single round. Therefore, f can now be considered as an input parameter to the fusion algorithm as opposed

[15] Only possible if all sensors have $e_i > 0$.

to a preliminary assumption. Note that if f is smaller than the actual number of faulty measurements per round, the resulting fusion polyhedron may not contain the true value.

The chosen value of f introduces a trade-off between accuracy and precision of the fusion polyhedron. In particular, decreasing f will result in a smaller (i.e., more precise) fusion polyhedron in any given round. On the other hand, it may increase the proportion of rounds where the fusion polyhedron 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 $(\mathcal{E}_i, e_i, w)$. Define a *global fault* as a round in which there are more than f faulty measurements. Recall that in such a case the fusion polyhedron is not guaranteed to contain the true value.

Definition (Global Fault)**.** *Given an upper bound f on the number of faulty measurements in a given round, a* global fault *occurs if more than f sensors provide faulty measurements in that round.*

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 values of $(E, W)_f$ depend 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' TFMs.

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 3 Computing the Global Fault Model of Sensor Fusion

```
Input: $n$ transient fault models of the form $(\mathcal{E}_i, e_i, w)$ and sensor fusion parameter $f$
$W_R \leftarrow w$
$E_S \leftarrow order\_descending(\bigcup e_i)$
$E \leftarrow 0$
while $W_R > 0$ and $E_S(f+1) > 0$ do
    for $\{i \leftarrow 1;\ i \leq f+1; i \leftarrow i+1\}$ do
        $E_S(i) \leftarrow E_S(i) - 1$
    end for
    $E_S \leftarrow order\_descending(E_S)$
    $W_R \leftarrow W_R - 1$
    $E \leftarrow E + 1$
end while
$W \leftarrow w$
return $(E, W)$
```

given the TFMs and f as input, outputs E and W. As formalized in Algorithm 3, 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 polyhedron 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 14. *The output, E, of Algorithm 3 is the largest number of global faults possible in a window of size W.*

Note that Algorithm 3 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 3 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 Figure 4.13 may be performed so that the best combination of accuracy vs. precision is chosen.

Algorithm 4 Filtered Fusion Polyhedron

```
Input: mapping function $m$, an array $FP$ containing $W$ fusion polyhedra (in chronological order) and a bound $E$ on the number of global faults
$FP_C \leftarrow \emptyset$
for $\{i \leftarrow 1;\ i \leq W-1; i \leftarrow i+1\}$ do
    $mapped\_P \leftarrow m(m(\cdots m(FP(i))))$ // map $i$ times
    $FP_C.add(mapped\_P)$
end for
$FP_C.add(FP(W))$
return $sensor\_fusion(FP_C, E)$
```

Using the intuition of Algorithm 3 (i.e., mapping historical sensor measurements to the current time and arguing about how many of them could be faulty in the worst case), it is now possible to derive a bounded fusion polyhedron that is guaranteed to contain the true value. To do this, we once again make use of the function m that maps past polyhedra to the current time:

$$m(P_{i,k}) = \{z \in \mathbb{R}^d \mid z = Ap + q, \forall p \in P_{i,k}, \|q\| \leq M\}.$$

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

The proposed algorithm is formalized in Algorithm 4. Essentially, all fusion polyhedra are mapped, using m, to the current time k, thus obtaining W polyhedra at k. Then we apply the original sensor fusion algorithm – since at most E mapped polyhedra are faulty, we output the smallest polyhedron that contains all points that lie in at least $W - E$ mapped polyhedra. Thus, a filtered fusion polyhedron is computed that is a conservative, but bounded, estimate of the system's current state. Note that Algorithm 4 is polynomial both in the number of sensors and the window size.

Proposition 10. *The complexity of Algorithm 4 is $O(CW^2 + W \log W)$ where C is the cost of the mapping function m. Constructing the input array FP is*

$O((n \log n)W)$ *where* n *is the number of sensors.*

It is important to note that Algorithm 4 can be computed even more efficiently by noting that when calculating the filtered fusion polyhedron at a given round, we can reuse the result of the calculations of the previous round, i.e., only one round of polyhedron mappings needs to be performed.

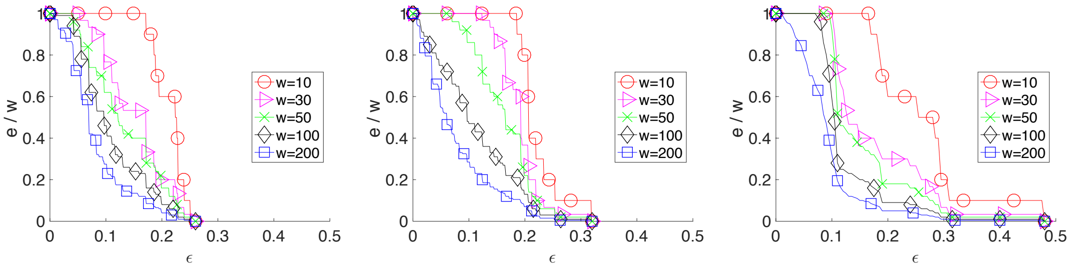
We note that Algorithm 4 does not always produce the smallest possible polyhedron that is guaranteed to contain the true value. On other hand, as shown in the above Proposition, it is efficient and can be implemented in real time whereas it is difficult to obtain an algorithm that outputs such a polyhedron and is not exponential in the number of sensors and rounds. Finally, Algorithm 4's output is guaranteed to contain the true state and is bounded (provided $E < \lceil W/2 \rceil$), so it is still in the spirit of worst-case analysis.

4.7.4 Evaluation

In this section, we evaluate the performance of the modified sensor fusion algorithm as well as the selection of the TFM parameters through a case study on the LandShark. For this case study, three of the LandShark's velocity sensors were used – the two wheel encoders and the GPS. 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 (i.e., one-dimensional intervals) to evaluate the proposed techniques in the presence of transient faults (e.g., tire slip).

Transient Fault Model Parameter Selection

This subsection illustrates the selection of the TFM parameters following the method described in Section 4.7.2. First, we collect the training data by driving the LandShark straight at a constant speed of 1 m/s on different surfaces such as grass, asphalt and snow, where the environment may cause transient faults (e.g., slipping



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

Figure 4.14: Empirical plots of the proportion of faults in a window (e/w) against the error bound (ϵ).

Table 4.6: Transient fault models for the sensors on the LandShark.

Window Size	L. Encoder		R. Encoder		GPS	
	ϵ	e	ϵ	e	ϵ	e
$w = 1$	0.26	n.a.	0.32	n.a.	0.48	n.a.
$w = 10$	0.229	2	0.234	2	0.295	2
$w = 30$	0.195	6	0.207	6	0.19	9
$w = 50$	0.195	11	0.199	11	0.19	9
$w = 100$	0.131	26	0.168	22	0.19	9
$w = 200$	0.117	36	0.126	37	0.19	10

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 Figure 4.14, which is the real-data equivalent of Figure 4.13.

Table 4.6 summarizes the chosen parameters, where the window size w is varied between 10, 30, 50, 100 and 200. For example, for $w = 50$ in GPS (Figure 4.14c), 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 Figure 4.14, 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 $w = 1$. 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., with $w = 200$, the interval sizes are more than twice smaller than those with $w = 1$).

Evaluation of Filtered Fusion Polyhedron

To evaluate the usefulness of the filtered fusion polyhedron (note that it is just a one-dimensional interval in this case study), we once again use three of the LandShark velocity sensors, namely the two encoders and GPS. As discussed in Section 4.7.3, there exists a trade-off between the precision and the accuracy of the fusion polyhedron depending on the choice of f. Thus, we evaluate these metrics using the LandShark data; note that we use the same TFM parameters as the ones shown in Table 4.6.

To do this, we proceed as follows: we first collect data from 17 runs of the LandShark, each consisting of about 500 velocity measurements by each sensor at 10 Hz. The true state is obtained in the same way as in TFM case study. Varying f between 0 and 1,[16] we perform sensor fusion at each round and check whether the fusion interval contains the true state (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 3 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 4.7 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

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

Table 4.7: Sensor fusion performance for different f. E ($\hat{E}$) is the theoretical (empirical) worst-case number of rounds with global faults.

Window Size	$f = 0$		$f = 1$	
	E	$\hat{E}$	E	$\hat{E}$
$w = 10$	6 (60%)	6 (60%)	3 (30%)	2 (20%)
$w = 30$	21 (70%)	9 (30%)	10 (33%)	3 (10%)
$w = 50$	31 (62%)	9 (18%)	15 (30%)	3 (6%)
$w = 100$	57 (57%)	36 (36%)	28 (28%)	8 (8%)
$w = 200$	83 (42%)	68 (34%)	41 (21%)	27 (14%)

Table 4.8: Average size of filtered fusion interval for different values for f and noise bound M.

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

becomes less tight. Furthermore, as f increases, 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 (e/w) ratio (resulting in better accuracy).

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 4.8 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 win-

Table 4.9: Average running time (in microsecond) of filtered fusion for different values for f and noise bound M.

Window Size	$f = 0$		$f = 1$	
	$M = 0.005$	$M = 0.001$	$M = 0.005$	$M = 0.001$
$w = 10$	33	35	34	35
$w = 30$	38	39	44	43
$w = 50$	43	47	43	48
$w = 100$	57	54	50	56
$w = 200$	72	87	72	86

dows 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.

Lastly, to analyze the time overhead of Algorithm 4 which calculates the filtered fusion interval, we measured the average running time of one round of the filter fusion for the different setups.[17] Table 4.9 shows that although the running time increases with the window size W, the time overhead is negligible overall (considering that the sensors' sampling frequency is 10Hz).

[17] The running time was measured on a machine with a 2.8 GHz Intel Core i7 processor.

Chapter 5

Context-Aware Sensor Fusion

Having developed the sensor fusion framework for safety detection with guarantees in Chapter 4, in this chapter, we present an approach for incorporating context measurements in order to further strengthen the guarantees of sensor fusion. As argued extensively in Chapters 1 and 3, context measurements are high-level representations of data collected from the system's environment sensors; as such, these discrete measurements can be used to provide (rough) information about the system state as well. Similar to the context-aware estimation case, context measurements can be incorporated into the sensor fusion framework as well. Context measurements are especially useful in scenarios where standard continuous sensors might be faulty or attacked (e.g., in a scenario with a perfectly attackable system that is unable to detect an attack on its continuous sensors [147]).

Similar to Chapter 3, in this chapter we focus on binary measurements as a rich subclass of all context measurements. In particular, each measurement $y^b_{i,k}$ is equal to 1 if a context element is detected and -1 otherwise. As argued in Chapter 3, examples of binary measurements include building detection scenarios (implying that the system must be close to the detected building) as well as threshold medical alarms (meaning that the patient might be in a critical state).

Contrary to the nominal case discussed in Chapter 3, where we model context measurements in a probabilistic setting, in this framework we employ worst-case analysis since we would like to provide safety guarantees about the system's state. Specifically, when a context measurement of 1 is received, it is mapped to a bounded set of possible values of the system state. Similar to the attack-resilient sensor fusion setting, we assume each set is a bounded polyhedron. Modeling context measurements as polyhedra captures a wide class of context measurements. For example, if a building is detected using image processing, a polyhedron (e.g., a trapezoid) in front of the building could be constructed in order to indicate the system can only detect the building from within that set. Similarly, if a radio beacon is detected, a rectangle around the beacon is constructed so as to indicate the system is close to that beacon.

Given this interpretation of context measurements, it is now possible to extend the continuous-sensor fusion algorithm (Algorithm 1) to the context-aware case. Note that context measurements can also be faulty/attacked, similar to continuous measurements. Thus, once again we make the same assumption that less than half of all sensors (continuous and binary) are attacked at any given time. With this in mind, the resulting algorithm is similar to the original sensor fusion algorithm, which essentially constructs all sets that might contain the true state.

We evaluate the context-aware sensor fusion algorithm in a case-study simulation using the LandShark robot. In this scenario, the LandShark only has access to GPS for estimation and control purposes. At the same time, an undetectable attack is performed on GPS such that the system believes it is safe when it is in fact heading towards an obstacle. By incorporating context measurements obtained from nearby buildings, however, the system can detect it is in an unsafe state and apply an emergency control response (e.g., shutdown).

5.1 Problem Statement

This section formulates the system and attack models used in this chapter before formulating the context-aware sensor fusion problem statement.

5.1.1 System Model

The system model is the same as the standard sensor fusion model, with the important addition of context measurements. In particular, no assumptions are made about system dynamics (since the technique is applied independently at each time step):

$$x_{k+1} = f(x_k, u_k) + \nu_k^p. \tag{5.1}$$

The nominal plant sensor model is also the standard abstract sensor model, i.e., each plant sensor s_i provides a direct measurement of the state at time k of the form

$$y_{i,k}^c = x_k + \nu_{i,k}^m, \tag{5.2}$$

which is then converted to a polyhedron $P_{i,k}$ such that

$$P_{i,k} = \{y_{i,k}^c + z \in \mathbb{R}^d \mid B_i z \leq b_i\}. \tag{5.3}$$

In addition to continuous plant measurements, we also consider context measurements in this chapter. As noted in the introduction of this chapter, we focus on binary context measurements, such that each measurement $y_{i,k}^b$ is either 1 or -1 depending on whether a context element is detected. When a measurement of 1 is received, the measurement is mapped to a bounded polyhedron $Q_{i,k}$ similar to the continuous abstract sensor model, i.e.,

$$\begin{aligned} y_{i,k}^b &= 1 \Rightarrow x_k \in Q_{i,k} \\ y_{i,k}^b &= -1 \Rightarrow x_k \in \bar{Q}_{i,k}, \end{aligned} \tag{5.4}$$

where

$$Q_{i,k} = \{z \in \mathbb{R}^n \mid D_i z \leq d_i\},$$

for some matrix D_i and vector d_i, and $\bar{Q}_{i,k}$ is the complement of $Q_{i,k}$. As mentioned above, such polyhedra can capture a wide variety of context detections, e.g., a building recognition scenario.

