2018

Security of Autonomous Systems under Physical Attacks: With application to Self-Driving Cars

Raj Gautam Dutta

University of Central Florida

Find similar works at: https://stars.library.ucf.edu/etd

University of Central Florida Libraries http://library.ucf.edu

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation


https://stars.library.ucf.edu/etd/5957
SECURITY OF AUTONOMOUS SYSTEMS UNDER PHYSICAL ATTACKS: WITH APPLICATION TO SELF-DRIVING CARS

by

RAJ GAUTAM DUTTA
B.E. Visvesvaraya Technological University, 2007
M.S. University of Central Florida, 2011

A dissertation submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in the Department of Electrical and Computer Engineering in the College of Engineering and Computer Science at the University of Central Florida Orlando, Florida

Summer Term
2018

Major Professor: Yier Jin
ABSTRACT

The thrust to achieve trustworthy autonomous cyber-physical systems (CPS), which can attain goals independently in the presence of significant uncertainties and for long periods of time without any human intervention, has always been enticing. Significant progress has been made in the avenues of both software and hardware for fulfilling these objectives. However, technological challenges still exist and particularly in terms of decision making under uncertainty. In an autonomous system, uncertainties can arise from the operating environment, adversarial attacks, and from within the system. As a result of these concerns, human-beings lack trust in these systems and hesitate to use them for day-to-day use.

In this dissertation, we develop algorithms to enhance trust by mitigating physical attacks targeting the integrity and security of sensing units of autonomous CPS. The sensors of these systems are responsible for gathering data of the physical processes. Lack of measures for securing their information can enable malicious attackers to cause life-threatening situations. This serves as a motivation for developing attack resilient solutions.

Among various security solutions, attention has been recently paid toward developing system-level countermeasures for CPS whose sensor measurements are corrupted by an attacker. Our methods are along this direction as we develop an active and multiple passive algorithms to detect the attack and minimize their effect on the internal state estimates of the system. In the active approach, we leverage a challenge authentication technique for detection of two types of attacks: the Denial of Service (DoS) and the delay injection on active sensors of the systems. Furthermore, we develop a recursive least square estimator for recovery of system from attacks. The majority of the dissertation focuses on designing passive approaches for sensor attacks. In the first method, we focus on a linear stochastic system with multiple sensors, whose measurements are fused in a central unit to estimate the state of the CPS. By leveraging Bayesian interpretation of the Kalman filter and combining it with the $\chi^2$ detector, we recursively estimate states within an error bound.
and detect the DoS and False Data Injection attacks. We also analyze the asymptotic performance of the estimator and provide conditions for resilience of the state estimate. Next, we propose a novel distributed estimator based on $\ell_1$ norm optimization, which could recursively estimate states within an error bound without restricting the number of agents of the distributed system that can be compromised. We also extend this estimator to a vehicle platoon scenario which is subjected to sparse attacks. Furthermore, we analyze the resiliency and asymptotic properties of both the estimators.

Finally, at the end of the dissertation, we make a preliminary effort to formally verify the control system of the autonomous CPS using the statistical model checking method. It is done to ensure that a real-time and resource constrained system such as a self-driving car, with controllers and security solutions, adheres to strict timing constrains.
To my parents and sisters,

for their constant

love and support.

To the love of my life,

for her endless patience

and encouragement.

To my teachers,

for guiding and

believing in me.
ACKNOWLEDGMENTS

I offer my heartfelt gratitude to my adviser Dr. Yier Jin, and co-adviser, Dr. Teng Zhang, for guiding, supporting and believing in me over the years as a Ph.D. candidate. Without their unwavering mentorship and constant motivation, my dream of earning a Ph.D. would not have been realized. I would also like to express sincere appreciation to my committee members Dr. Ronald F. DeMara, Dr. Shaojie Zhang, and Dr. Kalpathy B. Sundaram. Their constructive feedback and comments have helped me to improve my dissertation. In addition, I would like to thank Dr. Svetlana Shtrom, Dr. Raju Nagaiah, Dr. Robert Bernath, Dr. Christina Kittipatarin, Ms. Andrea Adkins, Mr. John Miner, Mrs. Jennifer McKinley, Mr. Brion Berman, and the rest of my current and past colleagues at the Technology Transfer office at the University of Central Florida’s Office of Research & Commercialization for supporting me and funding my M.S. and Ph.D. education. At the Tech Transfer office, I gained immense experience in the evaluation and protection of university Intellectual Property (IP) by assessing patentability and commercial potential. I would like to take this opportunity to thank all my teachers and professors throughout my life as a student in the U.S. and India for their constant inspiration and motivation. You’ve all made a substantial impact on my life.

I am eternally grateful to my parents, Rajen and Dhanada Dutta, and my sisters, Priyakshi Dutta and Pinakshi Dutta, for their unconditional love, support, blessings, sacrifice, and teachings of never giving up on your dreams. This work is dedicated to them. My special thanks goes to Sasha, who has contributed immensely toward my success. Words cannot describe my gratitude for the constant encouragement, and motivation – I feel blessed to have her in my life. I am grateful to aunty Marlene, uncle Ramnaresh, and Raul for welcoming me into their family and for their love and support throughout my Ph.D. I would also like to thank by brothers-in-law, Gaurav and Sandip, for their support.

As a Ph.D. student, I had the opportunity to meet and engage with people from different backgrounds. My stressful student life was well complemented by the friends I made in and out-
side of the Ph.D. lab. It is because of them that these last six and half years have been the most enjoyable time of my life. The technical and non-technical discussions during lunch breaks with Dean, Orlando, Travis, Kelvin, and Kaveh have been fun-filled and I have often left learning something new. A special thanks goes to Xiaolong, with whom I have carried out multiple projects and published many peer-reviewed scientific papers. Outside the lab, my life in Orlando would not be complete without my current and past roommates Ardalan, Saptarshi, and Soumyabrata. Conversations and hangouts with them have made my Ph.D. journey more gratifying. I would also like to thank Bala, Meghal, Sambuddha, Raghu, Chait, Ezequiel, Satadru, Debarati, Esha, Nirvani, Prachi, Shameek, Kalyan, Faisal, Saurabh, Singith, Prakrash, Kasun, Ranit, Gourab, Farrah, Rosemary, Sonia, Danielle, Jose, and Chad for the good memories.

The work in my thesis is partially supported by the National Science Foundation, Grant No. CNS-1818500.
# TABLE OF CONTENTS

LIST OF FIGURES .......................................................... xii

LIST OF TABLES ............................................................ xiv

CHAPTER 1: INTRODUCTION .................................................. 1
  1.1 Research Motivation .................................................. 1
  1.2 Physical Attacks ..................................................... 2
  1.3 Attacks on Autonomous Systems .................................... 3
  1.4 Existing Defences ................................................... 4
  1.5 Research Contributions ............................................. 5
  1.6 Dissertation Organization ......................................... 6

CHAPTER 2: LITERATURE REVIEW ........................................ 8

CHAPTER 3: ACTIVE DETECTION AND SENSOR MEASUREMENT RECOVERY . 17
  3.1 Overview ............................................................. 17
  3.2 System Model ....................................................... 18
  3.3 Attack Model ........................................................ 18
    3.3.1 Attacks on Radar Sensor ..................................... 20
  3.4 Estimation of Sensor measurements and Attack Detection ............ 22
    3.4.1 Problem Definition ........................................... 22
    3.4.2 Detection of Attacks ........................................ 23
    3.4.3 Estimation of Sensor Measurements ......................... 24
  3.5 Case Study .......................................................... 26
    3.5.1 Car-Following Model of Ground Vehicles .................... 26
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.2</td>
<td>Distributed estimation with attack</td>
<td>62</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Proof of Main Results</td>
<td>64</td>
</tr>
<tr>
<td>5.4</td>
<td>Experimental Results</td>
<td>67</td>
</tr>
<tr>
<td>5.5</td>
<td>Summary</td>
<td>70</td>
</tr>
<tr>
<td>CHAPTER 6: RESILIENT STATE ESTIMATION IN VEHICLE PLATOON</td>
<td>71</td>
<td></td>
</tr>
<tr>
<td>6.1</td>
<td>Overview</td>
<td>71</td>
</tr>
<tr>
<td>6.2</td>
<td>System Model</td>
<td>71</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Modeling Communication Network</td>
<td>72</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Closed-loop Platoon Dynamics</td>
<td>73</td>
</tr>
<tr>
<td>6.3</td>
<td>Attack Strategies</td>
<td>78</td>
</tr>
<tr>
<td>6.4</td>
<td>Problem Description</td>
<td>79</td>
</tr>
<tr>
<td>6.5</td>
<td>Methodology</td>
<td>79</td>
</tr>
<tr>
<td>6.5.1</td>
<td>Distributed estimation without attack</td>
<td>79</td>
</tr>
<tr>
<td>6.5.2</td>
<td>Distributed estimation with attack</td>
<td>84</td>
</tr>
<tr>
<td>6.5.3</td>
<td>Proof of Main Results</td>
<td>85</td>
</tr>
<tr>
<td>6.6</td>
<td>Experimental Results</td>
<td>89</td>
</tr>
<tr>
<td>6.7</td>
<td>Summary</td>
<td>92</td>
</tr>
<tr>
<td>CHAPTER 7: QUANTIFYING TRUST IN AUTONOMOUS SYSTEM</td>
<td>94</td>
<td></td>
</tr>
<tr>
<td>7.1</td>
<td>Overview</td>
<td>94</td>
</tr>
<tr>
<td>7.2</td>
<td>Attack Model</td>
<td>95</td>
</tr>
<tr>
<td>7.3</td>
<td>Framework for Quantifying Trust in Autonomous System</td>
<td>95</td>
</tr>
<tr>
<td>7.3.1</td>
<td>Functional Software Architecture of Autonomous Ground Vehicle</td>
<td>96</td>
</tr>
<tr>
<td>7.3.2</td>
<td>Safe State Estimation</td>
<td>97</td>
</tr>
<tr>
<td>7.3.3</td>
<td>Modeling of Autonomous Vehicle under Adversarial Attacks</td>
<td>98</td>
</tr>
<tr>
<td>7.3.4</td>
<td>Verification of Autonomous Ground Vehicle</td>
<td>98</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

1.1 Potential physical attacks on multi-input and multi-output autonomous CPS . 4
1.2 Vulnerabilities of an autonomous vehicle and power grid substation . . . . . 7
3.1 Car-following model with hierarchical control architecture of ACC system &
Detection, Estimation Method. . . . . . . . . . . . . . . . . . . . . . . . . . 27
3.2 Plots of Attacks and Detection, Estimation Outputs with constant deceleration
of leader vehicle . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
3.3 Plots of Attacks and Detection, Estimation Outputs with acceleration and
deceleration of leader vehicle . . . . . . . . . . . . . . . . . . . . . . . . . . 31
4.1 Idea behind estimation in Algorithm 1: After $\chi^2$ detect the attack at $k = (6-9)$
and (13-15), sensor measurements at these time points are discarded and state
estimation is done using the last good sensor value (at time points 5 and 12).
For other values of $k$, Algorithm 1 considers the measurements for estimation. 39
4.2 Car-following with ACC equipped follower whose radar measures position
($x^{(l)}$) and velocity ($v^{(l)}$) of the leader . . . . . . . . . . . . . . . . . . . . . . 44
4.3 Attack on sensor measurements of ACC system . . . . . . . . . . . . . . . . 47
4.4 Plots of relative distance (a) and relative velocity (b) between leader and fol-
lower vehicles for the attack free case. Estimation of Our Filter is as good as
the Kalman Filter (KF) and the Robust Kalman Filter (RKF) . . . . . . . . . 49
4.5 False Data Injection attack: (a,b) comparison of true and corrected values of
relative distance and relative velocity by Kalman Filter (KF), Robust Kalman
Filter (Robust KF) and Our Filter. . . . . . . . . . . . . . . . . . . . . . . . . 51
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.6</td>
<td>DoS attack: (a,b) comparison of true and corrected values of relative distance and relative velocity by Kalman Filter (KF), Robust Kalman Filter (Robust KF) and Our Filter.</td>
</tr>
<tr>
<td>5.1</td>
<td>Communication network of a distributed system of 10 agents and 12 edges</td>
</tr>
<tr>
<td>5.2</td>
<td>Performance of resilient distributed estimation algorithm in the attack free case.</td>
</tr>
<tr>
<td>5.3</td>
<td>Performance of resilient distributed estimation algorithm during attack. Agents under attack are marked with (a). Estimation error is bounded and small.</td>
</tr>
<tr>
<td>5.4</td>
<td>Performance of non resilient distributed estimation algorithm during attack. Agents under attack are marked with (a). Estimation error is large.</td>
</tr>
<tr>
<td>6.1</td>
<td>CACC vehicles platoon</td>
</tr>
<tr>
<td>6.2</td>
<td>Directed graph of homogeneous vehicle platoon. (0)-(4) represent numbering of vehicles, with (0) being the leader of the platoon.</td>
</tr>
<tr>
<td>6.3</td>
<td>Performance of resilient distributed estimator, (6.26), in the attack free case.</td>
</tr>
<tr>
<td>6.4</td>
<td>DoS attack: (a,b) comparison of distance error of attack resilient and non resilient distributed estimation algorithm.</td>
</tr>
<tr>
<td>6.5</td>
<td>FDI attack: (a,b) comparison of distance error of attack resilient and non resilient distributed estimation algorithm.</td>
</tr>
<tr>
<td>7.1</td>
<td>Functional software architecture of autonomous ground vehicle.</td>
</tr>
<tr>
<td>7.2</td>
<td>Model of Vehicular Sub-System</td>
</tr>
</tbody>
</table>
LIST OF TABLES