In this chapter, we develop approaches for the nominal sensor fusion framework, i.e., we assume that all non-attacked sensors (both continuous and binary) provide correct measurements. We leave context-aware sensor fusion with transient faults for future work.

5.1.2 Attack Model

The attack model for continuous sensors is also the same as in the standard sensor fusion setting, i.e., if a sensor is attacked then no assumptions are made about what measurements it sends. The attack model for context measurements has several aspects, as described below.

There are two ways in which an attack on a context measurement can manifest – it can indicate the state is in $Q_{i,k}$ when it is not (i.e., a false positive) or it can fail to indicate the state is in $Q_{i,k}$ when it is in fact there (i.e., a false negative). In this chapter, we focus on false positives only – in fact, as will become apparent in Section 5.2, the context-aware sensor fusion algorithm only considers positive context measurements in order to keep the worst-cases guarantees of sensor fusion. Thus, an attacked context measurement is assumed to provide false positives only (thereby wrongly implying the state is in the corresponding set $Q_{i,k}$). Incorporating false negatives in the fusion algorithm is left for future work (as discussed in Chapter 7).

Similar to the standard sensor fusion setup, in order to be able to provide worst-case guarantees, we assume that less than half of all sensors (continuous and binary) are attacked.

5.1.3 Problem Statement

Given the above system and attack models, respectively, the context-aware sensor fusion problem is as follows.

Problem. *Given a system with n sensors (both continuous and binary), the context-aware sensor fusion problem is how to obtain a fusion polyhedron in a single time step, i.e., a minimal-volume polyhedron that is guaranteed to contain the true state.*

5.2 Approach

Given the system and attack models in Section 5.1, the context-aware sensor fusion procedure is similar to the standard sensor fusion algorithm presented in Algorithm 1. The modified approach is shown in Algorithm 5. Let P denote the set of all n measurements, with f again denoting an upper-bound on the number of corrupted measurements (i.e., not containing the true value). Let there be n_c polyhedra from continuous sensors in P and n_b polyhedra from context measurements such that $n = n_c + n_b$. In order to find a fusion polyhedron that is guaranteed to contain the true value and is minimal in size, we find all intersections of $n - f$ measurements and take their convex hull.

Thus, the fusion polyhedron is once again the minimal polyhedron that is guaranteed to contain the true value. Note that no comparison can be made with the fusion polyhedron using only the continuous sensors because even computing such a fusion polyhedron would require assuming an upper bound on the number of attacked continuous sensors. The benefit of adding context measurements is that sensor fusion can provide worst-case guarantees in the presence of more (or all) continuous sensor attacks than in the no-context setting as long as a sufficient number of correct

Algorithm 5 Context-Aware Sensor Fusion Algorithm

```
Input: An array of sets $P$ of size $n$ ($n = n_c + n_b$) and an upper bound on the number
    of corrupted measurements $f$
$C \leftarrow$ combinations_n_choose_n_minus_f($P$)
$R_{\mathcal{N}_k,f} \leftarrow \emptyset$
for each $K$ in $C$ do
    add($R_{\mathcal{N}_k,f}$, intersection($K$))
end for
return conv($R_{\mathcal{N}_k,f}$)
```

context sensors are used (such that the total number of attacked sensors is less than half).

Note that Algorithm 5 makes no use of the sets $\bar{Q}_{i,k}$, i.e., it ignores context measurements when they are not received. The reason for this is that if these sets are incorporated in the algorithm, it would not be possible to provide a bounded polyhedron in the worst-case because the sets $\bar{Q}_{i,k}$ are unbounded (as complements of the bounded $Q_{i,k}$ sets). Thus, although some information is lost while ignoring context measurements of -1, we do so at the benefit of being able to maintain our worst-case guarantees. Providing guarantees while incorporating negative context measurements is left for future work.

5.3 Case-Study Evaluation

This section presents a case-study evaluation of the context-aware sensor fusion technique. We simulate an example of a so called perfectly attackable system [147], i.e., a system whose continuous sensor is under attack such that the estimation error can grow unbounded but the system cannot detect the attack. We show that the system can use context measurements in such a case in order to detect when it is in an unsafe state.

In particular, we simulate a scenario in which the LandShark is moving in an urban environment and trying to avoid an obstacle. The entire scenario is shown in

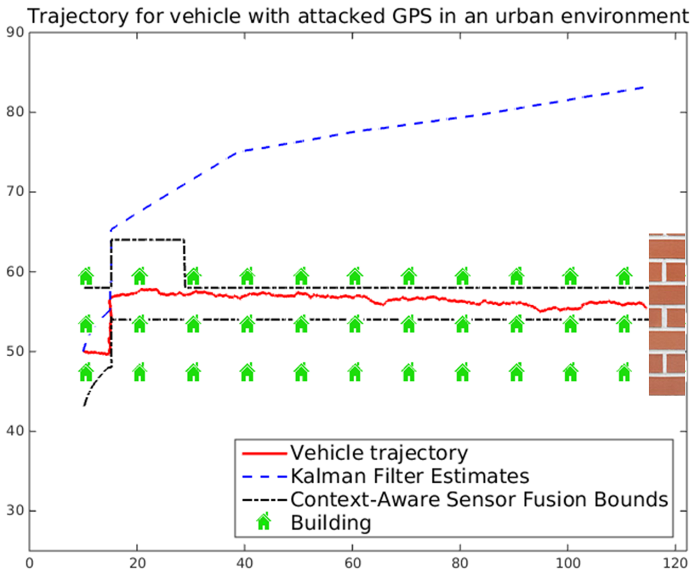


Figure 5.1: Perfectly attackable system using context measurements. Kalman filter estimates lead the system to believe it is safe whereas the context-aware sensor fusion bounds indicate that the system is unsafe.

Figure 5.1; the LandShark tries to avoid the wall on the East side, so it initially goes North until it believes it is safe. However, the LandShark's only position sensor, a GPS, is attacked such that the Kalman filter estimates fool the system in believing it is safe – the GPS attacks are also carried out in such a way so as to avoid detection by standard anomaly detectors (e.g., a chi-squared detector [147]). As a result, the system starts heading East too early and crashes into the wall.

On the other hand, we note that since the LandShark is going through an urban environment, it can use image processing to recognize nearby buildings and obtain context measurements from them. Thus, for each building on the map, the LandShark receives a context measurement (in the form of a square around the building) if it is in the proximity of that building. At each time step, all context measurements are used together with the GPS measurement in the context-aware sensor fusion algorithm; the upper and lower bounds of the resulting fusion polyhedron are also shown in Figure 5.1. As can be seen in the figure, the fusion polyhedron always contains the true value and, more importantly, indicates that the system is not safe, i.e., it is not North of the obstacle. Thus, this is an example in which the system can greatly benefit from context measurements and can avoid the obstacle (e.g., by going North until the fusion polyhedron contains no points that are inside the obstacle).

Chapter 6

Attack Detection in the Presence of Transient Sensor Faults

Having developed multiple techniques for estimation and safety detection in the previous chapters, in this chapter we note that all these techniques rely on sensors providing accurate information. In particular, although the sensor fusion approaches are indeed robust to attacks on half of the system's sensors, their performance could be improved if attacked sensors are identified and discarded. Thus, in this chapter we provide a general technique for sensor attack detection and identification.

One of the main requirements of such a detection algorithm is that it accounts for the fact that sensors might sometimes provide faulty measurements. As argued in Chapter 4, sensors often experience transient faults that usually do not last long and recover on their own (e.g., GPS losing connection in a tunnel and regaining it afterwards); thus, one can design controllers that are robust to such scenarios. Since transient faults are not a security threat for CPS, in this chapter we develop an attack detection algorithm that does not raise false alarms due to temporarily wrong sensor measurements and instead only flags actual sensor attacks.

As argued in Chapters 1 and 2, standard detection techniques either assume 1) the system is in a known nominal state, i.e., known initial condition, such that a change detection approach could be employed [25, 52, 94, 130, 138, 221] or 2) that a specific fault/attack is present such that specific detectors for that fault/attack could be developed, e.g., generalized likelihood ratio tests or sequential probability ratio tests [23, 24, 177, 178]. At the same time, these two assumptions are not justified in modern CPS that may never have a known nominal state (e.g., perfectly attackable systems [147]) and may not know in advance what faults/attacks they might experience.

Redundancy-based approaches eliminate the need for the above unrealistic assumptions by adding more sensors and assuming that less than half are under attack [78, 211]. A major shortcoming of existing redundancy-based attack detection works [109, 136] is that they conservatively treat transient faults as attacks. While there exist papers distinguishing attacks from faults [21], they make specific assumptions about the form or direction of faults/attacks, thus being unsuitable for our problem.

Different from existing works, in this dissertation we address the problem of sensor attack detection in the presence of transient sensor faults. Similar to Chapter 4, we use the abstract sensor model (in which each sensor provides a polyhedron) – this model is well suited for worst-case analysis due to the noise bounds it provides. In order to distinguish between attacks and faults, similar to Chapter 4, we make use of sensor transient fault models (TFMs) that are now being provided by some manufacturers [82]. Such a model consists of three dimensions: (1) polyhedron size, (2) window size, and (3) number of allowed faulty measurements per window. In the case when such TFMs are not available, one may refer to Section 4.7.2 for an approach to obtain such models from sensor data.

As noted in Section 4.7, depending on the attacker's goals and capabilities sensor attacks can manifest as either transient or non-transient faults. 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 dissertation, we provide resilience to both kinds of attacks: in this section, we present 1) a detector for attacks that manifest as non-transient faults, whereas in Section 4.7 we developed 2) a modified sensor fusion algorithm whose output is guaranteed to contain the true state even in the presence of sensor attacks that might appear as transient faults.

In order to develop a detector for attacks that manifest as non-transient sensor faults, we assume there exists a TFM for each sensor and propose a detection and identification algorithm for sensors that do not comply with their TFMs. 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 evaluate the performance of the detection algorithm (in terms of false alarm and detection rates) using real data collected from the LandShark robot. In particular, we collected measurement data from several runs of the LandShark; this data was then retrospectively augmented with several kinds of attacks. The proposed detector (configured with a large enough TFM window) is able to eventually detect all sensor attacks, thus illustrating the usefulness of this approach.

6.1 Problem Statement

This section presents the system and attack models considered in this chapter before formulating the problem of sensor attack detection in the presence of transient faults.

6.1.1 System Model

The system model used in this Section is the general abstract sensor model used in Chapter 4. Similar to Chapter 4, we note that the techniques developed here are

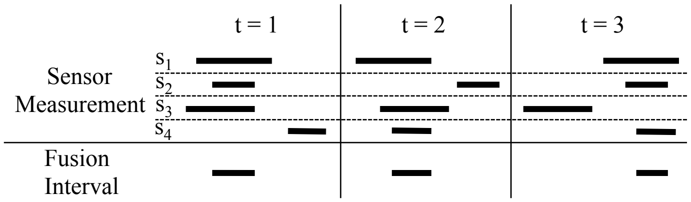


Figure 6.1: Illustration of the benefit of the transient fault model. Each of s_2, s_3 and s_4 provide one faulty measurement, but their other measurements are correct.

independent of system dynamics, hence no assumptions on dynamics are made:

$$x_{k+1} = f(x_k, u_k) + \nu_k^p. \tag{6.1}$$

The sensor model is also the same model as in Chapter 4, where each sensor s_i provides a direct measurement of the state at time k of the form[1]

$$y_{i,k} = x_k + \nu_{i,k}^m, \tag{6.2}$$

which is then converted to a polyhedron $P_{i,k}$ such that

$$P_{i,k} = \{y_{i,k} + z \in \mathbb{R}^d \mid B_i z \leq b_i\}. \tag{6.3}$$

In addition, similar to Section 4.7, each sensor has a corresponding TFM $(\mathcal{E}_i, e_i, w_i)$ that specifies an upper bound e_i on the number of faulty measurements in any window of size w_i. For sensor s_i, the set $\mathcal{E}_i$ contains the pair (B_i, b_i) that specifies the shape of the corresponding polyhedron $P_{i,k}$.

To illustrate the benefit of the TFM, consider Figure 6.1. If one were to treat all transient faults as attacks, then each of s_2, s_3 and s_4 would be declared as attacked because they each produce a faulty measurement in rounds 3, 2, and 1, respectively (these faulty measurements can be detected because they do not overlap with the

[1] Note that measurements are not explicitly treated as continuous or binary in this chapter since the technique treats them in the same way, i.e., by considering their corresponding polyhedra.

fusion interval at the respective times); however, it is more beneficial for the system to just discard the faulty measurements and continue the use the sensors at the times when they do provide correct measurements.

6.1.2 Attack Model

As mentioned to the introduction of this chapter, we focus on detecting attacks that manifest as non-transient sensor faults, i.e., the attacked sensor measurements do not conform to their corresponding TFMs. Thus, in this chapter we treat all non-transiently faulty sensors as attacked (even if an alarm is raised due to an actual non-transient fault, we argue that this is not a false alarm since such a sensor might compromise the system's operation).

Definition (Attacked Sensor)**.** *A sensor is considered* attacked *if it is non-transiently faulty.*

Once again, we emphasize that attacks that manifest as transient faults are handled in Section 4.7, where we developed a sensor fusion algorithm that provides guarantees even in the presence of attacks that manifest as transient faults.

Finally, no assumptions are made on the number of attacked sensors. As long as there is one non-attacked sensor in the system, attack detection is possible. Stronger assumptions are needed for attack identification, as noted in the following sections.

6.1.3 Problem Statements

The problem addressed in this chapter is sensor attack detection in the presence of transient sensor faults.

Problem. *Given a system with n sensors and a transient fault model $(\mathcal{E}_i, e_i, w_i)$ for each sensor, develop an algorithm to detect the existence of an attacked sensor and possibly identify which sensor is under attack.*

6.2 A Sound Algorithm for Attack Detection and Identification

In this Section we describe our approach to sensor attack detection and identification, which aims to differentiate sensor attacks from mere transient faults given each sensor's TFM. This section assumes that a TFM has already been identified for each sensor; one way of developing such models is presented in Section 4.7.2.

The detection algorithm developed in this chapter 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.* At a high level, we accumulate information about inconsistencies between sensor measurements over time and 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.

6.2.1 Weak and Strong Inconsistency

As usual, this section is built on the premise that the true state 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. Two sensors are weakly inconsistent at a given time if one of them provides a faulty measurement.

Definition (Weak Inconsistency)**.** *We say that sensors s_i and s_j are weakly inconsistent at time k if one of them provides a faulty measurement at time k.*

Since weak inconsistency is defined upon the unknown true state, 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 intervals. This condition is formally stated in the following lemma:

Lemma 3. *If two sensors, s_i and s_j, provide polyhedra that do not overlap at time k, i.e.,*

$$P_{i,k} \cap P_{j,k} = \emptyset,$$

then at least one of the two sensors provided a faulty measurement at time k.

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. 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 (Strong Inconsistency)**.** *We say that sensors s_i and s_j are strongly inconsistent at time k if one of them is non-transiently faulty at time k.*

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

Lemma 4. *Two sensors, s_i and s_j, are strongly inconsistent at time k if the following condition is true:*

$$\left(\sum_{k'=k-\min(w_i,w_j)+1}^{k'=k} WI(i,j,k') \right) > e_i + e_j \tag{6.4}$$

where $WI(i,j,k) = 1$ if s_i and s_j are weakly inconsistent at time k, and $WI(i,j,k) = 0$ otherwise.

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.

6.2.2 Attack Detection and Identification

In this subsection, we describe our approach to attack detection and 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. To propagate the strong inconsistencies over time, we use a sequential detection approach (motivated by sequential detection theory [212]) and accumulate the information over time. These statements are formalized in the remainder of this subsection.

Theorem 15. *If two sensors, s_i and s_j, are strongly inconsistent at any time k, then one of them must be attacked.*

Theorem 15 is the main result of this chapter. It says that the existence of a strong inconsistency between two sensors is sufficient for the existence of an attack. Thus, the attack detection algorithm developed in this chapter works by detecting strong inconsistencies between sensors using the sufficient condition in Lemma 4.

As the existence of strong inconsistency between two sensors cannot determine which sensor is attacked, we now address the attack identification problem. 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, or vice versa. For this reason, when $a \geq n-1$, there can exist no detector which correctly identifies attacks based on pairwise comparisons alone.

When $a < n-1$, there is a sufficient condition for identifying attacked sensors.

Theorem 16. *Assume $a < n - 1$ and let $d(i)$ denote the number of sensors that have been strongly inconsistent with s_i during the system's operation. Then, s_i can be identified as attacked if $d(i) > a$.*

Next, 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 5. *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.*

Finally, it is important to emphasize the soundness of the developed attack detection/identification approach.

Proposition 11. *The attack detection/identification methods proposed in Theorems 15 and 16 are sound. In other words, the algorithms raise no false alarm (assuming the transient fault model parameters are correctly specified).*

6.3 Case Study

In this section, we evaluate the performance of the attack detection/identification algorithm. For this case study, the same data as in Section 4.7.4 was used, i.e., three of the LandShark's velocity sensors were used – the two wheel encoders and the GPS. The gathered data corresponds to 2400 velocity measurements by each sensor at 10 Hz (i.e., about four minutes). The LandShark was driven on different surfaces (namely, grass, asphalt and snow) such that different types of faults might occur.

Thus, we use the redundancy of velocity measurements (i.e., one-dimensional intervals) to evaluate the proposed techniques in the presence of transient faults (e.g., tire slip).

Table 6.1: False alarm rate

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

6.3.1 Attack Detection Performance

To evaluate the performance of the attack detectors, we use the TFM parameters obtained in Section 4.7.4 and employ various attack scenarios as explained below.

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 6.1 shows the false alarm rates for the TFM parameters of Table 4.6; note that we use PI_w to indicate the proposed Pairwise-Inconsistency-based approach using a window of length w.[2] 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 (note that PI_1 is referred to as the sensor fusion (SF)-based detector since the interval sizes would be conservatively set large enough so that no faulty measurements are observed). 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 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

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

Table 6.2: 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

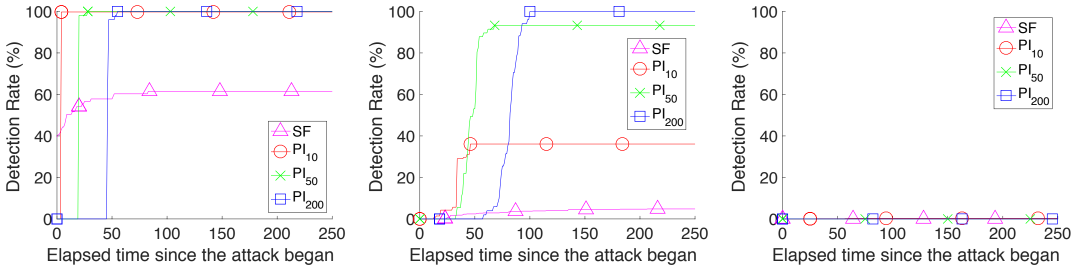
of 0.8 m/s to the attacked sensor. The random attack adds a uniformly distributed random noise between 0 and 0.8 m/s.[3]. The greedy attack replaces the measurement of the attacked sensor with a specially crafted measurement designed to maximize the uncertainty (i.e., the fusion interval size) in the system; this is also the stealthy attack discussed in Section 4.6.[4] 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 6.2 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 Figure 6.2, where the steady-state detection rates correspond to the detection rates in Table 6.2. Figure 6.2c shows that all detectors rarely detect any greedy attacks. From the cases

[3] 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).

[4] 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.

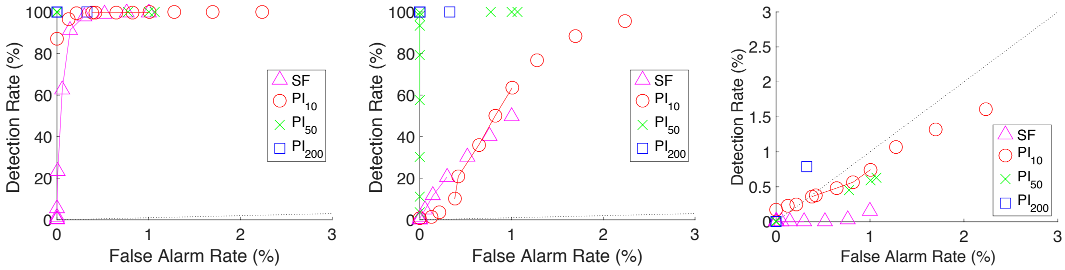
Figure 6.2: Time to detection plots under the three classes of attacks.

of biased and random attacks, Figure 6.2 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 4.7.4. 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 Figure 6.3, 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.[5]

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 [212]. 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 [212]. Therefore, the robustness of the PI-based detectors improves with window size in general. Note that PI_{10} performs marginally better than the

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



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

Figure 6.3: 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.

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 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 almost identical results 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.

Chapter 7

Conclusion

In conclusion, this dissertation addressed the problem of providing detection and estimation techniques in order to ensure the safety and security of modern CPS. In addition, all of these techniques provide guarantees about their performance, in expectation or in the worst case depending on the application. Our main contribution lies in the generality of the proposed approaches – while most existing works address safety/security problems by making unreasonable assumptions (either the system is in a known nominal state or the class of fault/attack is known), we make use of sensor redundancy and context information in order to avoid making such assumptions. In summary, we make contributions to three fields of this problem space: 1) nominal context-aware estimation, 2) safety detection and 3) sensor attack detection.

In Chapter 3, we note that incorporating context information for the purposes of estimation and detection is a novel idea in itself. Context measurements can be extracted from the system's environment data and are essentially high-level representations of low-level measurements (e.g., a recognized building in an image). Thus, they can be used for state estimation purposes, in addition to (or in lieu of) standard linear continuous measurements. In Chapter 3, we model context measurements probabilistically (i.e., each measurement has a known probability of occurring given the system's state) and develop a context-aware filter using two different types of measurement models. In particular, the specific contributions of Chapter 3 are:

- Formulation of the context-aware filtering problem for linear systems. Two classes of probability of context detection functions were investigated, namely inverse-exponential functions and sigmoid functions.
- Development of a Gaussian-Mixture-based filter and a sigmoid-based filter using the two proposed classes of probability of detection functions, respectively.
- Asymptotic analysis of the sigmoid-based filter. We provided sufficient conditions on the number of available context measurements under which the filter's uncertainty is bounded. In addition, we argued that the filter converges to a Newton Method in the limit (after repeated updates), thus providing evidence that it is likely asymptotically unbiased.
- Evaluation in simulation of both filters. Different features of the filters were explored in multiple case studies. The probit-based filter's evaluation also suggests the filter is likely asymptotically unbiased.
- Real-patient data evaluation of the sigmoid-based filter. We applied the filter to the problem of estimation of blood oxygen content using non-invasive measurements only.

Chapter 4 addresses the main problem considered in this dissertation, namely safety detection in the presence of arbitrary faults and attacks in sensors. We develop sensor fusion techniques to take advantage of the inherent sensor redundancy in modern CPS. We then provide multiple ways of improving the output of sensor fusion by using historical measurements and by analyzing different schedules of sensor measurement transmissions in order to minimize the attacker's impact on the system. In summary, the contributions of Chapter 4 are:

- Development of multidimensional sensor fusion algorithm where each sensor measurement is converted to a polyhedron. The output of sensor fusion, the fusion polyhedron, is guaranteed to contain the true state assuming less than half of all sensors are attacked.

- A modified sensor fusion algorithm incorporating measurement history. System dynamics are used in order to map previous measurements to the current time. We showed that the fusion polyhedron using history is always a subset of the one computed without history.

- Comparison of different sensor transmission schedules in terms of the expected size of the fusion interval. We provided both theoretical and simulation results in favor of the Ascending schedule, the one in which most precise sensors transmit first.

- Sensor fusion in the presence of transient faults. Since standard sensor fusion loses its guarantees in the presence of sensor faults in addition to attacks, we developed a modified algorithm whose output is still guaranteed to contain the true state. In order to capture transient faults, we provide a transient fault model (TFM) for each sensor limiting the number of faulty measurements in a given window; we also showed how to obtain a TFM from available sensor data.

- Evaluation of all proposed techniques both in simulation and in experiments using the LandShark robot.

In order to further strengthen the guarantees of sensor fusion, in Chapter 5 we incorporated an additional piece of information, namely context measurements. We developed a modified algorithm with both continuous and binary measurements whose output is once again guaranteed to contain the true state. Context-aware sensor fusion would be especially useful in scenarios where more than half of the standard continuous measurements might be under attack, in which case the addition of context measurements can allow the system to still provide worst-case guarantees about its safety. To evaluate this approach, we provided a case-study simulation of a perfectly attackable system that cannot detect the attack on GPS, thus leading to a crash; if context measurements (in the form of building detections) are used, however, the system can detect that it is unsafe and avoid the collision.

Finally, in Chapter 6, we note that all of the above techniques' performance could be improved if better data is provided by the system's sensors. Thus, we developed a sensor attack detection technique in order to identify and discard attacked sensors, thereby improving the performance of estimation and sensor fusion. The attack detection algorithm was developed so that it does not raise unnecessary alarms in the presence of transient faults, which are a normal part of system operation. This approach was evaluated on real data collected from the LandShark and augmented retrospectively with several kinds of attacks.

There are multiple potential avenues for future work based on the results in this dissertation. In this chapter, we focus on two main classes of possible extensions, namely generalizing the notion of context as well as investigating further the current applications of context measurements in estimation and sensor fusion. Note that both branches include context – this should be no surprise given the rising availability of context measurements from improved machine learning and detection algorithms. In particular, some questions one might ask in terms of formalizing context are:

- What is context in general? Can we formalize the difference between standard sensors and context sensors (not just mathematically, but also conceptually)?
- When is context useful? If we assume context is always a high-level representation of other sensor data, then what is the benefit of using context as opposed to the actual sensor data?

The more specific and immediate extensions of this dissertation concern directly the approaches presented here:

- Extend Theorem 1 to the multidimensional case in order to provide more intuition about when the sigmoid-based context-aware filter's uncertainty is bounded.

- Provide conditions under which the sigmoid-based context-aware filter converges to the true state.

- Develop context-aware estimation with discrete measurements that are not just binary but come from a larger (possibly infinite) set. The main challenge with such a problem is its combinatorial nature.

- Use historical context measurements in the context-aware sensor fusion algorithm in the same way as standard continuous measurements, e.g., by using pairwise intersection.

- Note that the proposed context-aware sensor fusion algorithm only makes use of context measurements when they are equal to 1, i.e., measurements of -1 are not used explicitly in the algorithm. Adding the negative measurements is challenging because it is not clear how to maintain a bounded fusion polyhedron. Thus, this is another possible extension of the context-aware sensor fusion algorithm.

- Incorporate a transient fault model for context measurements, similar to sensor fusion with transient sensor faults as presented in Section 4.7. This problem presents a different type of challenge, namely the fact that it introduces potential mode switches for the system. Since context measurements can only be observed from certain states, for each received measurement one needs to consider whether the system is in a state from which obtaining that context measurement is possible or whether the measurement is just a false alarm. Similarly, if a measurement is not received, it is possible that the system is not in a state from which this context measurement can be obtained but it is also possible that the context measurement was not received due to an imperfect detection or classification algorithm.