7.1 Summary of Trusted and Untrusted Signals of Automatons . . . . . . . . . 104
CHAPTER 1: INTRODUCTION

1.1 Research Motivation

Cyber-Physical Systems (CPS) are the result of seamless integration of computational, physical, and network components. They are becoming increasingly ubiquitous in many industrial sectors such as energy, space, health-care, agriculture, transportation, infrastructure, and manufacturing. Furthermore, incorporation of autonomous functionalities in these systems is opening the door to new applications such as search and rescue operation and entertainment. Thus, autonomous systems are enabling humans to perform tasks and explore places such as deep space, nuclear disaster sites, and deep sea, which were previously not possible.

Another example of autonomous system is self-driving vehicle, which uses sensors to sense its environment and navigate without human input. Fully automated cars are still at testing stage, but partial and conditional automation technology in vehicles have been around for last few years, whose evidence can be seen in modern cars equipped with features such as cooperative adaptive cruise control (CACC), lane keeping control (LKC), and intelligent parking assist. These vehicles are expected to help improve traffic throughput; increase safety by reducing collision; provide enhanced mobility for children, elderly, and disabled; lower fuel consumption; and improve quality of life of passengers while traveling.

Although, automated systems intend to revolutionize the way we live, many technological challenges still exist in its design, verification, and testing stages. In the case of system design, a developer has to think beyond their conventional design process and take into consideration the uncertainties arising from CPS operating in complex, unpredictable, and contested environments. During verification and testing, engineers have to deal with heterogeneity of components and size of these systems. In addition to these challenges, security is another major concern of autonomous system developers. Recent real-world attacks such as the massive power outage at...
the Ukrainian capital by the Crash Override malware, leveraging phishing emails to cause multiple component failure at a German steel mill, and spoofing of sensors (radar, Lidar, camera, GPS) has raised the need for designing secure CPS.

When it comes to vehicles, addition of Advanced driver-assistance systems (ADAS) has increased the number of potential attack surfaces and have made the system vulnerable to cyber threats. Furthermore, attacks such as jamming and spoofing of vehicle sensors were also shown to be possible, indicating that exclusive reliance on cyber-security techniques for CPS is insufficient [1]. Consequently, in this thesis, we develop security solutions for detection and mitigation of sensor (physical) attacks in autonomous systems with primary emphasis on vehicles. Furthermore, we provide a preliminary approach to formally verify (using the statistical model checking approach) various sub-systems (specifically controllers) of the vehicle against timing constrains. It is done to ensure that a real-time and resource limited system such a self-driving car operates safely by complying with timing restrictions.

1.2 Physical Attacks

Time-varying sensor signals carry information that are crucial for correct operation of safety-critical systems such as medical devices, nuclear power plants, and vehicles. Any attack on these signals could result in undesired consequences. In the dissertation, we refer to such attacks as “physical” to differentiate them from “cyber-attacks”.

Physical attacks can be categorized into invasive and non-invasive types [2]. In the case of invasive attack, an adversary can physically tamper with sensor’s hardware components and software. Whereas in the case of non-invasive attack, an adversary cannot physically alter the components of a sensor, but can (remotely) inject malicious signals after observing interaction of the sensor with the physical environment. Tamper resistant hardware and secure programs could be built to protect the system from invasive attacks. However, defending against non-invasive
attacks can be challenging. As the attack alters analog signals of the sensor, operating the system on corrupted measurements could result in catastrophic consequences. Thus, new solutions are required for securing systems from such sophisticated attacks.

The non-invasive physical attack can be further classified into “passive” and “active” types. In a “passive” attack an adversary observes the sensor signal to infer information. An example being the side channel attack on cryptosystem, where an adversary records the frequency of electrical/electromagnetic signals emitted by the cryptographic device to extract secret information. On the contrary, an “active” attack involves manipulation of sensor’s physical signal by an attacker, with the intention of harming, misleading, or degrading the performance of the system. An example of such attack is spoofing/false data injection (FDI) of radar signals of a vehicle’s adaptive cruise control (ACC) unit, where the attacker uses a special device to inject signals with the objective of providing false distance information of neighboring cars to the vehicle. Other types of active attack includes denial of service (DoS), where an adversary jams the sensor or communication signals to prevent the system from receiving measurements and replay, where the system observes valid measurements at incorrect time.

1.3 Attacks on Autonomous Systems

Inclusion of intelligence has seen the application of CPS in many new areas. These complex systems use sensors to gather information about the physical world. Then, a decision making unit consisting of modern controllers and digital electronic components processes the available information to determine the appropriate action for the system. Subsequently, the actuators of the system implements the control commands to carry out the action.

However, these advancements have increased vulnerability and security weaknesses in the CPS, example of which can be seen in the Figure.1.2. Moreover, these systems rely on sensor’s information for operation, which is another point of attack. Although, encryption of such infor-
Information can prevent privacy attacks, DoS and spoofing attacks are still possible due to sensor’s interaction with the physical world. Manipulating the environment around the CPS or corrupting the signals by hacking devices lead to new threat models, which were not considered in traditional cyber-security.

Thus, understanding how an attacker might modify and corrupt the sensor signals becomes crucial in assessing the dependability and security of autonomous CPS. For this purpose, control oriented or system-level methods have been developed, which models the behavior of attack and its influence on dynamics of the system. Figure 1.1 shows various physical attacks in a control oriented framework. Due to the threat posed by a successful attack on CPS, many recent research endeavors have been aimed at characterizing these attacks and building effective countermeasures [3–9].

![Diagram of Physical Attacks](image)

**Figure 1.1:** Potential physical attacks on multi-input and multi-output autonomous CPS

### 1.4 Existing Defences

In systems theory, robust control algorithms were developed to mitigate environmental uncertainties in autonomous systems. They could achieve robust performance and/or stability in the presence of bounded uncertainties/errors. However, these methods were not designed to overcome unbounded errors generated by adversarial attacks. To defend against attacks, approaches such as encryption, secure flashing, anomaly and intrusion-detection have been developed that detect and
prevent cyber and physical attacks in autonomous CPS [3, 10–15]. Encryption and secure flashing methods were developed for protecting the system from cyber attacks carried over communication networks.

To protect sensors of critical applications from invasive physical attacks, measures such as placing the sensor in secure location to increase difficulty of its access have been taken. While this step can secure the sensor’s hardware, the analog signals transmitted by active sensors such as radar to gather information about the environment are still accessible to an attacker. Consequently, an adversary can characterize the signals spectral properties and reflect a counterfeit signal, which can affect functioning of a system relying on sensors measurements. Such physical attacks are referred to here as non-invasive. Any countermeasures taken after the corrupted signal has been sampled and digitized in the system domain will not be effective. To overcome this problem, both active and passive methods have been developed. In the active approach, additional hardware to perform challenge authentication of sensor measurements was done in [16]. Then, by using a chi-square detector, spoofing attempts on sensors were detected. However, no effort was made to recover the CPS from attacks. Most of the recent advancements in countermeasure development have been on passive approaches [3, 11–15]. They are system-level solutions that entails design of a failure detector for finding abnormal system behavior and an attack resilient estimator for recovering system states. However, many of these approaches are not suitable for a real-time and resource constrained system.

1.5 Research Contributions

The contribution of this dissertation is multifold covering both algorithm design and theoretical analysis as well as approach for quantification of trust in autonomous CPS.

- Algorithms for detection and automatic recovery of autonomous CPS from attacks: We develop an active and multiple passive algorithms to detect physical attacks (FDI or DoS) under various
scenarios and to minimize their effect on the internal state estimates of the system. In the active approach, we leverage a challenge authentication technique for detection of attacks and use a recursive least square estimator for recovery of system from attacks. The passive approaches involves design of attack resilient estimators for centralized and distributed systems. We also analyze the asymptotic performance of the estimators and provide conditions for resilience of the state estimate.

- Quantifying trust in autonomous system: We first model the CPS using Networks of Priced Time Automata (NPTA). Then, we use probabilistic weighted computational tree logic (PWCTL) to specify timing properties of the controller of the CPS. By using a statistical model checker we ensure that the system adheres to the strict timing constrains.

1.6 Dissertation Organization

The rest of the dissertation is structured as follows: In Chapter 2, we review existing literature on physical attacks and existing defenses. In Chapter 3, we present an active attack detection method along with sensor measurement recovery algorithm. Chapter 4 proposes a Chi-Squared based passive attack detection method and an attack resilient estimator. In Chapter 5, we extend our scope to distributed systems and design a distributed estimator to recover the system from sensor attack. Chapter 6 is about application of our general distributed estimation algorithm toward sparse sensor attacks on vehicle platoon. In chapter 7, we provide an initial insight into a formal approach for quantifying trust of controllers of autonomous system and concluding remarks are in Chapter 8.
(a) Attack surfaces of a connected autonomous vehicle

(b) Attack surfaces of a electric power substation

Figure 1.2: Vulnerabilities of an autonomous vehicle and power grid substation
 Attacks on a CPS can be carried out either via physical access/invasive or remotely/non-invasive. There are several real-world examples where physical access led to execution of the attack [17–19]. Karnouskos et al. [17] studied various aspects of the Stuxnet worm that specifically targeted Siemens WinCC/PC S7 SCADA control software with the goal of modifying the behavior of Programmable Logic Controller (PLC) devices (manufactured by Vacon (Finnish vendor) and Fararo Paya (Iran)) connected to it. The worm was inserted in the Windows computer hosting the SCADA software via a USB flash drive. It manipulated the rotation frequency of nuclear centrifuges by injecting false control commands and kept itself stealth. The design of Stuxnet is not domain-specific and can be readily customized for other types of CPS. Koscher et al. [18] demonstrated attack on automotive Electronic Control Units (ECU) via the On Board Diagnostics (OBD-II) port of the vehicle. With the help of this port, an attacker (such as a mechanic, a person renting the car, a valet, or the car owner) can insert a malicious component into the Controller Area Network (CAN) of the car. A malware inserted via this component can enable an adversary to compromise various safety-critical components such as brake and engine. Liu et al. [19] analyzed false data injection attacks on electric power grids. In their threat model, an adversary can exploit knowledge of the configuration of the target power system to launch attacks, which can bypass detection techniques and introduce arbitrary errors in certain state variables. In all these attacks, an adversary required physical access to the Windows machine hosting SCADA software to plug the USB flash drive, access to the vehicle to connect to OBD-II port, and knowledge of the target power system, which is usually kept secret by power companies at control centers.