Appendix A

Proofs

A.1 Proof of Proposition 2

First note that the update equation takes the form:

$$\begin{aligned} p_{k|k}(x) &= \frac{p(y_k^b \mid x)\phi(x;\mu_{k|k-1},\Sigma_{k|k-1})}{\int p(y_k^b \mid x')\phi(x';\mu_{k|k-1},\Sigma_{k|k-1})dx'} \\ &= \frac{\Phi(y_k^b(b_k^T x + a_k))\phi(x;\mu_{k|k-1},\Sigma_{k|k-1})}{Z_k}, \end{aligned}$$

where

$$Z_k = \int \Phi(y_k^b(b_k^T x' + a_k))\phi(x';\mu_{k|k-1},\Sigma_{k|k-1})dx'.$$

The derivation for Z_k is carried out as follows:

$$\begin{aligned} Z_k &= \int \Phi(y_k^b(b_k^T x' + a_k))\phi(x';\mu_{k|k-1},\Sigma_{k|k-1})dx' = \mathbb{E}_x\left[\Phi(y_k^b(b_k^T x + a_k))\right] \\ &= \mathbb{E}_x\left[\mathbb{P}(z_1 \le y_k^b(b_k^T x + a_k))\right] = \mathbb{E}_{(x,z_1)}\left[\mathbb{1}_{z_1 \le y_k^b(b_k^T x + a_k)}\right] \\ &= \mathbb{P}(y_k^b(b_k^T x + a_k) - z_1 \ge 0) \\ &= \mathbb{P}\left(y_k^b(b_k^T \mu_{k|k-1} + a_k) + z_2\sqrt{b_k^T \Sigma_{k|k-1} b_k + 1} \ge 0\right) \end{aligned}$$

$$= \mathbb{P}(z_2 \geq -M_k) = 1 - \Phi(-M_k) = \Phi(M_k),$$

where z_1 and z_2 are standard Normal random variables independent of each other and of x.

A.2 Proof of Proposition 3

To show that the function

$$g(x) = \ln(p_{k|k}(x)) \tag{A.1}$$

is concave, we need to show that its Hessian (with respect to x) is negative definite. To see this, first note that

$$\begin{aligned} g(x) = &-\ln(Z_k) + \ln(\Phi(y_k^b(b_k^T x + a_k))) - \ln(\sqrt{(2\pi)^n |\Sigma_{k|k-1}|}) \\ &- \frac{1}{2}(x - \mu_{k|k-1})^T \Sigma_{k|k-1}^{-1}(x - \mu_{k|k-1}). \end{aligned}$$

The first derivative of $g(x)$ is:

$$g'(x) = b_k y_k^b \alpha(y_k^b(b_k^T x + a_k)) - \Sigma_{k|k-1}^{-1}(x - \mu_{k|k-1}),$$

where $\alpha(x) = \phi(x; 0, 1)/\Phi(x)$. The Hessian of $g(x)$ is:

$$\begin{aligned} g''(x) &= b_k b_k^T (y_k^b)^2 [-\alpha(y_k^b(b_k^T x + a_k))(y_k^b(b_k^T x + a_k)) - \alpha^2(y_k^b(b_k^T x + a_k))] - \Sigma_{k|k-1}^{-1} \\ &= -b_k b_k^T h(y_k^b(b_k^T x + a_k)) - \Sigma_{k|k-1}^{-1}. \end{aligned}$$

Since $b_k b_k^T$ is positive semidefinite and $\Sigma_{k|k-1}$ is positive definite, it remains to show that the term $h(y_k^b(b_k^T x + a_k))$ is non-negative; but this is true as shown in Proposition 5.

A.3 Proof of Proposition 4

First note that

$$\mu_{k|k} = \int x' \frac{\Phi(y_k^b(b_k^T x' + a_k))\phi(x'; \mu_{k|k-1}, \Sigma_{k|k-1})}{Z_k} dx'.$$

One way to compute the mean in closed form is, similar to the derivation in Chapter 3.9 in [165], by first computing the gradient with respect to $\mu_{k|k-1}$ of the following two equivalent expressions for Z_k:

$$\int \Phi(y_k^b(b_k^T x' + a_k))\phi(x'; \mu_{k|k-1}, \Sigma_{k|k-1}) dx' = \Phi(M_k). \tag{A.2}$$

The corresponding derivatives are:

$$\begin{aligned}\frac{\partial Z_k}{\partial \mu_{k|k-1}} &= \int \Sigma_{k|k-1}^{-1}(x' - \mu_{k|k-1})\Phi(y_k^b(b_k^T x' + a_k)) \cdot \phi(x'; \mu_{k|k-1}, \Sigma_{k|k-1}) dx' \\ &= y_k^b b_k \frac{\phi(M_k; 0, 1)}{\sqrt{b_k^T \Sigma_{k|k-1} b_k + 1}},\end{aligned}$$

where we used the fact that $\partial\Phi(x)/\partial x = \phi(x)$. Note that the first term in the integral on the left-hand side is $Z_k \Sigma_{k|k-1}^{-1} \mu_{k|k}$. The second term is $Z_k \Sigma_{k|k-1}^{-1} \mu_{k|k-1}$. Therefore, we get

$$Z_k \Sigma_{k|k-1}^{-1} \mu_{k|k} = Z_k \Sigma_{k|k-1}^{-1} \mu_{k|k-1} + b_k \frac{y_k^b \phi(M_k; 0, 1)}{\sqrt{b_k^T \Sigma_{k|k-1} b_k + 1}}.$$

Thus, we arrive at

$$\mu_{k|k} = \mu_{k|k-1} + y_k^b \Sigma_{k|k-1} b_k \frac{\alpha(M_k)}{\sqrt{b_k^T \Sigma_{k|k-1} b_k + 1}},$$

where we used the second expression for Z_k in order to get α. The final expression for $\mu_{k|k}$ is obtained by solving for χ_k in the equation $\alpha(M_k)(b_k^T \Sigma_{k|k-1} b_k + 1)^{-1/2} =$

$(b_k^T \Sigma_{k|k-1} b_k + \chi_k)^{-1}$.

The expression for the covariance matrix is:

$$\Sigma_{k|k} = \hat{\Sigma}_{k|k} - \mu_{k|k}\mu_{k|k}^T, \tag{A.3}$$

where

$$\hat{\Sigma}_{k|k} = \int x' x'^T \frac{\Phi(y_k^b(b_k^T x' + a_k))\phi(x'; \mu_{k|k-1}, \Sigma_{k|k-1})}{Z_k} dx'.$$

$\hat{\Sigma}_{k|k}$ can be computed in closed-form similar to the mean, by computing the Hessians with respect to $\mu_{k|k-1}$ of both sides of (A.2):

$$\begin{aligned}
&\int \Sigma_{k|k-1}^{-1}(x' - \mu_{k|k-1})(x' - \mu_{k|k-1})^T \Sigma_{k|k-1}^{-1} \Phi(y_k^b(b_k^T x' + a_k))\phi(x'; \mu_{k|k-1}, \Sigma_{k|k-1}) dx' \\
&\quad - \int \Sigma_{k|k-1}^{-1} \Phi(y_k^b(b_k^T x' + a_k))\phi(x'; \mu_{k|k-1}, \Sigma_{k|k-1}) dx' \\
&\quad = -y_k^b b_k b_k^T \frac{\phi(M_k; 0, 1)(b_k^T \mu_{k|k-1} + a_k)}{(b_k^T \Sigma_{k|k-1} b_k + 1)^{3/2}}.
\end{aligned}$$

Note that one of the terms in the integral on the left-hand side is $Z_k \Sigma_{k|k-1}^{-1} \hat{\Sigma}_{k|k} \Sigma_{k|k-1}^{-1}$. Therefore, we rearrange terms and divide by Z_k to obtain the following:

$$\begin{aligned}
\Sigma_{k|k-1}^{-1} \hat{\Sigma}_{k|k} \Sigma_{k|k-1}^{-1} = {} & \Sigma_{k|k-1}^{-1} + \Sigma_{k|k-1}^{-1} \mu_{k|k} \mu_{k|k-1}^T \Sigma_{k|k-1}^{-1} + \Sigma_{k|k-1}^{-1} \mu_{k|k-1} \mu_{k|k}^T \Sigma_{k|k-1}^{-1} \\
& - \Sigma_{k|k-1}^{-1} \mu_{k|k-1} \mu_{k|k-1}^T \Sigma_{k|k-1}^{-1} - y_k^b b_k b_k^T \frac{\alpha(M_k)(b_k^T \mu_{k|k-1} + a_k)}{(b_k^T \Sigma_{k|k-1} b_k + 1)^{3/2}}.
\end{aligned}$$

Finally, we arrive at the expression for $\hat{\Sigma}_{k|k}$:

$$\begin{aligned}
\hat{\Sigma}_{k|k} = {} & \Sigma_{k|k-1} + \mu_{k|k} \mu_{k|k-1}^T + \mu_{k|k-1} \mu_{k|k}^T - \mu_{k|k-1} \mu_{k|k-1}^T \\
& - y_k^b \Sigma_{k|k-1} b_k b_k^T \Sigma_{k|k-1} \frac{\alpha(M_k)(b_k^T \mu_{k|k-1} + a_k)}{(b_k^T \Sigma_{k|k-1} b_k + 1)^{3/2}}.
\end{aligned}$$

Thus, the covariance matrix can be computed by plugging in the expression for $\hat{\Sigma}_{k|k}$

in (A.3). To simplify it to the final form shown in the Proposition statement, we first plug in the expression for $\mu_{k|k} - \mu_{k|k-1}$ from (3.10) and then solve for γ_k.

A.4 Proof of Theorem 1

Consider the (scalar) modified algebraic Riccati equation (MARE) defined as:

$$g_\beta(x) = axa + q - \beta axb(bxb + 1)^{-1}bxa,$$

where $b = \min_i |b^i|$, i.e., the minimum-in-magnitude of all context weights. Note that if $\beta = 1$, then this becomes the standard algebraic Riccati equation, which converges for any σ_0. On the other hand if $\beta = 0$, the covariance matrix diverges for some σ_0 if a is unstable. We use the MARE to bound the expected value of context-aware filter's variance and give conditions on β for which the expectation is bounded.

We first bound the expected variance of the filter using the MARE. From (3.14), followed by applying the prediction step, we get (by using the simplified notation $\sigma_k = \sigma_{k|k-1}$):

$$\begin{aligned}
\mathbb{E}[\sigma_{k+1}] &= \mathbb{E}[a\sigma_k a + q - \tau_m a\sigma_k b_k(b_k\sigma_k b_k + \gamma_k^m)^{-1}b_k\sigma_k a - \tau_p a\sigma_k b_k(b_k\sigma_k b_k + \gamma_k^p)^{-1}b_k\sigma_k a]\\
&\leq \mathbb{E}[a\sigma_k a + q - \eta a\sigma_k b(b\sigma_k b + \gamma_k^m)^{-1}b\sigma_k a - \eta a\sigma_k b(b\sigma_k b + \gamma_k^p)^{-1}b\sigma_k a]\\
&\leq \mathbb{E}[a\sigma_k a + q - \eta a\sigma_k b(b\sigma_k b + \min\{\gamma_k^m, \gamma_k^p\})^{-1}b\sigma_k a]\\
&\leq \mathbb{E}[a\sigma_k a + q - \eta a\sigma_k b\left(b\sigma_k b + \frac{(1-h(0))(b\sigma_k b) + 1}{h(0)}\right)^{-1} b\sigma_k a]\\
&= \mathbb{E}[a\sigma_k a + q - \rho a\sigma_k b(b\sigma_k b + 1)^{-1}b\sigma_k a]\\
&= \mathbb{E}[g_\rho(\sigma_k)],
\end{aligned}$$

where $\rho = \eta h(0) < 1$, τ_m is the probability of $y_k^b = -1$ (with resulting γ_k^m); τ_p and γ_k^p are their analogues when $y_k^b = 1$. The first equality is the expected value of σ_{k+1} for each possible value of y_k^b. The second inequality uses the fact that both $\tau_p, \tau_m \geq \eta$.

In the third inequality we simply discard one of the two negative terms, keeping the one with smaller γ_k (i.e., the one that results in $M_k < 0$; note that $0 < h(x) < 1$ and $h'(x) < 0$, from Proposition 5). The last inequality is true because $h(x) > h(0)$ for any $x < 0$.

The rest of the proof mimics the proof of Theorem 3 in [190]. Consider the sequence $s_{k+1} = g_\rho(s_k)$, with $s_0 = \sigma_0$. We show that $\mathbb{E}[\sigma_k] \leq s_k$ using induction. Note that $\mathbb{E}[\sigma_k] \leq s_k$ implies:

$$\mathbb{E}[\sigma_{k+1}] \leq \mathbb{E}[g_\rho(\sigma_k)] \leq g_\rho(\mathbb{E}[\sigma_k]) \leq g_\rho(s_k) = s_{k+1},$$

where the first inequality was shown above, and the second and third inequalities are shown in Lemma 1 in [190]. Furthermore, as shown in Theorem 3 in [190], s_k is bounded from above, given that $\rho > \overline{\rho}$ ($\overline{\rho} \in [0, 1)$, as shown in [190]), i.e.,

$$\mathbb{E}[\sigma_k] \leq s_k \leq M_{\sigma_0}, \forall k.$$

A.5 Proof of Lemma 1

The proof proceeds by induction on k. The base case is shown in (3.14). For the induction step, we assume that $K < N$ updates result in the form in (3.18), with matrices Γ_k and B_k replaced by Γ_K and B_K, respectively. Given weights b_{k+K+1}, the next discrete update is

$$\Sigma_{k+K+1} = \Sigma_{k+K} - \Sigma_{k+K} b_{k+K+1} \beta^{-1} b_{k+K+1}^T \Sigma_{k+K} \tag{A.4}$$

where by induction

$$\Sigma_{k+K} = \Sigma_k - \Sigma_k B_K^T (B_K \Sigma_k B_K^T + \Gamma_K)^{-1} B_K \Sigma_k,$$
$$\beta = b_{k+K+1}^T \Sigma_{k+K} b_{k+K+1} + \gamma_{k+K+1}.$$

By rearranging terms and using the block matrix inversion lemma, Equation (A.4) can now be written as

$$\begin{aligned}\Sigma_{k+K+1} = \Sigma_k - \left[\begin{array}{cc} \Sigma_k B_K^T & \Sigma_k b_{k+K+1} \end{array} \right] \cdot \\ \cdot \left[\begin{array}{cc} B_K \Sigma_k B_K^T + \Gamma_K & B_K \Sigma_k b_{k+K+1} \\ b_{k+K+1}^T \Sigma_k B_K^T & b_{k+K+1}^T \Sigma_k b_{k+K+1} + \gamma_{k+K+1} \end{array} \right]^{-1} \\ \cdot \left[\begin{array}{c} B_K \Sigma_k \\ b_{k+K+1}^T \Sigma_k \end{array} \right],\end{aligned}$$

i.e.,

$$\begin{aligned}\Sigma_{k+K+1} = \Sigma_k - \Sigma_k \left[\begin{array}{cc} B_K^T & b_{k+K+1} \end{array} \right] \cdot \\ \cdot \left[\left[\begin{array}{c} B_K \\ b_{k+K+1} \end{array} \right] \Sigma_k \left[\begin{array}{cc} B_K^T & b_{k+K+1}^T \end{array} \right] + \left[\begin{array}{cc} \Gamma_K & 0 \\ 0 & \gamma_{k+K+1} \end{array} \right] \right]^{-1} \\ \cdot \left[\begin{array}{c} B_K \\ b_{k+K+1}^T \end{array} \right] \Sigma_k,\end{aligned}$$

which has the desired form of the Riccati (update) equation.

A.6 Proof of Theorem 2

We first prove sufficiency (<=). Let B be the matrix of persistently exciting b^i, i.e., $B = [b^1, \ldots, b^d]^T$. Note that B is square and invertible. Consider the sequence of times $k_1, k_2, \ldots$, where $k_1 = 1$ and $k_{t+1} = k_t + l_{k_t} + 1$; note that all b^i in B occur in between each pair of k_t and k_{t+1} by construction. Thus, using Lemma 1, it suffices to show that the eigenvalues of the covariance sequence

$$\Sigma_{k_{t+1}} = \Sigma_{k_t} - \Sigma_{k_t} B^T (B \Sigma_{k_t} B^T + \Gamma_{k_t})^{-1} B \Sigma_{k_t} \tag{A.5}$$

converge to 0 in probability. Note from (3.14) that no binary update can increase the eigenvalues of Σ_k, so any updates with weights and offsets not in $\mathcal{P}$ can be ignored as they do not affect the convergence.

Diagonalizing $\Sigma_{k_t} = UDU^T$, we rewrite (A.5):

$$\Sigma_{k_{t+1}} = U(D - D(D + M\Gamma_{k_t}M^T)^{-1}D)U^T, \tag{A.6}$$

where $M = U^T V^{-1}$. Diagonalizing $M\Gamma_{k_t}M^T = P\Lambda P^T$, we conclude that

$$\Sigma_{k_{t+1}} \preceq U(D - D(D + \delta_{max}I)^{-1}D)U^T, \tag{A.7}$$

where δ_{max} is the largest eigenvalue of $M\Gamma_{k_t}M^T$. Thus, each eigenvalue $\lambda^i_{k_t}$ is reduced at least by $(\lambda^i_{k_t})^2/(\lambda^i_{k_t} + \delta_{max})$. Therefore, $\lambda^i_{k_t} \rightarrow 0$ as long as δ_{max} is bounded from above. But δ_{max} is bounded if $\gamma^{max}_{k_t}$ (the largest γ_k between times k_t and k_{t+1}) is bounded. From (3.15), it can be seen that γ_k is bounded from above if the function h is bounded from below. But for each k, $M_k < 0$ with probability at least

$$\min_{y^b_k \in \{1,-1\}, (b^i, a^i) \in \mathcal{P}} \Phi(y^b_k((b^i)^T x^* + a_i)),$$

where x^* is the true (non-moving) state. Thus, h has a non-zero probability of having negative input, i.e., it is bounded from below by $h(0) = \alpha^2(0)$ (note that $h'(x) < 0$, from Proposition 5). Thus, the probability that h is never bounded from below converges to 0, i.e., $\lambda^i_{k_t} \xrightarrow{p} 0$.

To prove necessity (=>), note that if (b_k, a_k) is not persistently exciting, there exists a time K, such that the set of context weights b_k for $k > K$ does not span $\mathbb{R}^d$, i.e., the matrix B_K of all such weights is not full rank. We now show this implies that there exists at least one λ^i_k that does not go to 0. Returning to (3.18), note that there exists a rotation matrix U such that one eigenvector (call it p) of $\Sigma_k U^T$

is aligned with an eigenvector of $B_K^\perp$, the null space of B_K. Consider the matrix

$$G = U(\Sigma_k - \Sigma_k B_K^T (B_K \Sigma_k B_K^T + \Gamma_k)^{-1} B_K \Sigma_k) U^T.$$

G has the same eigenvalues as Σ_{k+K} but the eigenvalue corresponding to p is also an eigenvalue of Σ_k, i.e., this eigenvalue remains unchanged when B_K is not full rank.

A.7 Proof of Theorem 3

First note that applying the matrix inversion lemma to the covariance update in (3.14), we get:

$$\begin{aligned}\Omega_{k+1} &= (\Sigma_k - \Sigma_k b_{k+1}(b_{k+1}^T \Sigma_k b_{k+1} + \gamma_{k+1})^{-1} b_{k+1}^T \Sigma_k)^{-1} \\ &= \Sigma_k^{-1} + b_{k+1}\gamma_{k+1}^{-1} b_{k+1}^T.\end{aligned}$$

Therefore,

$$\Omega_{k+1}^s = \Omega_{k+1} - \Omega_k = b_{k+1}\gamma_{k+1}^{-1} b_{k+1}^T.$$

The mean at time $k+1$ is equal to (by using the mean update in (3.10)):

$$\begin{aligned}\mu_{k+1} &= \mu_k + \Sigma_k b_{k+1}(b_{k+1}^T \Sigma_k b_{k+1} + \chi_{k+1})^{-1} y_{k+1}^b \\ &= \mu_k + \Sigma_k b_{k+1} N_{k+1}^{-1} y_{k+1}^b,\end{aligned}$$

where $N_{k+1} = b_{k+1}^T \Sigma_k b_{k+1} + \chi_{k+1}$. Thus, the information mean of the "site" approximation becomes

$$\begin{aligned}\omega_{k+1}^s &= \Omega_{k+1}\mu_{k+1} - \Omega_k \mu_k \\ &= \Omega_{k+1}\mu_k + (I + L_{k+1}) b_{k+1} N_{k+1}^{-1} y_{k+1}^b - \Omega_k \mu_k\end{aligned}$$

$$= \Omega^s_{k+1}\mu_k + \Omega_k\mu_k + (I + L_{k+1})b_{k+1}N^{-1}_{k+1}y^b_{k+1} - \Omega_k\mu_k,$$

where $L_{k+1} = b_{k+1}\gamma^{-1}_{k+1}b^T_{k+1}\Sigma_k$, and we used the inverse-lemma expression for Ω_{k+1}.

A.8 Proof of Corollary 3

As shown in Theorem 2, if b_k is persistently exciting, then all eigenvalues of Σ_k converge to 0 for large k. To analyze the convergence of the natural parameters of the "site" approximations, first note that the first derivative of ψ is as follows:

$$\psi^{'}_{k+1}(x) = -b_{k+1}\alpha(y^b_{k+1}(b^T_{k+1}x + a_{k+1}))y^b_{k+1}. \tag{A.8}$$

The second derivative of ψ is:

$$\psi^{''}_{k+1}(x) = b_{k+1}b^T_{k+1}h(y^b_{k+1}(b^T_{k+1}x + a_{k+1})). \tag{A.9}$$

We first show that $\Omega^s_{k+1} = b_{k+1}\gamma^{-1}_{k+1}b^T_{k+1}$ converges to $\psi^{''}_{k+1}(\mu_k)$, i.e., that γ^{-1}_{k+1} converges to $h(y^b_{k+1}(b^T_{k+1}\mu_k + a_{k+1}))$. But this is clear from (3.15): as the eigenvalues of Σ_k converge to 0, γ^{-1}_{k+1} converges to $h(M_{k+1})$, and M_{k+1} converges to $y^b_{k+1}(b^T_{k+1}\mu_k + a_{k+1})$.

As derived in (3.22), the information mean is $\omega^s_{k+1} = \Omega^s_{k+1}\mu_k + (I+L_{k+1})b_{k+1}N^{-1}_{k+1}y^b_{k+1}$. First note that N^{-1}_{k+1} converges to $1/\chi_{k+1}$, which in turn converges to $\alpha(y^b_{k+1}(b^T_{k+1}\mu_k + a_{k+1}))$, as can be seen from (3.11). Thus, in order to show that the second term of ω^s_{k+1} converges to $-\psi^{'}_{k+1}(\mu_k)$, it suffices to show that L_{k+1} converges to 0. But this is clear from the definition of L_{k+1} in Theorem 3.

A.9 Proof of Proposition 6

Since there are at least $n - f$ correct polyhedra, the true state is contained in at least $n - f$ polyhedra, and hence it will be included in the fusion polyhedron.

A.10 Proof of Proposition 7

We first note that any set that is guaranteed to contain the true state must contain $R_{\mathcal{N}_k,f}$ since any point that is excluded may be the true state. This proves the proposition since $\texttt{conv}(R_{\mathcal{N}_k,f})$ is the smallest convex set that contains $R_{\mathcal{N}_k,f}$.

A.11 Proof of Lemma 2

Let p be any vertex of the convex hull. Then $p = \sum \theta_i v_i$, where the v_i are the vertices of the polyhedra, $\sum \theta_i = 1$ and $\theta_i \geq 0$ (i.e., p is a convex combination of the v_i's). This means that p lies on a hyperplane defined by some of the v_i's, hence it cannot be a vertex, unless it is one of the v_i's.

A.12 Proof of Theorem 4

We use a counting argument. Let $\mathcal{V}$ be the set of vertices of $S_{\mathcal{N}_k,f}$. By Lemma 2, each vertex in $\mathcal{V}$ is a vertex of one of the polyhedra formed by the intersection of $n - f$ of the sensor polyhedra (in step 4 of Algorithm 1). Therefore, it is contained in at least $n - f$ polyhedra. For each $p \in \mathcal{V}$, let P_p denote the number of polyhedra containing p. Consequently, $P_p \geq n - f$. Then

$$v_P(n - f) \leq \sum_{p \in \mathcal{V}} P_p.$$

The sum in the right-hand side can be split into two sums. One contains the number of polyhedra where each of the polyhedra contains all v_P vertices (we denote this number by a). Then the number of the remaining polyhedra is $n-a$. The part of the sum due to the polyhedra that contain fewer than v_P vertices can be bounded from above by $(n-a)(v_P-1)$ since each of these polyhedra contains at most v_P-1 vertices. We then have

$$v_P(n-f) \leq av_P + (n-a)(v_P-1),$$

which implies that $a \geq n-fv_P$, i.e., at least $n-fv_P$ polyhedra contain the v_P vertices of the fusion polyhedron. Since polyhedra, including the fusion polyhedron, are convex, we conclude that at least $n-fv_P$ polyhedra contain the fusion polyhedron. This completes the proof, since

$$|S_{\mathcal{N}_k,f}| \leq \max_{n-fv_P}\{|P| : P \in \mathcal{N}_k\} = \min_{fv_P+1}\{|P| : P \in \mathcal{N}_k\}.$$

A.13 Proof of Theorem 5

Assume the opposite – that there exists a point $x_A \in S_{\mathcal{N}_k,f}$ that is not in $\texttt{conv}(\mathcal{C}_k)$. Then for any convex combination $\sum \theta_i v_i = x_A$, where $v_i \in P_j$ for some j, at least one v_i must not be in any polyhedron in $\mathcal{C}_k$, meaning that it is contained in at most f polyhedra, where $f < n-f$. Therefore, there does not exist a convex combination $\sum \theta_i v_i = x_A$ with all v_i contained in at least $n-f$ polyhedra, and hence x_A cannot be in $S_{\mathcal{N}_k,f}$.