As carrying out attacks that require physical access has been viewed as unrealistic, attackers have started leveraging wireless communication network and sensing unit of the CPS to launch attacks remotely/non-invasively [1, 20–24]. We refer to the remote attacks that compromise software or network components of the system as cyber [20, 21, 24] and those that compromise sensor
measurements as *physical* [1, 22, 23, 25]. Checkoway *et al.* [20] provided comprehensive analysis of remote exploitation via a broad range of attack vectors (Wi-Fi, Bluetooth, cellular radio) present in modern automobiles. These attack vectors use various wireless communication wavelengths to get access of in-vehicle networks such as CAN, Local Interconnect Network (LIN), Media Oriented System Transports (MOST), and FlexRay [26]. As the engine control units of the vehicle communicate over CAN bus, most of the attacks are targeted toward it. Control of the CAN bus allows the attacker to take control of most of the critical functionalities of the vehicle. Anderson and Fuloria [21] demonstrated DoS attack on energy utilities where an adversary can remotely control the smart meters to disrupt electric and gas supplies. In [24], Miller and Valasek exploited vulnerabilities in the Jeep Cherokee’s telematic unit, which provided a cellular connection to the vehicle, and showed that they could take complete control over the vehicle via the Internet. They were able to control transmission, steering, brakes, heating system, radio, windshield wipers, and the digital display of a vehicle running on a highway.

Another form of remote attack, which we refer to as *physical*, target the information gathered by sensing unit of the CPS [1, 22, 23, 25]. The remote physical attack can be of two types: *passive* and *active*. Side-channel attack, where an adversary probes or observes the execution of the system to learn its behavior (timing, EM, acoustic etc.) is an example of passive attack. Faruque *et al.* [25] has shown that acoustic emanations of a desktop 3D printer can be used to reconstruct the printed object’s geometry. They reported an average accuracy of 78.35 % for axis prediction and an average length prediction error of 17.82 %.

The active remote attacks comprises of DoS (e.x. Jamming) and FDI. A DoS attack disrupts sensors data gathering capabilities and FDI attack manipulate measurements to change operational behavior of the system. Now, a CPS has many sensors (radar, lidar, Global Positioning System (GPS), camera, ultrasonic, and inertial measurement unit (IMU)) for collecting data of its internal and external environment [27]. Compromising the data gathered by these sensors can impact the control unit of the system. Bhatti and Humphreys [22] demonstrated and analyzed a decep-
tion attack in which an adversary covertly controlled a maritime surface vessel by broadcasting counterfeit civil GPS signals. Petit et al. [1] demonstrated remote jamming, replay, relay, and spoofing attacks on Camera and LIDAR sensors of autonomous vehicle. Shoukry et al. [23] injected magnetic fields to spoof the wheel speed of vehicles by placing a magnetic actuator near the Anti-locking Braking System’s (ABS) wheel speed sensor. They used the same physical property (magnetic field) as that intended to be sensed through the sensing channel of the target sensor for their spoofing attack. Thus, the cyber and physical attacks tends to compromise confidentiality, availability, and integrity properties of the CPS.

Recently, a lot of attention has been given toward the development of encryption, intrusion detection, game-theoretic, control-theoretic, and machine learning solutions, which can detect and protect CPS from remote cyber and physical attacks [3, 10–16]. We refer to all these methods as passive as they do not require modification of the hardware or incorporation of new devices. Woo et al. [10] developed a security protocol for CAN bus to protect it from cyber attack via a malicious application on driver’s smartphone. As CAN bus does not ensure confidentiality and authentication of messages broadcasted over the network, an adversary can easily eavesdrop on data or launch a replay attack. To overcome this problem they encrypted the messages before being broadcasted over the CAN bus to enforce confidentiality. Also, authentication of data was done during vehicle’s normal operation. Zimmer et al. [11] developed time based intrusion detection methods, which detected execution of unauthorized instructions in real-time CPS. They compared timing metrics against worst-case bounds to detect security breaches that occurred due to one or more network node of the system being compromised or an attacker being able to authenticate a node of the local network under their control. Mo and Sinopoli [14] used a game theoretic approach to detect false data injection attacks in CPS. In [12], a learning mechanism was used to construct a set of invariant called “safety envelope” from collected sensor data. When, the system violated these constrains, an alarm was raised depending on whether it was an attack or noise.

Among the existing countermeasures, control-theoretic or system-level methods are gain-
ing traction among researchers [3, 8, 13, 15, 16, 28–30]. They involve design of failure detectors and robust and resilient state estimators, which can withstand noisy and malicious sensor and actuator measurements under various attack and system scenarios. Fawzi et al. [3] corrected the maximum number of errors (due to sensor and actuator attacks) of a linear time-invariant system (without noise) by first framing the state estimation problem as $\ell_0$-optimization problem and then relaxing it to “$\ell_1/\ell_r$” norm. For the same setup, computationally efficient event-triggered projected gradient descent and event-triggered projected Luenberger observer algorithms were developed for reconstruction of system states in [8]. However, both [3, 8] consider that only half of the sensors could be tampered for exact estimation of states. For a continuous-time linear system with additive bounded disturbance and additive bounded measurement noise, a finite time Gramian-based and an asymptotic observer-based state estimators were developed in [15]. However, they required a large number of potential estimates for reconstructing the correct state.

The set of literature relevant to our work consider additive zero mean, Gaussian white noise with unbounded attack signal in the system model [13, 28–30]. Forti et al. [28] designed a computationally expensive hybrid Bernoulli filter (in Bayesian Framework) to simultaneously detect attacks (signal, packet substitution, and extra packet injection) and estimate system states. Their filter could recursively update in real-time the joint posterior density of the attacks and of the state vector, provided all measurement were available up to that time. A Robust Kalman filter (RKF) for estimating states during sparse sensor disturbances was developed in [29]. They modified the measurement update equation of the standard Kalman filter with the solution of $\ell_1$-based convex optimization problem. However, they did not provide any optimality guarantee. Mishra et al. [30] used a bank of Kalman filters for secure state estimation of noisy linear dynamical system subjected to sparse data injection attack. They identify the subset of sensors that are not attacked by using a block residue test and use their outputs to calculate the secure estimate. However, they assume that all sensors data are not corrupted and the number of Kalman filters used in their method is dependent on the number of attacked sensors. Pajic et al. [13] developed a robust state estimator
for systems with attack, noise, and modeling errors. They mapped their estimation problem into $l_\infty$ optimization problem and computed a worst-case bound on the state estimate error in the presence of additive modeling errors with known bounds. However, their optimization problem was NP-hard.

Efforts have also been made to protect distributed CPS such as power grid, vehicle platoon, and smart buildings, from attacks. Early researchers developed centralized and decentralized attack resilient filters, which requires aggregation of sensor measurements at a particular location or at all the components of the system for state estimation [31–33]. Due to the computational inefficiency of both centralized and decentralized approaches, distributed state estimators were designed, where a component (or agent) of the distributed CPS asymptotically estimated the system state based on partial information of the state from its neighbors [34–38]. Most of these methods are variation of the Distributed Kalman Filter (DKF) of [34]. The earliest DKF algorithm solved the estimation problem in two steps: in the first step, a dynamic average-consensus filter was used for fusion of sensor and covariance data and in the next step Kalman filter update rules were used for recursively estimating the states. Convergence of the DKF depended on the topology of the communication network. Subsequently, single time scale strategies were developed for the DKF.

Until recently, very few attempts were made on designing attack resilient distributed state estimators [39–41]. Khan and Stankovic [39] proposed attack detection and single message exchange state estimation methods for a compromised communication scenario. Their estimator relied on statistical consistency of nodal and local data sets and physical-layer feedback. Matei et al. [40] designed a multi-agent filtering scheme in conjunction with a trust-based mechanism to secure the state estimates of power grid under false data injection attack. In their approach, an agent of the grid compute local state estimates based on their own measurement and of their trusted neighbors. However, both [39, 40] did not provide any theoretical guarantees of their method. Mitra and Sundaram [41] developed a secure distributed observer for the Byzantine adversary model, where some nodes of the network were compromised by an adversary. Prior to state estimation,
they decomposed the linear system model using Kalman’s observability decomposition method. Then, Luenberger observer at each node estimated the states corresponding to detectable eigenvalues. The undetectable portions of the states at each node were estimated using secure consensus algorithm, which used measurements of well-behaving neighboring nodes. However, their method requires the network to be highly connected to mitigate the effect of a small number of adversarial nodes.

Specific approaches have also been developed for application such as vehicle platoon. Sajjad et al. [42] designed an insider attack aware sliding mode control scheme that uses only local sensor data and a decentralized attack detector to reduce the number and severity of collision in a platoon. Their solution makes the assumption that the bidirectional platoon is homogeneous with all cars sharing the same control law. They consider a single attacker that modifies the control law of a vehicle to induce an oscillatory behavior in the platoon. Kafash et al. [43] proposed an approach for limiting the capabilities of an attacker by imposing artificial bounds on the control inputs that drive the system. With the help of these bounds, they could restrict the system from reaching unsafe states during sensor or actuator attacks. They used convex optimization to quantify the reachable states (good states) and developed methods to design new bounds, which can prevent the reachable set from entering the set of unsafe or dangerous states. The main advantages of the various system-level methods are that they allow the CPS to use the same old controllers and enable recovery of the system from attack.

In [16], an active approach for remote physical attack detection was proposed, which required additional hardware to perform challenge authentication of sensor measurements. Then, by using a chi-square detector, they detected spoofing attempts on sensors. However, no effort was made to recover the CPS from attacks.

As CPS are resource-constrained systems, ensuring the controllers and security solutions follow the strict timing, communication, and memory constrains is important. Furthermore, these solutions should respect the semantic gap between design and implementation for correct opera-
tion. Consequently, Lin et al. [44], Pasqualetti and Zhu [45] and Zheng et al. [46] proposed frameworks that analyse the impact of security solutions and consider the gap between controller design and implementation. Lin et al. [44] analysed the impact of message authentication mechanism with time-delayed release of keys on real-time constraints of a ground-vehicle. Such a security solution was developed to protect Time Division Multiple Access (TDMA)-based protocol, which is used in many safety-critical systems such as automobile and avionics electronic systems because of their more predictable timing behavior. To ensure the increased latencies due to delayed key release did not violate timing requirements, an algorithm to optimize task allocation, priority assignment, network scheduling, and key-release interval length during the mapping process from the functional model to the architectural model, with consideration of the overhead was developed. This algorithm combined simulated annealing with a set of efficient optimization heuristics. However, their approach did not consider the impact of their security solution on sampling periods and control performance. Pasqualetti and Zhu’s [45] method could analyze control performance, security, and end-to-end timing of a resource-constrained CPS under network (cyber) attack that can compromise systems privacy (confidentiality). They have also quantified inter dependency between the three system objectives by means of a minimal set of interface variables and relations. In their work, they have considered an adversary that has complete knowledge of the system model and can reconstruct system states from measurements. As a first step, the physical plant was modeled as a continuous time LTI system. The control input was determined using an output-based control law. A relationship was established to show that the control performance improved with reduced sampling time. Next, resiliency of the encryption method, protecting messages transmitted by sensor to controller was evaluated. It was observed that the encryption method increased the sampling period thereby degrading control performance. While implementing the control function on a CPS platform, the end-to-end delay was calculated by incorporating time incurred during sensing, computation, and communication. During development of the scheduling algorithm for the system, it was ensured that the measured delay was within the sampling period. Based on their analysis,
they concluded that the control and the security algorithms should be designed based on the implementation platform so as to optimize performance and robustness. Zheng et al. [46] quantify the impact of their security solution on control performance and schedulability. They also analyzed the tradeoffs between security level and control performance while ensuring the resource and real-time constraints were satisfied. For demonstration, a CPS with multiple controllers that share computation and communication resources was considered. A controller, which was modelled as a control task, processed information collected from sensors on a shared resource and commanded actuators to carry out task. To prevent attackers from eavesdropping on the communication medium for obtaining system’s internal information, messages from sensors were encrypted. The decryption of these messages were modeled as task. Each of these tasks were given an activation period and worst case execution time. In the system, the control tasks competed for computation resources whereas as messages competed for communication resources. Incorporating the message encryption mechanism introduced resource overhead that impacted schedulability and control performance. To avoid this issue, they framed an optimization problem where control performance (a function of control task period) was the objective function and security level, computation resource, communication resource, and end-to-end latency were constraints. By varying the security level (function of messages to be encrypted), they ensured that the system achieved optimal control performance and platform schedulability.

Verification of autonomous ground vehicles has been also receiving increasing attention over the years [47–50]. Stursberg et al. [47] modeled the cruise control system of vehicle using hybrid automata with nonlinear continuous dynamics and polyhedral guard and invariant sets. Then, they used counterexample-guided verification approach to ensure the cruise control system of two cars in one lane was operating correctly to avoid collision. However, their method could not be scaled to arbitrary number of cars. Loos et al. [48] used quantified hybrid program for modeling distributed car control system, where every car was controlled by adaptive cruise control. They used the automated theorem prover- KeYmaera - to prove that the control model satisfied the
collision avoidance protocol for arbitrary number of cars on a street. Wongpiromsarn et al. [49] proposed the framework, called periodically controlled hybrid automata, for modeling control systems in which inputs to actuators were given after a fixed time unit. They verified safety and progress properties of the planner-controller sub-system of an autonomous ground vehicle using sum of square decomposition and semi-definite programming. Althoff et al. [50] used reachability analysis for safety verification of evasive maneuvers of autonomous ground vehicle with constant velocity and under uncertainties.

However, these verification methods do not consider an autonomous ground vehicle under adversarial attacks [47–50]. To address this scenario, many attack detection and prevention methods were developed for ground vehicles [51–53]. Zheng et al. [51] analyzed security and safety of in-vehicle control system and vehicle-to-vehicle communication network using hybrid modeling, formal verification, and automated synthesis techniques. They considered cooperative adaptive cruise control (CACC) as their test case and used an automated theorem prover to prove a collision avoidance property under time delay attack on vehicle communication network. Mundhenk et al. [52] used probabilistic model checker - PRISM, for quantifying security vulnerability (in terms of confidentiality, integrity and availability) of automotive architecture at design-time. Before using the model checker, they transformed the automotive communication architecture which consists of ECUs, heterogeneous bus system, and intra-vehicle networks into Continuous-Time Markov Chain with transitions given by exploitability rate and patching rate. However, they did not consider safety of an operating vehicle under adversarial attack.
CHAPTER 3: ACTIVE DETECTION AND SENSOR MEASUREMENT RECOVERY

3.1 Overview

In this chapter, we propose a solution for detecting attacks on analog signals of sensors such as radar, lidar, or ultrasonic. We also provide an algorithm for estimating sensor measurements when it is under attack. Our detection method and estimation algorithm are implemented before the measurements enter the digital domain of the system. Thus, we are able to defend the system before corrupted measurements could effect its operations. The main contributions of this chapter are summarized as follows.

- We use a challenge response authentication (CRA) method for detecting attacks on sensors. This method does not require any redundant sensors and it does not produce any false positives or false negatives.

- We develop an algorithm to estimate correct sensor measurements using recursive least square (RLS) approach, after an attack has been detected. Consequently, it enables the autonomous CPS to recover from an attack.

- We also modify the standard intelligent-driver car following model (IDM) by integrating it with an adaptive cruise control (ACC) system.
3.2 System Model

In this section, the dynamics of autonomous CPS is modeled as a discrete-time linear time-invariant (LTI) system without process noise and is given by the following equations:

\[ x_{k+1} = Ax_k + Bu_k \]  
\[ y_k = Cx_k + v_k \]

where, \( x_k \in \mathbb{R}^n \) is a real-valued system state vector at time \( k \), \( u_k \in \mathbb{R}^m \) is a real-valued control input at \( k \), and \( y_k \in \mathbb{R}^p \) is a real-valued sensor measurement vector at time \( k \). \( v_k \sim N(0, R) \) is Gaussian measurement noise with zero mean and covariances \( R = E[v_k v_k^T] \). Also, \( A \) is the system matrix, \( B \) is the control matrix, and \( C \) is the output matrix. We assume that matrices \( A, B, C \) are known.

3.3 Attack Model

Attack on an autonomous CPS, such as a ground-vehicle, can be carried out either via physical access or remotely [1, 16, 54]. Here, we consider remote attacks, as attackers getting physical access of the vehicle's internal operation may be an infeasible assumption. Remote exploitation can be carried out either on communication networks or on sensing units of the vehicle [1, 16, 54]. In this chapter, we have considered attack on the sensing unit.

An autonomous vehicle has many sensors for collecting data of its internal and external environment. Compromising the data gathered by these sensors can impact the decisions made by the motion control units of the vehicle. By following the attack models described in [1, 16], we assume that the non-invasive attacker targets the external sensors, has limited knowledge of sensors firmware or software, and is in the vicinity of the attacked system. We also assume that they use resources that can conceal their remote spoofing attacks. Furthermore, the attack can be mounted
on the target vehicle while its stationary or in motion.

Under these assumptions, we consider an autonomous vehicle, whose sensors such as ultrasonic, radar, or lidar are under Denial of Service (DoS) attack or delay injection based spoofing attack. As delay is an inherent property of received signals of sensors and is essential for making various measurements, distinction between original signal and received signal based on timing characteristics is not possible. Thus, to find the presence of such attacks, a detection method should be developed, which can differentiate original signals from the delay injected counterfeit signals and it should have high sensitivity.

The autonomous CPS model under sensor attack can be represented using the following equations:

\[ x'_{k+1} = Ax'_k + Bu'_k \]  \hspace{1cm} (3.3)
\[ y'_k = Cx'_k + y^a_k + v_k \]  \hspace{1cm} (3.4)

where,

\[ y^a_k = \begin{cases} 0 & \text{if delay injection attack}, \\ r \in \mathbb{R}^p & \text{if DoS attack} \end{cases} \]

In case of delay injection attack, \( y'_k \) is a counterfeit signal, which is similar to the normal signal except with a longer delay. An attacker can use inexpensive hardware and adversarial machine learning techniques to analyze the actual sensor signals and generate counterfeit signals. Whereas, in DoS attack, correct sensor measurements are suppressed with a stronger signal, \( y^a_k = r \), by method such as jamming. As such the system receives corrupted sensor measurements \( y'_k \). Next, we describe a model of radar sensor of a vehicle and also explain the effects of DoS attack and the delay injection attack on the sensor.
3.3.1 Attacks on Radar Sensor

We consider a model of a long-range automotive radar, which uses mm-wave for object detection. Such a radar measures distance and relative velocity to the target object. It has a carrier frequency of 77 GHz and is operated as Frequency Modulated Continuous Wave (FMCW) system for better sensitivity and simplicity of design [55].

For detecting an object, a mm-wave radar continuously transmits triangular frequency modulated waveforms. Due to Doppler effect, the signals received from the reflecting object (moving or stationary) by the radar are shifted in frequency from the transmitted signal by a delay, $\tau$. Subsequently, the received signal is mixed with a portion of the transmitted signal in a mixture of the radar’s FMCW system. From the mixed output signal, two beat frequencies are extracted: $f_{b+}$ is for the positive portion of the slope and $f_{b-}$ is for negative portions of the slope of the mixed signal [55]. These two frequencies are determined using the following equations:

$$f_{b+} = \frac{2d}{c} \frac{B_s}{T_s} - \frac{2d}{\lambda}$$  \hspace{1cm} (3.5)  

$$f_{b-} = \frac{2d}{c} \frac{B_s}{T_s} + \frac{2d}{\lambda}$$  \hspace{1cm} (3.6)

where $d$ is distance to the target object in meters, $c$ is the speed of light, delay $\tau = \frac{2d}{c}$, sweep bandwidth $B_s = 150$ Mhz, sweep time $T_s = 2$ msec, and wavelength $\lambda = 3.89$ mm.

From these two beat frequencies, distance and relative velocity to the object can be calculated as following:

$$d = \frac{cT_s}{4B_s}(f_{b+} + f_{b-})$$  \hspace{1cm} (3.7)  

$$\Delta v = \frac{\lambda}{4}(f_{b-} - f_{b+})$$  \hspace{1cm} (3.8)

We can also calculate the power of the signal received from the target object by using the
following equation:

\[
P_r = \frac{P_t G \lambda^2 \sigma_s}{(4\pi)^3 d^4 L}
\]

(3.9)

where, \( P_r \) is the received signal power; \( P_t = 10 \, \text{mW} \) is the maximum transmitted signal power; \( G = 28 \, \text{dBi} \) is the antenna gain; \( \sigma_s \) is the scattering cross section of the target object located at the distance \( d \); and \( L = 0.10 \, \text{db} \) is the radar system losses.

To implement our challenge response authentication method, we modify the modulation unit of the radar to enable probing of the environment at random times. Subsequently, by implementing the Algorithm 2, we can detect the attack before it enters the domain of automotive control system.

**Denial of Service attack.** In this case, an attacker can use a self-screening jammer to transmit a signal with more power than the original signal received by the ACC’s mm-wave radar. The power of the jamming signal is given by the following equation:

\[
P_{\text{jammer}} = \frac{P_J G_J \lambda^2 G B}{(4\pi)^2 d^2 B_J L_J}
\]

(3.10)

where, \( P_J, G_J, B_J, L_J \) represents jammer’s peak power, antenna gain, operating bandwidth, and losses respectively. \( B \) is the operating bandwidth of the mm-wave radar. Rest of the parameters, \( \lambda, G, d \), have same value as that of the radar. To carry out a successful attack, the following power ratio should be less than unity.

\[
\frac{P_r}{P_{\text{jammer}}} = \frac{P_t \sigma_s B_J L_J}{4\pi P_J G_J d^2 B L}
\]

(3.11)

When the condition is valid, the radar will start receiving corrupted measurements, which can lead to vehicular collision.
**Delay injection attack.** In this case, an attacker replays a counterfeit signal with additional physical delay ($\tau$) to create an illusion that the object is further away than the actual distance. To carry out this attack, an adversary should have special hardware, which generates a signal with similar characteristics as the original reflected signal, except with more delay. Such a reflected signal will change values of beat frequencies (Eqn. 3.5, 3.6) extracted by the radar’s receiver, which then will effect distance and velocity measurements. In the absence of true distance measurements, a vehicle will not be able to slow down or accelerate as desired.

### 3.4 Estimation of Sensor measurements and Attack Detection

We develop Algorithm 2 for attack detection and sensor measurement estimation. The detection method is based on a challenge response authentication technique and is particularly useful for detecting delay injection attacks, where the attacked signal has the same characteristics as that of the original signal, except with longer delay. Such an increase of delay in sensor measurements can disrupt normal operation of the system. Moreover, this method is also capable of detecting DoS attacks. Once an attack is detected, our estimation method, which is based on recursive least square algorithm (Algorithm 1), predicts sensor measurements for the duration of attack. With these estimated measurements, the controller determines an optimal control input for the CPS system. Unlike [16], our method enables recovery of the system from attack. Such a recovery mechanism is particularly useful for CPS that cannot be brought to a hault. Before explaining our method in details, we define the attack detection and estimation problem.

#### 3.4.1 Problem Definition

Given a close loop system whose control input, $u_k$, is determined according to sampled sensor measurements, $y_k$ and the sensors are under DoS or delay injection attack producing corrupted measurements $(y'_k, \ldots, y'_n)$ over a finite interval $[k_1, k_n], k_1 \neq 0, k_n < \infty$, we want to de-
sign a detector that can find the presence of an attack and an estimator that can predict outputs \((\hat{y}_k, \ldots, \hat{y}_{kn})\) during the duration of attack.

### 3.4.2 Detection of Attacks

For the challenge response authentication technique, we consider sensors such as radar, ultrasonic, lidar. Such sensors probe the environment with self-generated signals for gathering information. To incorporate a challenge on transmitted/probing signals of sensors, we modify its modulation system with a pseudo-random binary modulation unit. As such, the transmitted signal, \(p'(t)\), of the sensor is modulated as

\[
p'(t) = m(t) \cdot p(t), \quad m(t) \in [0, 1]
\]

where, \(m(t)\) is the binary modulation signal and \(p(t)\) is the actual signal. Now, the modulated signal \(p'(t)\) of the sensor changes values according to the following conditions:

\[
p'(t) = \begin{cases} 
0 & \text{if } m(t) = 0 \text{ for } t \in T_c, \\
p(t) & \text{if } m(t) = 1 \text{ for } t \in \mathbb{N} \setminus T_c.
\end{cases}
\]

where, \(T_c\) is the set of time points \(t\) at which outgoing probing signals are suppressed. At the corresponding sample time point \(k\), received sensor measurements, \(y'_k\) (sampled signal), of the receiving unit of an sensor should be zero. For all the other time points, the modulated signal is same as the actual signal and the receiver produces a non-zero output. Only in the case of an attack, the receiver gives a non-zero output for \(p'(t) = 0\). By comparing the expected output of the receiver at sampled time points \(k \in T_k \subseteq T_c\) against the actual output, we can detect the presence or absence of an attack. A DoS attack can be easily detected by this method as the receiving unit of sensor will produce a large non-zero output \((y'_k = r)\) when no signal was transmitted at time points
t \in T_c$. However, in the case of delay injection attack, a smart adversary attempting to transmit counterfeit signals with additional delay could conceal their attack when the modulated signal $p'(t)$ is zero at time points $T_c$. Now, due to the unavoidable time delay incurred by adversaries hardware, the time required to carry out the attack is always more than zero. As a result, attackers spoofing attempts can be detected using a simple detector that compares value of expected signal with the received signal.