A.14 Proof of Theorem 6

Consider any point $p \in m(R_{\mathcal{N}_k,f}) \cap R_{\mathcal{N}_{k+1},f}$. Then p lies in at least $n-f$ polyhedra in $\mathcal{N}_{k+1}$, and there exists a q such that $p \in m(q)$ that lies in at least $n-f$

polyhedra in $\mathcal{N}_k$. Thus, p lies in at least $2n - 2f$ polyhedra in $m(\mathcal{N}_k) \cup \mathcal{N}_{k+1}$, i.e., $p \in R_{m(\mathcal{N}_k)\cup\mathcal{N}_{k+1},2f}$, implying

$$\texttt{conv}(m(R_{\mathcal{N}_k,f}) \cap R_{\mathcal{N}_{k+1},f}) \subseteq \texttt{conv}(R_{m(\mathcal{N}_k)\cup\mathcal{N}_{k+1},2f}) = S_{m(\mathcal{N}_k)\cup\mathcal{N}_{k+1},2f}.$$

A.15 Proof of Theorem 7

Note that for any sets $\mathcal{A}$ and $\mathcal{B}$, $\texttt{conv}(\mathcal{A} \cap \mathcal{B}) \subseteq \texttt{conv}(\mathcal{A})$, and thus

$$\texttt{conv}(m(R_{\mathcal{N}_k,f}) \cap R_{\mathcal{N}_{k+1},f}) \subseteq \texttt{conv}(R_{\mathcal{N}_{k+1},f}) = S_{\mathcal{N}_{k+1},f}.$$

Furthermore, any point $p \in \texttt{conv}(m(R_{\mathcal{N}_k,f}) \cap R_{\mathcal{N}_{k+1},f})$ is a convex combination of points q_i in $m(R_{\mathcal{N}_k,f})$. But $m(R_{\mathcal{N}_k,f}) \subseteq m(S_{\mathcal{N}_k,f})$ (since $R_{\mathcal{N}_k,f} \subseteq S_{\mathcal{N}_k,f}$) and the fact that $m(S_{\mathcal{N}_k,f})$ is convex imply $p \in m(S_{\mathcal{N}_k,f})$. Accordingly,

$$\begin{aligned}&\texttt{conv}(m(R_{\mathcal{N}_k,f}) \cap R_{\mathcal{N}_k,f}) \subseteq S_{\mathcal{N}_k,f} \text{ and}\\ &\texttt{conv}(m(R_{\mathcal{N}_k,f}) \cap R_{\mathcal{N}_{k+1},f}) \subseteq m(S_{\mathcal{N}_k,f})\end{aligned}$$

implying

$$\texttt{conv}(m(R_{\mathcal{N}_k,f}) \cap R_{\mathcal{N}_{k+1},f}) \subseteq m(S_{\mathcal{N}_k,f}) \cap S_{\mathcal{N}_{k+1},f}.$$

A.16 Proof of Theorem 8

Note that, since the fusion interval is always guaranteed to contain the true value, the number of corrupted polyhedra in *map_S_and_fuse* is still at most f, but the number of correct ones is now at least $n + 1 - f$. In addition, note that

$$m(R_{\mathcal{N}_k,f}) \cap R_{\mathcal{N}_{k+1},f} \subseteq m(S_{\mathcal{N}_k,f})$$

since $m(R_{\mathcal{N}_k,f}) \subseteq m(S_{\mathcal{N}_k,f})$. Furthermore, any point $p \in R_{\mathcal{N}_{k+1},f}$ is contained in $n-f$ polyhedra in $\mathcal{N}_{k+1}$. Thus, all points in $m(R_{\mathcal{N}_k,f}) \cap R_{\mathcal{N}_{k+1},f}$ are contained in $n+1-f$ polyhedra in $m(S_{\mathcal{N}_k,f}) \cup \mathcal{N}_{k+1}$, and hence in $R_{m(S_{\mathcal{N}_k,f}) \cup \mathcal{N}_{k+1},f}$. Since the fusion polyhedron is convex,

$$\texttt{conv}(m(R_{\mathcal{N}_k,f}) \cap R_{\mathcal{N}_{k+1},f}) \subseteq S_{m(S_{\mathcal{N}_k,f}) \cup \mathcal{N}_{k+1},f}.$$

A.17 Proof of Theorem 9

Let p be any point in $R_{m(\mathcal{N}_k) \cap_p \mathcal{N}_{k+1},f}$. Then p lies in at least $n-f$ polyhedra in $m(\mathcal{N}_k)$ and at least $n-f$ polyhedra in $\mathcal{N}_{k+1}$. Hence,

$$R_{m(\mathcal{N}_k) \cap_p \mathcal{N}_{k+1},f} \subseteq R_{\mathcal{N}_{k+1},f}.$$

Furthermore, there exists a point $q = A^{-1}p$ that is contained in $n-f$ intervals in $\mathcal{N}_k$. Therefore, p is also contained in $m(R_{\mathcal{N}_k,f})$. Then

$$R_{m(\mathcal{N}_k) \cap_p \mathcal{N}_{k+1},f} \subseteq m(R_{\mathcal{N}_k,f}) \cap R_{\mathcal{N}_{k+1},f}, \text{ i.e.,}$$
$$S_{m(\mathcal{N}_k) \cap_p \mathcal{N}_{k+1},f} \subseteq \texttt{conv}(m(R_{\mathcal{N}_k,f}) \cap R_{\mathcal{N}_{k+1},f}).$$

A.18 Proof of Proposition 8

Note that pairwise intersection does not increase the number of attacked polyhedra. If a sensor is not attacked, then both of its polyhedra (in time k and $k+1$) contain the true value; in addition, the map m preserves the correctness of polyhedra, hence any pairwise intersection will also contain the true value. Thus, the number of attacked and non-attacked sensors is the same, therefore Proposition 6 implies that the fusion polyhedron contains the true value.

A.19 Proof of Proposition 9

Each of the polyhedra computed after pairwise intersection (i.e., $m(P_{i,k}) \cap P_{i,k+1}$) is a subset of the corresponding polyhedron when no history is used (i.e., $P_{i,k+1}$). Consequently, the fusion polyhedron will always be a subset of the fusion polyhedron obtained when no history is used.

A.20 Proof of Theorem 10

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 $I_l \in \mathcal{C}^R$, I_l must overlap with at least one point in $S_{\mathcal{C}^S \cup \Delta, 0}$ (the overlap must contain the true state) and since $|I_l| \leq (|a_{min}| - |S_{\mathcal{C}^S \cup \Delta, 0}|)/2$ then I_l 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 4.9a 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 4.9b presents an example of this case. The unseen interval, I_3, cannot change the largest and smallest points contained in at least one correct interval.

A.21 Proof of Theorem 11

Let I_l and I_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 I_l and its upper bound cannot be larger than the upper bound of I_u. Thus, the width of $S_{\mathcal{N},f}$ is bounded by the sum of the widths of I_l and I_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.

A.22 Proof of Theorem 12

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 I_c. Since I_c is correct it must contain $S_{\mathcal{C},0}$, which implies $d + |S_{\mathcal{C},0}| \leq |I_c| \leq |I_{max}|$, where I_{max} is the largest correct interval. Let I_a be any attacked interval that contains A. Because $|I_a| \geq |I_{max}|$, I_a 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.

A.23 Proof of Theorem 13

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

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

A.24 Proof of Theorem 14

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 k 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 $k' > k$. Without loss of generality, we can assume s_i does not have a faulty measurement at k'.[1] 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 k, we do not affect the magnitude of E (since the number of faulty measurements in each round remains the same). By replacing all such pairs we eventually transform s into a new schedule s' that is exactly the schedule suggested by the proposed algorithm here. Therefore, Algorithm 3 is optimal.

[1] 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 k and replace it with a faulty measurement by s_j (still within its TFM).

A.25 Proof of Proposition 10

First note that the mapping function m is called $O(W^2)$ times (line 3 inside the loop). Additionally, computing the fusion polyhedron in FP_C (line 6) requires $O(W \log W)$ time (as shown in [136], the sensor fusion algorithm takes $O(n \log n)$ time, where n is the number of sensors).

As for the second claim, note that the cost of obtaining one element of FP is again $O(n \log n)$, i.e., one run of the sensor fusion algorithm. Since the size of FP is W, the claim follows.

A.26 Proof of Lemma 3

Since the two polyhedra have an empty intersection, the true state can lie in at most one of them, i.e., at least one of them cannot contain the true state.

A.27 Proof of Lemma 4

Note that a weak inconsistency at time k' implies at least one sensor provides a faulty measurement at k', 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.

A.28 Proof of Theorem 15

If two sensors are strongly inconsistent, then one of them must be non-transiently faulty. Since non-transient faults are equivalent to attacks in this work, the Lemma follows.

A.29 Proof of Theorem 16

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.

A.30 Proof of Lemma 5

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 TFMs, and one cannot conclude that an attack exists.

A.31 Proof of Proposition 11

The claim follows from the fact that both Theorems provide sufficient conditions for the existence of an attack.

Bibliography

[1] Alaris 8015 PCU. http://www.medonecapital.com/equipment/pumps/infusion/alaris-8015-pcu.

[2] The Predator B Remotely Piloted Aircraft. http://www.ga-asi.com/predator-b.

[3] Roche Cobas B 123 POC System. http://www.roche.com/products/product-details.htm?region=us&type=product&id=146.

[4] Shenzhen Bestman Instrument Co. Pulse Oximeter. http://www.szbestman.com/contents/76/669.html.

[5] The Dräger Apollo Anesthesia Machine. http://www.draeger.com/sites/enus_us/Pages /Hospital/Apollo.aspx.

[6] Why Intuitive issued a recall for da Vinci surgical system. https://www.advisory.com/daily-briefing/2013/12/06/intuitive-says-da-vinci-surgical-system-can-stall-issues-recall. Accessed: 2016-08-26.

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

[8] US National Highway Traffic Safety Administration. Investigation pe 16-007. https://static.nhtsa.gov/odi/inv/2016/INCLA-PE16007-7876.pdf.

[9] T. Alpcan and T. Basar. A game theoretic analysis of intrusion detection in access control systems. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 2, pages 1568–1573. IEEE, 2004.

[10] S. Amin, A. Cárdenas, and S. Sastry. Safe and secure networked control systems under denial-of-service attacks. In *International Workshop on Hybrid Systems: Computation and Control*, pages 31–45. Springer, 2009.

[11] S. Amin, G. A. Schwartz, and S. S. Sastry. On the interdependence of reliability and security in networked control systems. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 4078–4083. IEEE, 2011.

[12] S. Amin, G. A. Schwartz, and S. S. Sastry. Security of interdependent and identical networked control systems. *Automatica*, 49(1):186–192, 2013.

[13] R. Anati, D. Scaramuzza, K. Derpanis, and K. Daniilidis. Robot localization using soft object detection. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4992–4999, 2012.

[14] B. D. O. Anderson and J. B. Moore. *Optimal filtering.* Courier Corporation, 2012.

[15] R. Anderson. *Security engineering.* John Wiley & Sons, 2008.

[16] D. Arney, M. Pajic, J. Goldman, I. Lee, R. Mangharam, and O. Sokolsky. Toward patient safety in closed-loop medical device systems. In *Proceedings of the 1st International Conference on Cyber-Physical Systems*, pages 139–148, 2010.

[17] K. J. Åström and P. Eykhoff. System identification: A survey. *Automatica*, 7(2):123–162, 1971.

[18] N. Atanasov, R. Tron, V. Preciado, and G. Pappas. Joint Estimation and Localization in Sensor Networks. In *IEEE Conf. on Decision and Control*, pages 6875–6882, 2014.

[19] N. Atanasov, M. Zhu, K. Daniilidis, and G. Pappas. Semantic localization via the matrix permanent. In *Robotics: Science and Systems*, 2014.

[20] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete and Computational Geometry*, 8(1):295–313, 1992.

[21] C. Basile, M. Gupta, Z. Kalbarczyk, and R. K. Iyer. An approach for detecting and distinguishing errors versus attacks in sensor networks. In *International Conference on Dependable Systems and Networks (DSN'06)*, pages 473–484. IEEE, 2006.

[22] T. Bass. Intrusion detection systems and multisensor data fusion. *Communications of the ACM*, 43(4):99–105, 2000.

[23] M. Basseville. Detecting changes in signals and systemsa survey. *Automatica*, 24(3):309–326, 1988.

[24] M. Basseville and I. V. Nikiforov. *Detection of abrupt changes: theory and application*, volume 104. Prentice Hall Englewood Cliffs, 1993.

[25] R. V. Beard. *Failure accomodation in linear systems through self-reorganization.* PhD thesis, Massachusetts Institute of Technology, 1971.

[26] D. P. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific Belmont, MA, 1995.

[27] D. P. Bertsekas and I. B. Rhodes. Recursive state estimation for a set-membership description of uncertainty. *Automatic Control, IEEE Transactions on*, 16(2):117–128, 1971.

[28] P. J. Bickel and K. A. Doksum. *Mathematical statistics: basic ideas and selected topics*, volume 2. CRC Press, 2015.

[29] C. Bishop. *Pattern recognition and machine learning*, volume 1. Springer New York, 2006.

[30] 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*, pages 1–7, July 2010.

[31] M. Blaschko and C. Lampert. Object localization with global and local context kernels. In *British Machine Vision Conference (BMVC)*, pages 63.1–63.11, 2009.

[32] F. E. Block, L. Nuutinen, and B. Ballast. Optimization of alarms: a study on alarm limits, alarm sounds, and false alarms, intended to reduce annoyance. *Journal of clinical monitoring and computing*, 15(2):75–83, 1999.

[33] P. Bogdan, S. Jain, K. Goyal, and R. Marculescu. Implantable pacemakers control and optimization via fractional calculus approaches: A cyber-physical systems perspective. In *Proceedings of the Third International Conference on Cyber-Physical Systems*, pages 23–32, 2012.

[34] T. Bohlin and S. F. Graebe. Issues in nonlinear stochastic grey box identification. *International Journal of Adaptive Control and Signal Processing*, 9(6):465–490, 1995.

[35] M. Bond and R. Anderson. Api-level attacks on embedded systems. *Computer*, 34(10):67–75, 2001.

[36] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[37] M. Breton, A. Farret, D. Bruttomesso, S. Anderson, L. Magni, S. Patek, C. Dalla Man, J. Place, S. Demartini, S. Del Favero, C. Toffanin, C. Hughes-Karvetski, E. Dassau, H. Zisser, F. J. Doyle III, G. De Nicolao, A. Avogaro, C. Cobelli, E. Renard, and B. Kovatchev. Fully integrated artificial pancreas in type 1 diabetes modular closed-loop glucose control maintains near normoglycemia. *Diabetes*, 61(9):2230–2237, 2012.

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

[39] S. Buchegger and T. Alpcan. Security games for vehicular networks. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 244–251. IEEE, 2008.

[40] M. Buevich, D. Schnitzer, T. Escalada, A. Jacquiau-Chamski, and A. Rowe. Fine-grained remote monitoring, control and pre-paid electrical service in rural microgrids. In *Information Processing in Sensor Networks, IPSN-14 Proceedings of the 13th International Symposium on*, pages 1–11. IEEE, 2014.