Our method can also be used on passive sensors, but with additional hardware. Lines 7-9 of the Algorithm 2 represents the steps of our detection method.

### 3.4.3 Estimation of Sensor Measurements

After detecting the attack, we estimate future values of the sensor by using the Recursive Least Square (RLS) estimation method as shown in Algorithm 1 [56]. The RLS algorithm estimate recursively in time the measurement values, \( \{w_0, w_1, \ldots, w_n\} \).

**Algorithm 1 Recursive Least Square Estimation (RLSEstimate)**

**Input:**
1: \( h_i \) \hspace{1cm} \triangleright Entries of measurement matrix
2: \( y'_i \) \hspace{1cm} \triangleleft Corrupted Sensor Measurement

**Output:** \( W = \{w_i\} \)
3: Initialize: \( w_0 \leftarrow 0, P_0 \leftarrow \delta I \);
4: for Each time instant, \( k \) do
5: \( g \leftarrow h_k^T P_{(k-1)} \);
6: \( \gamma \leftarrow \lambda + gh_k \);
7: \( j_k \leftarrow g' / \gamma \);
8: \( e_k \leftarrow y'_k - w^T_{(k-1)} h_k \);
9: \( w_k \leftarrow w_{(k-1)} + j_k e_k \);
10: \( P' \leftarrow j_k g \);
11: \( P_k \leftarrow \frac{P_{(k-1)} - P'}{\lambda} \);
12: end for
13: return \( W \);

where, \( \lambda \) is the forgetting/weighting factor taking values between \((0, 1)\), \( \delta \) is a positive number (we consider it as 1), \( g \) is the gain vector, \( \gamma \) is the conversion factor, and \( P \) is the correlation
matrix. Here, $e_k$ is the error signal with $y'_k$ being the measurements from attacked sensor, $w_{(k-1)}$ being the predicted values from the RLS estimator, and $h_k$ is the measurement matrix entries at time $k$. In this algorithm, given $y'_i$ and $h_i$, we find estimated values, $w_i$, at each time point $k = 0, 1, 2, \ldots n$, that minimizes weighted sum of square error between predicted output and the sensor measurements. At the end of each iteration, the RLS algorithm updates the estimation error covariance matrix $P_k$. During the duration of attack, we compute the control input, $u_k$, with the estimated values, $w_i$. As such, we ensure that the system operates within the safety bounds given by us. Overall complexity of the RLS algorithm is $O(n^2)$ and it provides excellent performance while operating in real-time systems.

Prior to using the RLS algorithm, we detect the attack and perform pre-processing of data as shown in lines (6-10) of the Algorithm 2.

Algorithm 2 Algorithm for attack detection and measurement estimation

Input:
1: list zero ▷ Time points $t \in T_c$ of zero sensor outputs.
2: $y'_i$ ▷ Corrupted Sensor Measurement
3: $h_i$ ▷ Entries of Measurement Matrix

Output: attack detect, $list_{y'}$ ▷ attack detection time ($t_{ad}$), and list of estimated sensor outputs ($\hat{y}'_i$).

4: $list_{y'}$, $list_{\hat{y}'}$;
5: attack detect $\leftarrow$ False;
6: for Each $y'_i$ Input do
7: add $y'_i$ to $list_{y'}$;
8: if attack detect $==$ False then
9: if $y'_i \in list_{zero}$& Val($y'_i$) $\neq 0$ then
10: attack detect $\leftarrow$ True;
11: $list_{\hat{y}'} \leftarrow$ RLS Estimate($h_i, y'_i$);
12: end if
13: else
14: attack detect $\leftarrow$ False;
15: $list_{y'}, list_{\hat{y}'}$;
16: end if
17: end for
3.5 Case Study

For demonstration, we build a car-following model in which a follower vehicle is equipped with an adaptive-cruise control (ACC) system and it follows a leader vehicle on the same lane. The ACC system uses mm wave radar sensor to measure relative distance and velocity to a preceding vehicle (presumably an attacker vehicle). We use the radar sensor model of Section 4.1 for this purpose. For our experiments, we consider parameters of Bosch LRR2 long-range $(2 \leq d \leq 200 \text{ meter})$ mm-wave radar. To implement our challenge-response authentication (CRA) method in this radar, we modify its modulation unit. We assume that the sensor measuring velocity of the follower vehicle $(v_{F_v})$ is trusted. Our car-following model, modified CRA radar, and the detection as well as estimation methods are shown in Figure 4.3. Subsequently, simulation results of attacks on the follower vehicle and of our detection and estimation methods are shown in Figures 3.2 and 3.3.

3.5.1 Car-Following Model of Ground Vehicles

The ACC system of the follower vehicle operates in two modes (i) speed control and (ii) spacing control. In the absence of preceding vehicles, the ACC system operates the vehicle in the speed control mode, where it drives at a user-set speed $(v_{set})$. When a preceding vehicle is detected on the road by the radar, the ACC system of the follower vehicle decides on whether to continue driving in the speed control mode based on distance between the vehicles. If the distance is less than a desired value $(d_{des})$, given by Eqn. 4.13, the ACC system switches to spacing mode. In this mode, the desired distance, which is proportional to the headway time $(\tau_h = 3 \text{ sec})$ between the vehicles, minimum stopping distance $(d_0 = 5 \text{ m})$, and speed of the follower vehicle $(v_{F_{vk}})$, is maintained by controlling both throttle and brakes.

\[
d_{des(k)} = d_0 + \tau_h v_{F_{vk}}
\]  

(3.12)
We model longitudinal control of the ACC equipped follower vehicle as a hierarchical control architecture, consisting of an upper level controller and a lower level controller, shown in Figure 4.3 [57]. The upper level controller determines the desired longitudinal acceleration ($a_{des}$) according to distance ($d$), relative velocity $\Delta v$ between a leader vehicle ($L_v$) and a follower vehicle ($F_v$) (measured using a radar) and speed of the follower vehicle ($v_{F_v}$). The upper level linear output feedback controller is implemented based on a constant time headway (CTH) policy, which states that the desired speed of the follower vehicle ($v_{des}$) should be proportional to the inter-vehicular distance ($d$) and inversely proportional to the headway time ($\tau_h$). The controllers output dynamics based on CTH policy and transfer function of lower level controller is given by the following discrete time equation [58],

$$v_{des(k+1)} = \frac{1}{K_L} v_{F_v(k)} + \frac{T_L}{\tau_h \cdot K_L} \Delta v(k) + \frac{T_L}{k \cdot \tau_h \cdot K_L} \Delta d(k)$$  (3.13)

where, $K_L = 1.0$ is the system gain and $T_L = 1.008$ is the time constant for the follower vehicle [57]. The clearance error between the vehicles is $\Delta d(k) = d(k) - d_{des(k)}$, relative speed is $\Delta v(k) = v_{L_v(k)} - v_{F_v(k)}$, $k$ is discrete time in seconds. Here, $v_{L_v}$ is speed of the leader vehicle. From the value of $v_{des}$, we derive the output, $a_{des}$, of the upper level controller.

![Figure 3.1: Car-following model with hierarchical control architecture of ACC system & Detection, Estimation Method.](image)

The lower level controller of the ACC system determines the acceleration of pedal ($a_{pedal}$)
and brake pressure \((P_{brake})\) of the follower vehicle to ensure the desired acceleration \(a_{des}\) is tracked by actual acceleration \(a_{F_v}\). The closed loop transfer function of this controller with follower vehicle as the plant is given by the following first-order equation:

\[
a_{F_v} = \frac{K_L}{T_L s + 1} a_{des}
\]  

(3.14)

While designing the upper level controller, internal and external disturbances are neglected to ensure the lower level controller works correctly and satisfy dynamics of Eqn. 4.15. Similarly, nonlinearity at the lower level controller are compensated using inverse longitudinal dynamics. Now, our car-following model to simulate vehicular traffic flow (longitudinal) dynamics is built by enhancing the intelligent-driver model (IDM) with the hierarchical control model of ACC equipped follower vehicle, as shown in Figure 4.3. With the help of our model, we can describe acceleration and deceleration among vehicles in a satisfactory way. To find continuously changing velocity of the leader vehicle, we use the following equation

\[
v_{F_l(k+1)} = v_{F_l(k)} + a_{F_l(k+1)}
\]  

(3.15)

where, \(a_{F_l}\) and \(v_{F_l}\) are acceleration and velocity of the leader vehicle respectively. Similarly, we derive values of actual and desired acceleration \((a_{F_v,des})\) of the follower vehicle by using the following equation

\[
a_{F_v,des(k+1)} = v_{v,des(k+1)} - v_{v,des(k)}
\]  

(3.16)
To find positions of leader ($x_{F_l}$) and follower ($x_{F_v}$) vehicles, we use the following equation

$$x_{F_{l,v}(k+1)} = x_{F_{l,v}(k)} + v_{F_{l,v}(k+1)} + \frac{1}{2} a_{F_{l,v}(k+1)}$$

(3.17)

In our simulation, we use Eqn.(3.17) to measure distance, $d_{(k+1)} = x_{F_{l}(k+1)} - x_{F_{v}(k+1)}$, between the leader and the follower vehicles. In an actual scenario, values of $d$ and $\Delta v$ of the car-following model are calculated using the radar Eqn. 3.7 and Eqn. 3.8. From Figure 4.3, we can see that the internal states of the ACC equipped follower vehicle are $a_{des}$, $P_{brake}$ and $a_{pedal}$. Corrupted distance and relative velocity measurements of radar, effects calculation of state, $a_{des}$, which then influences output ($v_{F_v}$) of the system.

### 3.5.2 Simulation and Results

The goal of our simulation is to demonstrate the attacks on an ACC equipped follower vehicle and show effect of our detection and estimation methods on velocity of the vehicle. We consider two car-following scenarios: (i) the leader vehicle decelerates at a constant acceleration of -0.1082 $m/sec^2$ (Figure 3.2) and (ii) the leader vehicle decelerates and accelerates at -0.1082 $m/sec^2$ and +0.012 $m/sec^2$ respectively (Figure 3.3). The follower vehicle has to slow down accordingly to ensure the inter-vehicular distance is greater than the desired distance ($d_{des}$) to avoid rear end collision. We consider 65 miles/hr and $v_{set} = 67$ miles/hr as the initial velocities of the leader and the follower vehicles respectively. The leader starts slowing down when the initial distance between the vehicles is 100m. For such a scenario, an adversaries intention is to provide corrupted data ($d_{a}$, $\Delta v_{a}$ of Figure 4.3) to the ACC system’s radar sensor for causing a collision.

We simulate the attacks and the car-following scenario in MATLAB. For design, simulation, and analysis of the radar sensor, we use the Phased Array System Toolbox. The root MUSIC algorithm is used to extract beat frequencies from radar data. We derive values of distance and
relative velocity between vehicles from the measured beat frequencies.

- **Follower vehicle under DoS attack**

To carry out this attack, we consider a self-screening jammer whose $P_J = 100$ mW, $G_J = 10$ dbi, $B_J = 155$ Mhz, and $L_J = 0.10$ db. When the follower vehicle is attacked with such a jammer, measurements of the radar sensor becomes corrupted. In the Figure 3.2a and 3.3a, we show that the DoS attack begins at time $k = 182$ sec, after which the sensor receives very high value of corrupted distance and velocity measurements. Prior to it, the attacked signal follows the actual reflected signal. Due to incorporation of our CRA based detection method on the radar, it receives measurements whose values are zero at certain time instances such as at $k = 15, 50, 175$ sec of Figures 3.2a and 3.3a. This occurs because the radar do not transmit any data at those time points. As, at $k = 182$ sec, the attacked signal value is not zero, our detector could detect the attack and notify it to the system. Subsequently, after $k = 182$ sec till the end of attack, our estimation method provides distance and relative velocity data to the upper level controller of the ACC system. As such, our method prevents the vehicle from performing task with undesired consequences.