[41] A. Burgos, A. Goñi, A. Illarramendi, and J. Bermúdez. Real-time detection of apneas on a pda. *IEEE Transactions on Information Technology in Biomedicine*, 14(4):995–1002, 2010.

[42] E. Byres and J. Lowe. The myths and facts behind cyber security risks for industrial control systems. In *Proceedings of the VDE Kongress*, volume 116, pages 213–218, 2004.

[43] A. A. Cárdenas, S. Amin, and S. Sastry. Research challenges for the security of control systems. In *HotSec*, 2008.

[44] A. A. Cardenas, S. Amin, and S. Sastry. Secure control: Towards survivable cyber-physical systems. *System*, 1(a2):a3, 2008.

[45] Z. Chair and P.K. Varshney. Optimal data fusion in multiple sensor detection systems. *Aerospace and Electronic Systems, IEEE Transactions on*, AES-22(1):98–101, Jan 1986.

[46] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

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

[48] H. Chen and D. Wagner. Mops: an infrastructure for examining security properties of software. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 235–244. ACM, 2002.

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

[50] S. Chen, J. Weimer, M. R. Rickels, A. Peleckis, and I. Lee. Towards a model-based meal detector for type i diabetics. In *Medical Cyber-Physical Systems Workshop 2015*, 2015.

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

[52] R. N. Clark, D. C. Fosth, and V. M. Walton. Detecting instrument malfunctions in control systems. *IEEE Transactions on Aerospace and Electronic Systems*, AES-11(4):465–473, 1975.

[53] C. Cobelli and E. Carson. *Introduction to modeling in physiology and medicine.* Academic Press, 2008.

[54] V. A. Convertino, S. L. Moulton, G. Z. Grudic, C. Rickards, C. Hinojosa-Laborde, R. Gerhardt, L. Blackbourne, and K. Ryan. Use of advanced machine-learning techniques for noninvasive monitoring of hemorrhage. *Journal of Trauma-Injury, Infection, and Critical Care*, 71(1):S25–S32, 2011.

[55] O. L. V. Costa and S. Guerra. Stationary filter for linear minimum mean square error estimator of discrete-time markovian jump systems. *IEEE Transactions on Automatic Control*, 47(8):1351–1356, 2002.

[56] C. Cowan, C. Pu, D. Maier, J. Walpole, P. Bakke, S. Beattie, A. Grier, P. Wagle, Q. Zhang, and H. Hinton. Stackguard: Automatic adaptive detection and prevention of buffer-overflow attacks. In *Usenix Security*, volume 98, pages 63–78, 1998.

[57] C. Croux and A. Ruiz-Gazen. High breakdown estimators for principal components: the projection-pursuit approach revisited. *Journal of Multivariate Analysis*, 95(1):206–226, 2005.

[58] S. Cruickshank and N. Hirschauer. The alveolar gas equation. *Continuing Education in Anaesthesia, Critical Care & and Pain*, 4:24–27, 2004.

[59] M. Cvach. Monitor alarm fatigue: an integrative review. *Biomedical Instrumentation & Technology*, 46(4):268–277, 2012.

[60] G. Dehaene and S. Barthelmé. Bounding errors of expectation-propagation. In *Advances in Neural Information Processing Systems*, pages 244–252, 2015.

[61] G. Dehaene and S. Barthelmé. Expectation propagation in the large-data limit. *arXiv preprint arXiv:1503.08060*, 2015.

[62] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1322–1328. IEEE, 1999.

[63] Bureau d'Enquêtes et d'Analyses. Final report on the accident on 1st June 2009 to the Airbus A330-203 registered F-GZCP operated by Air France flight AF 447 Rio de Janeiro–Paris. *Paris: BEA*, 2012.

[64] S. Dharmadhikari and K. Joag-Dev. *Unimodality, convexity, and applications.* Elsevier, 1988.

[65] W. Diffie and M. Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[66] M. C. F. Donkers, W. P. M. H. Heemels, N. Van de Wouw, and L. Hetel. Stability analysis of networked control systems using a switched linear systems approach. *IEEE Transactions on Automatic control*, 56(9):2101–2115, 2011.

[67] A. Doucet, N. De Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 176–183. Morgan Kaufmann Publishers Inc., 2000.

[68] V. D'silva, D. Kroening, and G. Weissenbacher. A survey of automated techniques for formal software verification. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(7):1165–1178, 2008.

[69] M. Dueker. Kalman filtering with truncated normal state variables for bayesian estimation of macroeconomic models. *Economics Letters*, 93(1):58–62, 2006.

[70] D. L. Dvorak. Nasa study on flight software complexity. *NASA office of chief engineer*, 2009.

[71] J. Edworthy and E. Hellier. Fewer but better auditory alarms will improve patient safety. *Quality and Safety in Health Care*, 14(3):212–215, 2005.

[72] B. R. Eggins. *Biosensors: an introduction.* Wiley Chichester, UK:, 1996.

[73] G. A. Einicke. Iterative frequency-weighted filtering and smoothing procedures. *IEEE Signal Processing Letters*, 21(12):1467–1470, 2014.

[74] L. El Ghaoui and G. Calafiore. Robust filtering for discrete-time systems with bounded noise and parametric uncertainty. *Automatic Control, IEEE Transactions on*, 46(7):1084–1089, 2001.

[75] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 263–270. ACM, 1999.

[76] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5):10143–10162, 1994.

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

[78] H. Fawzi, P. Tabuada, and S. Diggavi. Secure state-estimation for dynamical systems under active adversaries. In *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pages 337–344. IEEE, 2011.

[79] W. O. Fenn, H. Rahn, and A. B. Otis. A theoretical study of the composition of the alveolar air at altitude. *American Journal of Physiology*, 146:637–653, 1946.

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

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

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

[83] M. Fu and C. E. de Souza. State estimation for linear discrete-time systems using quantized measurements. *Automatica*, 45(12):2937–2945, 2009.

[84] S. A. Gadsden and S. R. Habibi. A new robust filtering strategy for linear systems. *Journal of Dynamic Systems, Measurement, and Control*, 135(1):014503, 2013.

[85] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J. Fernandez-Madrigal, and J. Gonzalez. Multi-hierarchical semantic maps for mobile robotics. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2278–2283, 2005.

[86] A. Goldsmith. *Wireless Communications*. Cambridge university press, 2005.

[87] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113, 1993.

[88] M. Görges, B. A. Markewitz, and D. R. Westenskow. Improving alarm performance in the medical intensive care unit using delays and clinical context. *Anesthesia & Analgesia*, 108(5):1546–1552, 2009.

[89] M. Green and J. B. Moore. Persistence of excitation in linear systems. *Systems & Control Letters*, 7(5):351–360, 1986.

[90] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *Robotics, IEEE Transactions on*, 23(1):34–46, 2007.

[91] J. Grossklags, N. Christin, and J. Chuang. Secure or insure?: a game-theoretic analysis of information security games. In *Proceedings of the 17th international conference on World Wide Web*, pages 209–218. ACM, 2008.

[92] B. Groza, S. Murvay, A. Van Herrewege, and I. Verbauwhede. Libra-can: a lightweight broadcast authentication protocol for controller area networks. In *International Conference on Cryptology and Network Security*, pages 185–200. Springer, 2012.

[93] V. Gupta, B. Hassibi, and R. M. Murray. Optimal lqg control across packet-dropping links. *Systems & Control Letters*, 56(6):439–446, 2007.

[94] F. Gustafsson. *Adaptive filtering and change detection*, volume 1. Wiley New York, 2000.

[95] S. Habibi. The smooth variable structure filter. *Proceedings of the IEEE*, 95(5):1026–1059, May 2007.

[96] K. Han, S. D. Potluri, and K. G. Shin. On authentication in a connected vehicle: secure integration of mobile devices with vehicular networks. In *Cyber-Physical Systems (ICCPS), 2013 ACM/IEEE International Conference on*, pages 160–169. IEEE, 2013.

[97] A. Harvey and A. Luati. Filtering with heavy tails. *Journal of the American Statistical Association*, 109(507):1112–1122, 2014.

[98] T. Hoppe, S. Kiltz, and J. Dittmann. Security threats to automotive can networkspractical examples and selected short-term countermeasures. *Reliability Engineering & System Safety*, 96(1):11–25, 2011.

[99] M. Huber. *Nonlinear Gaussian Filtering: Theory, Algorithms, and Applications*, volume 19. KIT Scientific Publishing, 2015.

[100] M. F. Huber and U. D. Hanebeck. The hybrid density filter for nonlinear estimation based on hybrid conditional density approximation. In *Information Fusion, 2007 10th International Conference on*, pages 1–8. IEEE, 2007.

[101] M. Hubert, P. J. Rousseeuw, and S. Verboven. A fast method for robust principal components with applications to chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 60(1):101–111, 2002.

[102] I. Hwang, H. Balakrishnan, and C. Tomlin. State estimation for hybrid systems: applications to aircraft tracking. *IEE Proceedings Control Theory and Applications*, 153(5):556, 2006.

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

[104] O. C. Imer, S. Yüksel, and T. Başar. Optimal control of lti systems over unreliable communication links. *Automatica*, 42(9):1429–1439, 2006.

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

[106] K. Ito and K. Xiong. Gaussian filters for nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 45(5):910–927, 2000.

[107] R. Ivanov, J. Weimer, A. Simpao, M. Rehman, and I. Lee. Early detection of critical pulmonary shunts in infants. In *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*, ICCPS '15, pages 110–119. ACM, 2015.

[108] R. Ivanov, J. Weimer, A. F. Simpao, M. A. Rehman, and I. Lee. Prediction of critical pulmonary shunts in infants. *IEEE Transactions on Control Systems Technology*, PP(99):1–17, 2016.

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

[110] S. Joshi and S. Boyd. Sensor selection via convex optimization. *Transactions on Signal Processing*, 57(2):451–462, 2009.

[111] A. Juditsky, H. Hjalmarsson, A. Benveniste, B. Delyon, L. Ljung, J. Sjöberg, and Q. Zhang. Nonlinear black-box models in system identification: Mathematical foundations. *Automatica*, 31(12):1725–1750, 1995.

[112] S. J. Julier and J. K. Uhlmann. New extension of the kalman filter to nonlinear systems. In *AeroSense'97*, pages 182–193. International Society for Optics and Photonics, 1997.

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

[114] N. Katenka, E. Levina, and G. Michailidis. Local vote decision fusion for target detection in wireless sensor networks. *Signal Processing, IEEE Transactions on*, 56(1):329–338, Jan 2008.

[115] G. Kelman. Digital computer subroutine for the conversion of oxygen tension into saturation. *Journal of Applied Physiology*, 21(4):1375–1376, 1966.

[116] G. Kitagawa. The two-filter formula for smoothing and an implementation of the gaussian-sum smoother. *Annals of the Institute of Statistical Mathematics*, 46(4):605–623, 1994.

[117] A. Konkani, B. Oakley, and T. J. Bauld. Reducing hospital noise: a review of medical device alarm management. *Biomedical Instrumentation & Technology*, 46(6):478–487, 2012.

[118] T. Kos, I. Markezic, and J. Pokrajcic. Effects of multipath reception on GPS positioning performance. In *ELMAR, 2010 PROCEEDINGS*, pages 399–402. IEEE, 2010.

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

[120] J. Kretschmer, T. Becher, A. Riedlinger, D. Schadler, N. Weiler, and K. Moller. A simple gas exchange model predicting arterial oxygen content for various FiO2 levels. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 465–468, July 2013.

[121] N. R. Kristensen, H. Madsen, and S. B. Jørgensen. Parameter estimation in stochastic grey-box models. *Automatica*, 40(2):225–237, 2004.

[122] J. Kropff, S. Del Favero, J. Place, C. Toffanin, R. Visentin, M. Monaro, M. Messori, F. Di Palma, G. Lanzola, A. Farret, F. Boscari, S. Galasso, P. Magni, A. Avogaro, P. Keith-Hynes, B. P. Kovatchev, D. Bruttomesso, C. Cobelli, J. H. DeVries, E. Renard, and L. Magni. 2 month evening and night closed-loop glucose control in patients with type 1 diabetes under free-living conditions: a randomised crossover trial. *The Lancet Diabetes & Endocrinology*, 3(12):939–947, 2015.

[123] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi. A taxonomy of computer program security flaws. *ACM Computing Surveys (CSUR)*, 26(3):211–254, 1994.

[124] D. J. Lee and M. E. Campbell. Smoothing algorithm for nonlinear systems using gaussian mixture models. *Journal of Guidance, Control, and Dynamics*, 38(8):1438–1451, 2015.

[125] T. M. Lehmann, C. Gönner, and K. Spitzer. Survey: Interpolation methods in medical image processing. *Medical Imaging, IEEE Transactions on*, 18(11):1049–1075, 1999.

[126] C. Lin and A. Sangiovanni-Vincentelli. Cyber-security for the controller area network (can) communication protocol. In *Cyber Security (CyberSecurity), 2012 International Conference on*, pages 1–7. IEEE, 2012.

[127] C. Lin, Q. Zhu, and A. Sangiovanni-Vincentelli. Security-aware modeling and efficient mapping for can-based real-time distributed automotive systems. *IEEE Embedded Systems Letters*, 7(1):11–14, 2015.

[128] Y. Liu, P. Ning, and M. K. Reiter. False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security (TISSEC)*, 14(1):13, 2011.

[129] L. Ljung and S. Gunnarsson. Adaptation and tracking in system identificationa survey. *Automatica*, 26(1):7–21, 1990.

[130] D. Luenberger. Observers for multivariable systems. *IEEE Transactions on Automatic Control*, 11(2):190–197, 1966.

[131] T. F. Lunt. A survey of intrusion detection techniques. *Computers & Security*, 12(4):405–418, 1993.

[132] W. Mader, Y. Linke, M. Mader, L. Sommerlade, J. Timmer, and B. Schelter. A numerically efficient implementation of the expectation maximization algorithm for state space models. *Applied Mathematics and Computation*, 241:222–232, 2014.