- **Follower vehicle under Delay injection attack**

For the delay injection attack, we consider a scenario where the adversary generates counterfeit signals to increase distance between vehicles. In our case, the distance measurements received by the radar after $k = 180$ sec are 6 meter more than the actual distance. Such corrupted values of distance effects the desired acceleration ($a_{des}$) measurements of the ACC controller. As a result, the follower vehicle does not slow down as desired, which can be seen in Figures 3.2b and 3.3b. On using our detection method in the same way as was in the DoS attack, we observe that the attack occurs at $k = 182$ sec. Due to the attack, the velocity of the follower increases and the distance reduces between the vehicles. To make the ACC equipped follower vehicle drive properly

---

1The spikes going to zero in Figure 2 are indication of radar sensor not producing any output at challenge times $k = 15, 50, 175$ etc. They do not imply relative velocity between the vehicles going to zero at those times. Any other value of spikes in the Figure 2 are indications of noise in radar data.
during the duration of the attack, we use our estimation method, which corrects the distance and relative velocity measurements after the attack is detected.

![Graph showing detection and estimation outputs](image1)

(a) DoS Attack on reflected signal of radar and Detection, Estimation Output

![Graph showing detection and estimation outputs](image2)

(b) Delay Attack on reflected signal of radar and Detection, Estimation Output

Figure 3.2: Plots of Attacks and Detection, Estimation Outputs with constant deceleration of leader vehicle

![Graph showing detection and estimation outputs](image3)

(a) DoS Attack on reflected signal of radar and Detection, Estimation Output

![Graph showing detection and estimation outputs](image4)

(b) Delay Attack on reflected signal of radar and Detection, Estimation Output

Figure 3.3: Plots of Attacks and Detection, Estimation Outputs with acceleration and deceleration of leader vehicle

• **Results**

With the help of the detection method, we were able to detect both the attacks at $k = 182$
Subsequently, we use recursive least square algorithm to predict distance and relative velocity values for the duration of the attack ($k = 182$ sec to $k = 300$ sec). The algorithm had run-times of $1.2e+7$ nanoseconds and $1.3e+7$ nanoseconds for both cases of jamming and delay injection attacks respectively. Our detection method did not produce any false positives or false negatives for both the attack scenarios.

3.6 Summary

We introduced a challenge-response authentication based method for detection of two types of attacks: the Denial of Service (DoS) and the delay injection, on sensors of autonomous systems. The recursive least square approach is used for estimation of sensor measurements when it is under attack. With these estimated measurements, safe control inputs of the autonomous CPS are derived, which enables the system to recover and operate safely in the presence of attacks. A case study was presented to show resiliency of adaptive cruise control system of ground vehicle, leveraging our proposed solutions to counter these attacks. However, the detection method fails when an adversary with adequate resources can sample the incoming signals from sensors faster than the defender. Our future research will address this limitation and we will provide defence mechanisms to prevent such adversaries from attacking sensors of autonomous systems. We will also extend our case study on autonomous ground vehicle to include a non-linear system model with lateral dynamics.
CHAPTER 4: PASSIVE DETECTION AND RESILIENT STATE ESTIMATION

4.1 Overview

In this chapter, we model the CPS as a stochastic linear system with zero mean, white Gaussian noise. Malicious Denial of Service (DoS) or False Data Injection (FDI) attacks corrupt the measurements of the sensors and actuators of the considered system. Consequently, we develop a passive attack resilient estimator in the Bayesian framework and combine it with the Chi-squared detector to address the problem of simultaneous attack detection and state estimation. We show the asymptotic convergence of estimation error to zero when there is no attack and we provide an upper bound on the error during attack. By bounding the estimation error, we were able to obtain approximate state estimates, when sensors of the system were compromised. The unique features of our method are:

- Our method can recursively estimate states with better performance than the standard Kalman filter and Robust Kalman filter of [29] against DoS and FDI attacks.

- Our method can approximately (within an error bound) reconstruct the states when the sensors were attacked.

- Our method applies explicit formulas, which make it computationally less expensive than the state-of-the-art recursive method [28]. The number of Gaussian mixture components in [28] increases exponentially with time and they apply pruning and merging procedures to obtain approximate solutions.
4.2 System Model without Attacks

We model the dynamics of the CPS as a linear time-invariant (LTI) system with process and measurement noise, which is described by the following equations:

\[
\begin{align*}
    x_{k+1} &= Ax_k + Bu_k + w_k \\
    y_k &=Cx_k + v_k
\end{align*}
\]

where, \(x_k \in \mathbb{R}^n\) is a real-valued system state vector at time \(k\), \(u_k \in \mathbb{R}^m\) is a real-valued control input vector, and \(y_k \in \mathbb{R}^q\) is a real-valued sensor measurement vector, \(w_k \sim \mathcal{N}(0, \Sigma_w)\) is the additive white Gaussian system noise with zero mean and covariance \(\Sigma_w\), and \(v_k \sim \mathcal{N}(0, \Sigma_v)\) is the additive white Gaussian measurement noise with zero mean and covariance \(\Sigma_v\) (in this work, \(\mathcal{N}(\mu, \Sigma)\) represents a Gaussian distribution with mean \(\mu\) and covariance \(\Sigma\)). Both the noises are assumed to be independent of each other. Here, \(A\) is the system matrix and \(B, C\) are the transformation matrices. We assume that the time-invariant matrices \(A, B, C\) are known.

Traditionally, an optimal Kalman filter (KF) is used for state estimation in LTI system with additive white Gaussian process and measurement noises. The KF algorithm has two stages: (i) prediction or estimation update and (ii) measurement update. The prediction stage is represented by the following equations:

\[
\begin{align*}
    \hat{x}_{k+1|k} &= A\hat{x}_{k|k} + Bu_k;
    P_{k+1|k} &= AP_kA^T + \Sigma_w
\end{align*}
\]

where, \(P_k\) is the estimation error covariance matrix.

The measurement update equations of the KF are:

\[
\begin{align*}
    z_{k+1} &= y_{k+1} - C\hat{x}_{k+1|k};
    K_{k+1} &= P_{k+1|k}C^T(CP_{k+1|k}C^T + \Sigma_v)^{-1};
    \hat{x}_{k+1} &= \hat{x}_{k+1|k} + K_{k+1}z_{k+1}, \text{ and }
    P_{k+1} &= (I - K_{k+1}C)P_{k+1|k}
\end{align*}
\]

where, \(z_k \in \mathbb{R}^q\) is the estimation residue (without attack) at time \(k\) with Gaussian distribution \(z_k \sim \mathcal{N}(0, CP_{k+1|k}C^T + \Sigma_v)\) and \(K_k\) is the time-varying Kalman gain matrix. When the
matrices \((A, B)\) and \((A, C)\) of the system are assumed to be stabilizable and detectable respectively, we get a steady-state KF, whose time-varying error covariance matrices \(P_{k|k-1}\) converges to \(P\) i.e. \(P = \lim_{k \to \infty} P_{k|k-1}\) and the time-varying Kalman gain converges to a constant value i.e. \(K = \lim_{k \to \infty} K_k\) and thus it reduces to \(K = PC^T(CPC^T + \Sigma_v)^{-1}\). The estimation error of the filter is given by \(e_k = (x_k - \hat{x}_k)\).

The \(\chi^2\) detector is generally used with KF in a control system to detect attacks. Based on the estimation residue of the steady-state KF, we can define a function, \(g_k = z_k^T(CPC^T + \Sigma_v)z_k\), whose value if greater than a user specified threshold \((\eta)\) i.e. \(g_k > \eta\), then an alarm is triggered by the \(\chi^2\) detector. Now, the probability of triggering an alarm at time \(k\) can be given as \(\beta_k \triangleq P(g_k > \eta)\).

4.3 Attack Model

We assume that the adversary has knowledge of the model of the plant and the filtering algorithm used during the feedback control i.e. static matrices of the system A, B, and C, filter gain K, and distribution of noises are known. Unlike existing work on state estimation against attack, we assume that an adversary can manipulate any number of sensors and actuators of the system. Now, an attack can be developed that can bypass the threshold of the \(\chi^2\) detector. However, we argue that such a stealthy attack will not have any impact on the stability of the system. Thus, in our model we only consider those attacks which can perturb the dynamics of the system. For simplicity of analysis, we consider attacks on sensors of the system and restrict our attention to the measurement/output equation: \(y_k = Cx_k + v_k\). However, our results can easily be extended to the case where there are attacks on actuator inputs.

We consider two types of attacks on the sensor signals: (i) Denial of Service (DoS) and (ii) False Data Injection (FDI) attacks. We assume that the attacks do not corrupt all the measurements.
after its initiation. The goal of an adversary is to make the system unstable by maximizing the state estimation error [59]. In case of DoS attack such as Jamming, impact on the system can be captured by the following attacked output signal, $y'_k$:

$$y'_k = y_{ad}^k$$ (4.3)

where, $y_{ad}^k$ is the DoS attack signal at certain time points. The goal of such an attack is to suppress the correct sensor measurements with a signal of more power.

The FDI attack requires an adversary to inject fake data in the sensor measurements with the goal of maximizing the impact on the system and not getting detected. The output equation considering the attack can be given as:

$$y'_k = y_k + y_{af}^k$$ (4.4)

where, $y_{af}^k$ is the fake injected data. In both these scenarios, the attack occurs after time $k > k_a$. Now, an adversary can carry out any one of these attacks, but not simultaneously.

### 4.4 Problem Description:

Given a stochastic linear time-invariant system whose sampled sensor measurements $y_k$ are under DoS or FDI attack, producing corrupted measurements $y'_k$ over a finite interval $[k_1, k_n]$, $k_1 \neq 0, k_n < \infty$, we want to use a detector that can find the presence of an attack and design a filter that can estimate system states with bounded estimation error during the duration of attack.

### 4.5 Resilient State Estimation and Analysis

In this section, we first explain our Bayesian inspired computationally efficient recursive algorithm, which is combined with the Chi-squared ($\chi^2$) detector to simultaneously detect and estimates states
that are resilient against DoS and FDI attacks. Then, we provide the condition for resilience of the state estimate and analyze the asymptotic performance of our estimator.

4.5.1 Resilient State Estimation Algorithm

The standard Kalman filter (KF) is robust to noise, but it has not been designed to mitigate the effect of adversarial attacks on state estimates. The $\chi^2$ detector has been used with KF to detect attack, but to the best of our knowledge, there has been few attempts on recursively estimating $x_k$ during an attack. Toward this objective, we propose the following computationally efficient algorithm, that combine the $\chi^2$ detector with our Bayesian inspired estimator.

First, let us briefly review the Bayesian interpretation of KF, which uses Bayes rule, $p(a|b)p(b) = p(b|a)p(a)$, to express the posterior probability in terms of the likelihood and the prior. Assuming that the distribution of $x_k$ follows the Gaussian distribution $\mathcal{N}(\hat{x}_k, P_k)$, we obtain a prior distribution that is $P(x_{k+1}|x_k) \sim \mathcal{N}(\hat{x}_{k+1}|k, P_{k+1}|k)$. By combining this prior information with $y_{k+1} \sim \mathcal{N}(Cx_{k+1}, \Sigma_v)$ and Bayes rule, we can shown that the posterior distribution is $x_{k+1} \sim \mathcal{N}(\hat{x}_{k+1}, P_{k+1})$, where $\hat{x}_{k+1}$ and $P_{k+1}$ are as defined in the KF.

As a result, given $x_k \sim \mathcal{N}(\hat{x}_k, P_k)$, $y_{k+1} = Cx_{k+1} + v_k$ should follow the distribution $\mathcal{N}(C\hat{x}_{k+1}|k, CP_{k+1}|kC^T + \Sigma_v)$. To detect the attack in $y_{k+1}$, a $\chi^2$ detector is applied to $z_{k+1}^T(CP_{k+1}|kC^T + \Sigma_v)^{-1}z_{k+1}$.

To apply the Bayes rule for resilient estimation, we again assume that for each $k$, $x_k \sim \mathcal{N}(\hat{x}_k, P_k)$. However, depending on whether there is an attack at $y_{k+1}$, the estimation of the posterior distribution of $x_{k+1}$ would be different.