[133] L. Magni, D. M. Raimondo, L. Bossi, C. Dalla Man, G. De Nicolao, B. Kovatchev, and C. Cobelli. Model predictive control of type 1 diabetes: an in silico trial. *Journal of diabetes science and technology*, 1(6):804–812, 2007.

[134] R. Mahler. *Statistical Multisource-Multitarget Information Fusion.* Artech House, Inc., 2007.

[135] S. MartíNez and F. Bullo. Optimal sensor placement and motion coordination for target tracking. *Automatica*, 42(4):661–668, 2006.

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

[137] J. McHugh. Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Transactions on Information and System Security (TISSEC)*, 3(4):262–294, 2000.

[138] R. K. Mehra and J. Peschon. An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica*, 7(5):637–640, 1971.

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

[140] M. Milanese and C. Novara. Unified set membership theory for identification, prediction and filtering of nonlinear systems. *Automatica*, 47(10):2141–2151, 2011.

[141] M. Milanese, C. Novara, K. Hsu, and K. Poolla. The filter design from data (fd2) problem: Nonlinear set membership approach. *Automatica*, 45(10):2350–2357, 2009.

[142] H. T. Milhorn. *Application of control theory to physiological systems*, volume 1. Saunders Philadelphia, 1966.

[143] T. P. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

[144] T. Mitchell, R. Hutchinson, M. Just, R. Niculescu, F. Pereira, and X. Wang. Classifying instantaneous cognitive states from fMRI data. In *AMIA Annual Symposium Proceedings*, pages 465–469, 2003.

[145] Y. Mo, E. Garone, A. Casavola, and B. Sinopoli. False data injection attacks against state estimation in wireless sensor networks. In *49th IEEE Conference on Decision and Control (CDC)*, pages 5967–5972. IEEE, 2010.

[146] Y. Mo and B. Sinopoli. Secure control against replay attacks. In *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, pages 911–918. IEEE, 2009.

[147] Y. Mo and B. Sinopoli. False data injection attacks in control systems. In *Preprints of the 1st workshop on Secure Control Systems*, pages 1–6, 2010.

[148] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *AAAI/IAAI*, pages 593–598, 2002.

[149] T. Morimoto, T. K. Gandhi, A. C. Seger, T. C. Hsieh, and D. W. Bates. Adverse drug events and medication errors: detection and classification methods. *Quality and safety in health care*, 13(4):306–314, 2004.

[150] R. Neal. Slice sampling. *Annals of statistics*, pages 705–741, 2003.

[151] O. Nelles. *Nonlinear system identification: from classical approaches to neural networks and fuzzy models.* Springer Science & Business Media, 2001.

[152] H. Nickisch and C. Rasmussen. Approximations for binary gaussian process classification. *Journal of Machine Learning Research (JMLR)*, 9(Oct):2035–2078, 2008.

[153] H. Nickisch and C. Rasmussen. Approximations for binary gaussian process classification. *Journal of Machine Learning Research (JMLR)*, 9(Oct):2035–2078, 2008.

[154] D. Noble. Modeling the heart–from genes to cells to the whole organ. *Science*, 295(5560):1678–1682, 2002.

[155] R. Olfati-Saber. Distributed kalman filtering for sensor networks. In *Decision and Control, 2007 46th IEEE Conference on*, pages 5492–5498. IEEE, 2007.

[156] R. Olfati-Saber. Kalman-consensus filter: Optimality, stability, and performance. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 7036–7042. IEEE, 2009.

[157] M. Pajic, R. Mangharam, O. Sokolsky, D. Arney, J. Goldman, and I. Lee. Model-driven safety analysis of closed-loop medical systems. *IEEE Transactions on Industrial Informatics*, 10(1):3–16, 2014.

[158] M. Pajic, P. Tabuada, I. Lee, and G. J. Pappas. Attack-resilient state estimation in the presence of noise. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 5827–5832. IEEE, 2015.

[159] M. Pajic, J. Weimer, N. Bezzo, P. Tabuada, O. Sokolsky, I. Lee, and G. J. Pappas. Robustness of attack-resilient state estimators. In *ICCPS'14: ACM/IEEE 5th International Conference on Cyber-Physical Systems (with CPS Week 2014)*, pages 163–174. IEEE Computer Society, 2014.

[160] A. Pantelopoulos and N. G. Bourbakis. A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Transactions on Systems, Man, and Cybernetics*, 40(1):1–12, 2010.

[161] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal. Identification of hybrid systems: A tutorial. *European journal of control*, 13(2-3):242–260, 2007.

[162] P. Papadimitratos, V. Gligor, and J. Hubaux. Securing vehicular communications-assumptions, requirements, and principles. In *Workshop on Embedded Security in Cars (ESCAR)*, pages 5–14, 2006.

[163] R. J. Patton and J. Chen. Observer-based fault detection and isolation: robustness and applications. *Control Engineering Practice*, 5(5):671–682, 1997.

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

[165] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning.* The MIT Press, 2006.

[166] Citron Research. Intuitive Surgical: Angel with Broken Wings, or the Devil in Disguise? , 2013. Citron Reports on Intuitive Surgical (NASDAQ:ISRG).

[167] A. Ribeiro, G. B. Giannakis, and S. I. Roumeliotis. Soi-kf: Distributed kalman filtering with low-cost communications using the sign of innovations. *IEEE Transactions on Signal Processing*, 54(12):4782–4795, 2006.

[168] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[169] A. Roederer, J. Weimer, J. DiMartino, J. Gutsche, and I. Lee. Robust monitoring of hypovolemia in intensive care patients using photoplethysmogram signals. In *Proceedings of the 37th International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* to appear. EMBC, 2015.

[170] A. Roederer, J. Weimer, J. Dimartino, J. Gutsche, and I. Lee. Towards noninvasive monitoring of hypovolemia in intensive care patients. In *Medical Cyber-Physical Systems Workshop 2015*, 2015.

[171] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*, volume 589. John Wiley & Sons, 2005.

[172] A. H. Rutkin. ‘Spoofers’ Use Fake GPS Signals to Knock a Yacht Off Course. MIT Technology Review, August 2013.

[173] M. R. Sampford. Some inequalities on mill’s ratio and related functions. *The Annals of Mathematical Statistics*, 24(1):130–132, 1953.

[174] S. Saria, D. Koller, and A. Penn. Learning individual and population level traits from clinical temporal data. In *Proceedings of Neural Information Processing Systems*, pages 1–9, 2010.

[175] S. Saria, A. Rajani, J. Gould, D. Koller, and A. Penn. Integration of early physiological responses predicts later illness severity in preterm infants. *Science translational medicine*, 2(48):48–65, September 2010.

[176] S. Särkkä, P. Bunch, and S. J. Godsill. A backward-simulation based rao-blackwellized particle smoother for conditionally linear gaussian models. *IFAC Proceedings Volumes*, 45(16):506–511, 2012.

[177] L. Scharf and C. Demeure. *Statistical Signal Processing.* Addison-Wesley Publishing Company, 1991.

[178] L. L. Scharf and B. Friedlander. Matched subspace detectors. *Signal Processing, IEEE Transactions on*, 42(8):2146–2157, 1994.

[179] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry. Foundations of control and estimation over lossy networks. *Proceedings of the IEEE*, 95(1):163–187, 2007.

[180] B. Schneier. *Applied cryptography: protocols, algorithms, and source code in C.* john wiley & sons, 2007.

[181] S. Shafer, J. P. Rathmell, and R. Stoelting. *Stoelting's Pharmacology & Physiology.* Wolters Kluwer, 2014.

[182] D. Shepard, J. Bhatti, and T. Humphreys. Drone hack. *GPS World*, 23(8):30–33, 2012.

[183] L. Shi, M. Epstein, and R. M. Murray. Kalman filtering over a packet-dropping network: A probabilistic perspective. *IEEE Transactions on Automatic Control*, 55(3):594–604, 2010.

[184] M. Short and M. J. Pont. Fault-tolerant time-triggered communication using can. *Industrial Informatics, IEEE Transactions on*, 3(2):131–142, 2007.

[185] 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*, pages 55–72. Springer, 2013.

[186] R. H. Shumway and D. S. Stoffer. An approach to time series smoothing and forecasting using the em algorithm. *Journal of time series analysis*, 3(4):253–264, 1982.

[187] S. Siebig, S. Kuhls, M. Imhoff, U. Gather, J. Schölmerich, and C. E. Wrede. Intensive care unit alarmshow many do we need?*. *Critical care medicine*, 38(2):451–456, 2010.

[188] D. Simon and D. L. Simon. Constrained kalman filtering via density function truncation for turbofan engine health estimation. *International Journal of Systems Science*, 41(2):159–171, 2010.

[189] A. F. Simpao, E. Y. Pruitt, S. D. Cook-Sather, H. Gurnaney, and M. Rehman. The reliability of manual reporting of clinical events in an anesthesia information management system (AIMS). *Journal of Clinical Monitoring and Computing*, 26(6):437–439, 2012.

[190] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry. Kalman filtering with intermittent observations. *Automatic Control, IEEE Transactions on*, 49(9):1453–1464, 2004.

[191] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995.

[192] S. C. Smith and P. Seiler. Estimation with lossy measurements: jump estimators for jump systems. *IEEE Transactions on Automatic Control*, 48(12):2163–2171, 2003.

[193] H. Sorenson. *Kalman Filtering: Theory and Application.* IEEE Press, 1985.

[194] M. Stacey and C. McGregor. Temporal abstraction in intelligent clinical data analysis: A survey. *Artificial intelligence in medicine*, 39(1):1–24, 2007.

[195] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky. Nonparametric belief propagation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 605–612, 2003.

[196] K. Sun, K. Saulnier, N. Atanasov, G. J. Pappas, and V. Kumar. Dense 3-d mapping with spatial correlation via gaussian filtering. Submitted.

[197] J. A. K. Suykens and J. P. L. Vandewalle. *Nonlinear Modeling: advanced black-box techniques.* Springer Science & Business Media, 2012.

[198] A. Teixeira, D. Pérez, H. Sandberg, and K. H. Johansson. Attack models and scenarios for networked control systems. In *Proceedings of the 1st international conference on High Confidence Networked Systems*, pages 55–64. ACM, 2012.

[199] C. Temple. Avoiding the babbling-idiot failure in a time-triggered communication system. In *Fault-Tolerant Computing, 1998. Digest of Papers. Twenty-Eighth Annual International Symposium on*, pages 218–227. IEEE, 1998.

[200] Y. Tharrault, G. Mourot, J. Ragot, and D. Maquin. Fault detection and isolation with robust principal component analysis. *International Journal of Applied Mathematics and Computer Science*, 18(4):429–442, 2008.

[201] S. Thrun. Toward robotic cars. *Commun. ACM*, 53(4):99–106, April 2010.

[202] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT press, 2005.

[203] R. Tooher and C. Pham. *Technology Overview: da Vinci Surgical Robotic System*. Citeseer, 2004.

[204] N. A. Trayanova. Whole-heart modeling applications to cardiac electrophysiology and electromechanics. *Circulation Research*, 108(1):113–128, 2011.

[205] Trusted Computing Group. TCG Trusted Network Communications TNC Architecture for Interoperability, 2012.

[206] C. L. Tsien and J. C. Fackler. Poor prognosis for existing monitors in the intensive care unit. *Critical care medicine*, 25(4):614–619, 1997.

[207] H. J. A. F. Tulleken. Grey-box modelling and identification using physical knowledge and bayesian techniques. *Automatica*, 29(2):285–308, 1993.

[208] A. Van Herrewege, D. Singelee, and I. Verbauwhede. Canauth-a simple, backward compatible broadcast authentication protocol for can bus. In *ECRYPT Workshop on Lightweight Cryptography*, volume 2011, 2011.

[209] M. P. Vitus, W. Zhang, A. Abate, J. Hu, and C. J. Tomlin. On efficient sensor scheduling for linear dynamical systems. *Automatica*, 48(10):2482–2493, 2012.

[210] B. Vo, B. Vo, and R. P. S. Mahler. Closed-form solutions to forward–backward smoothing. *IEEE Transactions on Signal Processing*, 60(1):2–17, 2012.

[211] M. C. Vuran, Ö. B. Akan, and I. F. Akyildiz. Spatio-temporal correlation: theory and applications for wireless sensor networks. *Computer Networks*, 45(3):245–259, 2004.

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

[213] L. Y. Wang, J. Zhang, and G. G. Yin. System identification using binary sensors. *IEEE Transactions on Automatic Control*, 48(11):1892–1907, 2003.

[214] Z. Wang, D. W. C. Ho, and X. Liu. Variance-constrained filtering for uncertain stochastic systems with missing measurements. *IEEE Transactions on Automatic control*, 48(7):1254–1258, 2003.

[215] J. Weimer, S. Chen, A. Peleckis, M. R. Rickels, and I. Lee. Physiology-invariant meal detection for type 1 diabetes. *Diabetes Technology & Therapeutics*, 18(10):616–624, 2016.

[216] J. Weimer, R. Ivanov, A. Roederer, S. Chen, and I. Lee. Parameter invariant design of medical alarms. *IEEE Design & Test*, 2015.

[217] S. A. Weinzimer, G. M. Steil, K. L. Swan, J. Dziura, N. Kurtz, and W. V. Tamborlane. Fully automated closed-loop insulin delivery versus semiautomated hybrid control in pediatric patients with type 1 diabetes using an artificial pancreas. *Diabetes care*, 31(5):934–939, 2008.

[218] J. B. West. *Respiratory Physiology: The Essentials*. Lippincott Williams & Wilkins, 2012.

[219] N. Wiener. *Extrapolation, interpolation, and smoothing of stationary time series*, volume 2. MIT press Cambridge, 1949.

[220] J. Williams. *Information Theoretic Sensor Management*. PhD thesis, MIT, 2007.

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

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