We first derive the prior distribution of $x_{k+1}$. Let

$$P_{k+1|k} = AP_kA^T + \Sigma_w, \hat{x}_{k+1|k} = A\hat{x}_k + Bu_k,$$ (4.5)

then the prior information of $x_{k+1}$ is $x_{k+1} \sim \mathcal{N}(\hat{x}_{k+1|k}, P_{k+1|k})$. As a result, when $x_k$ is given, to
detect whether \( y_{k+1} \) is attacked, we should apply the \( \chi^2 \) detector to
\[
g_{k+1} = (y_{k+1} - C\hat{x}_{k+1|k})^T P_{k+1|k}^{-1} (y_{k+1} - C\hat{x}_{k+1|k}).
\]

When an attack is not detected at \( y_{k+1} \), then we combine the prior distribution of \( x_{k+1} \sim \mathcal{N}(\hat{x}_{k+1|k}, P_{k+1|k}) \) with the information \( y_{k+1} \sim \mathcal{N}(C x_{k+1}, \Sigma_v) \) to obtain the posterior distribution of \( x_{k+1} \). Applying Bayes rule, the posterior distribution of \( x_{k+1} \) is proportional to the product of these two probability density functions:
\[
x_{k+1} \sim \exp \left( -\frac{1}{2} \left[ (y_{k+1} - C x_{k+1})^T \Sigma_v^{-1} (y_{k+1} - C x_{k+1}) \\
+ (x_{k+1} - \hat{x}_{k+1|k})^T P_{k+1|k} (x_{k+1} - \hat{x}_{k+1|k}) \right] \right)
\]
By calculation, we obtain \( x_{k+1} \sim \mathcal{N}(\hat{x}_{k+1}, P_{k+1}) \) with
\[
P_{k+1} = \left( C^T \Sigma_v^{-1} C + P_{k+1|k}^{-1} \right)^{-1}.
\]

\( \hat{x}_{k+1} = P_{k+1} \left( C^T \Sigma_v^{-1} y_{k+1} + P_{k+1|k}^{-1} \hat{x}_{k+1|k} \right). \) (4.7)

When the detector suggests that there is an attack at \( y_{k+1} \), then we propose to drop the information of \( y_{k+1} \). As a result, the posterior distribution of \( x_{k+1} \) is the same as its prior distribution, that is, \( x_{k+1} \sim \mathcal{N}(\hat{x}_{k+1}, P_{k+1}) \), where \( \hat{x}_{k+1} = \hat{x}_{k+1|k}, \ P_{k+1} = P_{k+1|k} \).

Our procedure is summarized in Algorithm 3 and Figure 4.1 and its computational time complexity is \( O(\max(n, q)^3) \), same as of KF. The proposed method resembles an ”event-triggered” approach: when a detection alarm is triggered, update the Kalman filter in open-loop; otherwise, update the KF as usual. Our solution is simple, but effective against sensor attacks. We also use the Bayesian perspective to derive \( \hat{x}_k \) and \( P_k \) during filters open-loop operation.

Compared to an \( \ell_1 \) optimization based RKF [29], our method is simpler and more computationally efficient, as we do not solve an optimization problem. In addition, RKF fixes \( P_k \) to
be the $P$ of the steady-state KF, which is different from our setting where $P_k$ is derived from the Bayesian perspective and could be different for different $k$. As shown later in Section 4.6.2, our proposed algorithm outperforms RKF by producing smaller estimation error.

**Algorithm 3** Attack Detection & Resilient State Estimation

**Input:** Observation $\{y'_k\}_{k \geq 1} \in \mathbb{R}^q$; detection threshold $\eta$; model parameters $A, B, C, \Sigma_v, \Sigma_w, \{u_k\}_{k \geq 0}$.

**Output:** Estimated values $\hat{x}_k, k \geq 1$

**Initialize:** $\hat{x}_0 \in \mathbb{R}^n$ and $P_0 \in \mathbb{R}^{n \times n}$;

1: for $k = 0, 1, 2, \ldots$ do
2: Calculate $\hat{x}_{k+1|k}$ and $P_{k+1|k}$ using (4.5).
3: Apply $\chi^2$ detector to
4: $g_{k+1} = (y_{k+1} - C\hat{x}_{k+1|k})^T(CP_{k+1|k}CT + \Sigma_v)^{-1}(y_{k+1} - C\hat{x}_{k+1|k})$
5: if $g_{k+1} > \eta$ then $P_{k+1} = P_{k+1|k}$ and
6: $\hat{x}_{k+1} = \hat{x}_{k+1|k}$.
7: else Calculate $P_{k+1}$ and $\hat{x}_{k+1}$ using (4.6) and (4.7)
8: end if
9: Return: $\hat{x}_{k+1}$.
10: end for

---

Figure 4.1: Idea behind estimation in Algorithm 1: After $\chi^2$ detect the attack at $k = (6-9)$ and (13-15), sensor measurements at these time points are discarded and state estimation is done using the last good sensor value (at time points 5 and 12). For other values of $k$, Algorithm 1 considers the measurements for estimation.

### 4.5.2 Performance Analysis

In the analysis, we make the following assumptions:
\( (A, C) \) is detectable.

\[ \Sigma_v \text{ and } \Sigma_w \text{ are full rank matrices.} \]

We remark that the detectability of \( (A, C) \) is usually assumed for the analysis of Kalman filter, since it implies the convergence of \( P_k \) as \( k \to \infty \). For the proposed algorithm we let the error at the \( k \)-th time to be

\[ e_k = x_k - \hat{x}_k \]

and the error dynamic can be written as

\[
e_{k+1} = \begin{cases} 
   P_{k+1} \left( C^T (v_{k+1} + y^a_{k+1}) + P_{k+1|k}^{-1} (Ae_k - w_k) \right), & \text{if } g_{k+1} \leq \eta \\
   Ae_k - w_k, & \text{if } g_{k+1} > \eta,
\end{cases}
\]

(4.8)

where \( y^a_k \) is the attack vector. In addition, the condition \( g_{k+1} > \eta \) can be translated as a condition depending on \( e_k \):

\[
(v_{k+1} + y^a_{k+1} - C(Ae_k - w_k))^T (CP_{k+1|k}CT + \Sigma_w)^{-1} (v_{k+1} + y^a_{k+1} - C(Ae_k - w_k)) > \eta.
\]

(4.9)

To analyze the perform of the proposed algorithm, we investigate the following three scenarios:

- The noises \( w_k \) and \( v_k \) are zero and there is no attack (though \( \Sigma_w \) and \( \Sigma_v \) are still used as parameters in the algorithm).

- The noises \( w_k \) and \( v_k \) are sampled from \( \mathcal{N}(0, \Sigma_w) \) and \( \mathcal{N}(0, \Sigma_v) \) and there is no attack.

- The noises \( w_k \) and \( v_k \) are sampled from \( \mathcal{N}(0, \Sigma_w) \) and \( \mathcal{N}(0, \Sigma_v) \) and the attack exists.

For the first scenario, we have the following result:
**Theorem 4.5.1.** If the initialization satisfies that \( e_0^T P_0^{-1} e_0 < \eta \), and \( w_k = v_k = 0 \) for all \( k \geq 1 \), then \( g_k < \eta \) for all \( k \geq 1 \). In addition, as \( k \to \infty \), the estimation error converges to zero in the sense that \( e_k^T P_k^{-1} e_k \to 0 \).

This suggests that when there is no noise and no attacks, the “alarm trigger” of the proposed algorithm is not activated and the estimator converges to the correct state estimate. Since, the alarm trigger is not activated, \( P_k \) is the same as the error covariance of the Kalman filter, and thus it converges. As a result, the error \( ||e_k|| \) converges to zero.

However, one cannot expect that \( ||e_k|| \) converges to zero in the second scenario, due to the existence of noise. In addition, the noise could be so large that the alarm trigger is activated. However, in the following statement we show that if the initialization and the threshold \( \eta \) is well chosen, then with high probability the alarm trigger would not be activated and additionally the error is bounded:

**Theorem 4.5.2.** If \( \eta_0 < \eta/4 \) and \( \eta_i < \eta \) for \( i = 1, 2, \ldots, k-1 \), then with probability \( Pr(\chi^2 < \eta/4) \), \( \eta_k < \eta \) and \( e_k^T P_k^{-1} e_k < \eta \).

This theorem suggests that as long as the threshold \( \eta \) is reasonably chosen, then in this scenario there is a high probability that the alarm trigger is not activated.

To investigate resilience to attacks in the third scenario, we follow [60] and define it as “A CPS system is resilient, if the disturbance on the state estimate caused by an arbitrary attack is bounded”. In fact, we have the following property on resilience, which shows that an arbitrary attack can only cause bounded error:

**Theorem 4.5.3.** There exists a constant \( c \) such that \( ||\hat{x}_{k+1} - \hat{x}_{k+1|k}|| \leq c \) for all \( k \geq 0 \), where \( c \) is independent of the attack and it only depends on \( A, C, P_{k+1}, \Sigma_w \) and the threshold \( \eta \) used in the detection algorithm.

The theorem suggests that any attack only causes a bounded error, and in this sense it is
resilient to attacks. In comparison, an arbitrarily large attack could cause a arbitrarily large error for Kalman filter.

4.5.3 Proofs

Proof of Theorem 4.5.1. We first show that \( g_1 < \eta \) based on the assumption \( e_0^T P_0^{-1} e_0 < \eta \). Applying (4.8) with the assumptions on \( w_k \) and \( v_k \), \( g_1 \) reduces to

\[
g_1 = (C A e_0)^T (C (A P_0 A^T + \Sigma_w) C^T + \Sigma_v) -1 C A e_0
\]

\[
\leq (C A e_0)^T (C A P_0 A^T C^T) -1 C A e_0 = e_0^T P_0^{-1} e_0 < \eta.
\]

Since \( g_1 < \eta \), (4.8) implies \( e_{i} = P_2 P_2^{-1} A e_{i-1} \), and

\[
e_2^T P_2^{-1} e_2 = e_1^T A^T P_2^{-1} P_2 P_2^{-1} A e_1 \leq e_1^T A^T P_2^{-1} P_2 P_2^{-1} A e_1
\]

\[
= e_1^T A^T P_2^{-1} A e_1 \leq e_1^T A^T (A P_2 A^T)^{-1} A e_1 = e_1^T P_1^{-1} e_1.
\]

Using induction, it can be shown that \( g_k < \eta \) for all \( k \geq 1 \) and

\[
e_1^T P_1^{-1} e_1 \geq e_2^T P_2^{-1} e_2 \geq e_3^T P_3^{-1} e_3 \geq \cdots
\]

By checking the condition for equality in this decreasing sequence, one can show that \( \lim_{k \to \infty} e_k^T P_k^{-1} e_k = 0 \).

Proof of Theorem 4.5.2. It can be verified from the Bayesian perspective that if \( e_0 \sim N(0, P_0) \), then \( e_i \sim N(0, P_i) \) for any \( i \geq 1 \).

Assume that \( e_k = L_k e_0 + e'_k \), where \( L_k \) is a linear operator and \( e'_k \) only depends on the noise \( v_k, w_k \) and is independent of \( e_0 \). Since \( e_k \sim N(0, P_k) \) if \( e_0 \sim N(0, P_0) \), so we have \( e'_k \sim \)
\[ N(0, P_k - L_k^T P_0 L_k). \] As a result, we have

\[
e_k^T P_k^{-1} e_k \leq 2(L_k e_0)^T P_k^{-1} (L_k e_0) + 2 e_k^T P_k^{-1} e_k',
\]

\[
\leq 2(L_k e_0)^T (L_k^T P_k L_k)^{-1} (L_k e_0) + 2 e_k^T (P_k - L_k^T P_k L_k)^{-1} e_k'.
\]

Since \( \eta_0 = (L_k e_0)^T (L_k^T P_k L_k)^{-1} (L_k e_0) < \eta/4 \) and \( e_k^T (P_k - L_k^T P_k L_k)^{-1} e_k' \sim \chi^2 \), so the theorem is proved. \( \square \)

**Proof of Theorem 4.5.3.** When an attack is detected, then \( \hat{x}_{k+1} - \hat{x}_{k+1|k} = 0 \). Thus, it is sufficient to prove the setting when the \( \chi^2 \) detector does not detect an attack, i.e., \( g_{k+1} < \eta \). Then, we have

\[
\hat{x}_{k+1} - \hat{x}_{k+1|k} = (C^T \Sigma^{-1}_v C + P_{k+1|k})^{-1} \left( C^T \Sigma^{-1}_v y_{k+1} + P_{k+1|k} \hat{x}_{k+1|k} \right) - (C^T \Sigma^{-1}_v C + P_{k+1|k})^{-1} \left( C^T \Sigma^{-1}_v C \hat{x}_{k+1|k} + P_{k+1|k} \hat{x}_{k+1|k} \right) = (C^T \Sigma^{-1}_v C + P_{k+1|k})^{-1} C^T \Sigma^{-1}_v (y_{k+1} - C \hat{x}_{k+1|k}).
\]

Combining it with the result from Lemma 4.5.4 that \( \|y_{k+1} - C \hat{x}_{k+1|k}\| \leq c_1 \), we prove the theorem. \( \square \)

**Lemma 4.5.4.** There exist constants \( c_1, c_2 > 0 \) depending on \( A, C, P_{k+1}, \Sigma_w \) such that if \( \|y_{k+1} - C \hat{x}_{k+1|k}\| \geq c_1 \) then the \( \chi^2 \) detector will detect an attack, i.e., \( g_{k+1} > \eta \) and when \( \|y_{k+1} - C \hat{x}_{k+1|k}\| \leq c_2 \), then the \( \chi^2 \) detector will not detect an attack, i.e., \( g_{k+1} \leq \eta \).

**Proof.** Note that it is sufficient to prove that if \( \|y_{k+1} - C \hat{x}_{k+1|k}\| \geq c_1 \), then

\[
(y_{k+1} - C \hat{x}_{k+1|k})^T P_{k+1}^{-1} (y_{k+1} - C \hat{x}_{k+1|k}) \geq \eta.
\]

(4.10)

Setting \( c_1 = \sqrt{\lambda_{\max}(P_{k+1})} \eta \), where \( \lambda_{\max}(P_{k+1}) \) represents the largest eigenvalue of \( P_{k+1} \). Then,
it can be verified that (4.10) holds as long as \( \|y_{k+1} - C\hat{x}_{k+1|k}\| \geq c_1 \).

Similarly, lemma is proved when \( c_2 = \sqrt{\lambda_{\min}(P_{k+1})} \), where \( \lambda_{\min}(P_{k+1}) \) is the smallest eigenvalue of \( P_{k+1} \).

### 4.6 Case Study

For demonstration, we consider a car-following system (shown in Figure 4.2) in which a follower vehicle \( f \) is equipped with an adaptive-cruise control (ACC) unit and it follows a leader vehicle \( l \) on the same lane. The ACC system uses mm wave radar (external) such as Bosch LRR2 long-range (\( 2 \leq d \leq 200 \) meter) and internal sensors to measure position and velocity of the preceding and follower vehicle. We consider attacks that corrupt position \( (x^{(f)}_{\text{attack}}) \) and velocity \( (v^{(f)}_{\text{attack}}) \) measurements of the follower vehicle, as shown in Figure 4.3. As such, the goal of our Algorithm 1 is to minimize the effect of corrupted sensor data on the inputs (relative distance and relative velocity) of the ACC controller. Our method can be also extended to multiple vehicle platoon. In such a case, our Algorithm 1 will operate on each vehicle by considering details of the state-space model of the platoon and network topology. In the following sub-sections, we elaborate on the car-following setup and discuss our simulation results.

![Figure 4.2: Car-following with ACC equipped follower whose radar measures position \( (x^{(l)}) \) and velocity \( (v^{(l)}) \) of the leader](image-url)
4.6.1 Leader and Follower Vehicle Models

We use kinematic equations to describe leader vehicle \((l)\) dynamics. Changing velocity of the leader is generated using,

\[
v_{k+1}^{(l)} = v_k^{(l)} + a^{(l)} \Delta t
\]

(4.11)

where, \(k \in \{0, 1, 2, \ldots\}\) is the number of time iteration and \(a^{(l)}, v^{(l)}\) are constant acceleration and velocity respectively. Here, \(\Delta t = 0.01\) is the size of time increment. The position of the leader \((x^{(l)})\) at any time can be determined using the equation,

\[
x_{k+1}^{(l)} = x_k^{(l)} + v_k^{(l)} \Delta t + \frac{1}{2} a^{(l)} (\Delta t)^2
\]

(4.12)

These velocity and position measurements of the leader are captured by the radar of the follower vehicle.

The ACC system (shown in Figure 4.3) drives the follower vehicle \((f)\) at a user-set speed \((v_{set})\) in the absence of a preceding/leader vehicle. When a vehicle is detected, the ACC unit uses the constant time-gap (CTH) spacing policy (4.13) for maintaining desired inter-vehicular distance \((d^{(l,f)})\) to its preceding vehicle,

\[
d^{(l,f)} = d_r + hv^{(f)}
\]

(4.13)

where, \(h\) is the headway time \((h = 3 \text{ sec})\) between the vehicles, \(d_r\) is the minimum stopping distance \((d_r = 5 \text{ m})\), and speed of the follower vehicle is \(v^{(f)}\). According to [61], such a spacing-policy improves traffic throughput and safety.

Autonomous controller designed based on the CTH policy has a hierarchical architecture [62]. The upper level controller of the architecture determines the desired longitudinal acceleration
\((a^{(fd)})\) according to speed of the follower vehicle \((v^{(f)})\), relative velocity \((\dot{u})\), and relative distance \((u)\) between the leader and the follower vehicles. As such, the control law is given by the following equation,

\[
a^{(fd)} = -\frac{1}{h}(\dot{u} + \gamma u + \gamma hv^{(f)})
\]

\(u = x^{(l)} - x^{(f)}\)

where, \(\gamma = 0.9\) is a system parameter and \(x^{(f)}\) is position of the follower at any time.

The lower level controller of the architecture determines the acceleration of pedal \((a_{\text{pedal}})\) and brake pressure \((P_{\text{brake}})\) of the vehicle. Due to the presence of actuator dynamics and the lower controller, acceleration \((a^{(f)})\) obtained is not same as the desired value \((a^{(fd)})\). This is shown by the following equation,

\[
a^{(f)} + \tau \dot{a}^{(f)} = a^{(fd)}
\]

where, \(\tau = 1.008\) is the time constant, and \(\dot{a}^{(f)}\) is the jerk. While designing the upper level controller, internal and external disturbances are neglected to ensure the lower level controller works correctly and satisfy dynamics of \((4.15)\). Similarly, non-linearity at the lower level controller are compensated using inverse longitudinal dynamics.

Based on \((4.14)\) and \((4.15)\), the ACC vehicle dynamics can be represented in the following discrete time, multi-input multi-output state-space form,

\[
X^{(f)}_{k+1} = H_k X^{(f)}_k + Gu_k + w^{(f)}_k
\]

\[
Y^{(f)}_k = C X^{(f)}_k + v^{(f)}_k
\]

where, \(X^{(f)} = [x^{(f)}, v^{(f)}, a^{(f)}] \in \mathbb{R}^{3 \times 1}\) is the state vector, relative distance, \(u\), is the input, \(Y^{(f)} = \)
\([x(f), v(f)] \in \mathbb{R}^{2 \times 1}\) is the output vector, and the matrices are \(H_k = I + \Delta t A_k, G = \Delta t B\), and

\[
C = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix}
\]

Notice, that \(A_k \in \mathbb{R}^{3 \times 3}\) is a time-varying matrix that changes according to acceleration \((A_a)\) and deceleration \((A_d)\) of the vehicle, where

\[
A_a = \begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & -(1 + \frac{1}{\tau}) & \frac{1 + \frac{1}{\tau}}{\tau}
\end{bmatrix},
A_d = \begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & -(1 + \frac{1}{\tau}) & \frac{1 + \frac{1}{\tau}}{\tau}
\end{bmatrix}.
\]

The input matrix, \(B = [(1 + \frac{1}{\tau}) - \gamma/\tau h, -\frac{1}{\tau h}, 0] \in \mathbb{R}^{3 \times 1}\) and time-varying Gaussian process and measurement noises are given by \(w_k^{(f)}\) and \(v_k^{(f)}\) respectively. During attacks, the output (4.17) changes according to the equations (6.18 and 6.17) and it produces malicious measurements \((x_{\text{attack}}^{(f)}, v_{\text{attack}}^{(f)})\).

To mitigate the effect of attacks, we use our Algorithm 3, which produces estimated values of position \((\hat{x}^{(f)})\) and velocity \((\hat{v}^{(f)})\) (shown in Figure 4.3). With these estimates, we obtain corrected values of inputs: relative distance \((u)\) and relative velocity \((\hat{u})\), of the ACC controller.

---

**Figure 4.3: Attack on sensor measurements of ACC system**
4.6.2 Simulation and Results

The car-following scenario consisting of leader and follower vehicle models, two attacks: (i) False Data Injection (FDI) and (ii) Denial of Service (DoS), and our resilient state estimation and detection algorithm are simulated in MATLAB. In the car-following scenario, we consider the leader vehicle decelerates and accelerates at \(-0.1082 \text{ m/sec}^2\) and \(+0.012 \text{ m/sec}^2\) respectively. The follower vehicle has to slow down accordingly to ensure the inter-vehicular distance is greater than the desired distance \(d(l,f)\) to avoid rear end collision. We consider 65 miles/hr and 60 miles/hr as the initial velocities of the leader and the follower vehicles respectively. The leader starts slowing down when the distance between the vehicles is 10 m. For such a scenario, an adversaries intention is to corrupt measurements of the internal sensors of the follower vehicle so that it leads to undesired consequences.

**Setup for Algorithm 1**

The threshold of the \(\chi^2\) detector of the algorithm is fixed for both the attacks at \(\eta = c_d\), where we choose the constant \(c_d = 20\). The initial state for the leader vehicle is set to \(X_0^{(l)} = [12, 29.05, -0.108]^T\). For the follower vehicle it is \(X_0^{(f)} = [2, 26.82, 0.112]^T\) and covariance of the process and measurement noises are assumed to be \(\Sigma_w = \text{diag}(1, 1, 1)\) and \(\Sigma_v = \text{diag}(1, 1)\) respectively. The estimation results of our method is compared against the standard Kalman filter and the Robust Kalman filter [29]. The estimator [29] requires a parameter \(\lambda\) and we pick the one that gives optimal performance.

**Case 1: Attack free scenario**

We first evaluate our algorithm against the standard Kalman filter and Robust Kalman filter in the attack free case. For our experiment, we consider a time frame of \((0 - 1.5) \text{ sec}\) with step size of 0.01 sec and \(k = \{1, 2, \ldots, 150\}\) iterations. Figures 4.4a and 4.4b compares true and corrected values of relative distance and relative velocity respectively. We observe that the three estimators minimizes the effect of noise in both the measurements. To highlight the performance
of our filter, we calculate the estimation error using \( \sqrt{\frac{1}{150} \sum_{i=1}^{150} \| \mathbf{X}_{\text{true}} - \hat{\mathbf{X}}_{\text{estimate}} \|^2} \) and found that the error produced by our algorithm for followers’ position (3.604) and followers’ velocity (3.207) measurements are less than the Robust Kalman filter: 3.673 (followers’ position) and 3.318 (followers’ velocity).

![Comparison of true and estimated value of relative distance](image1)

![Comparison of true and estimated value of relative velocity](image2)

Figure 4.4: Plots of relative distance (a) and relative velocity (b) between leader and follower vehicles for the attack free case. Estimation of Our Filter is as good as the Kalman Filter (KF) and the Robust Kalman Filter (RKF)
Case 2: False Data Injection attack

We consider a scenario where an adversary corrupt measurements of internal sensors of the follower vehicle after a certain time point. In case of FDI attack on our experimental system, malicious data of random value are added to the position ($x^{(f)}$) and velocity ($v^{(f)}$) outputs. We assume that the signal attack vector $y_{k}^{af}$ is injected into the output data at random time points after $k = 5$, but the attack does not corrupt all the measurements after its initiation. For instance, in our experiment, the attack occurs at time points such as 0.05, 0.11, 0.23 – 0.25, 0.44, 0.5 – 0.52, 1.18 – 1.2 and 1.46. The $\chi^2$ detector of our Algorithm 1 promptly detect the attacks by comparing the value of the function $g_k$ against the threshold $\eta$. Subsequently, the estimated values of position ($\hat{x}^{(f)}$) and velocity ($\hat{v}^{(f)}$) generated by our algorithm is used for calculating relative position and relative velocity of the ACC controller. Figures 4.5a and 4.5b provides comparison between the true and corrected values of relative distance and relative velocity. Note that the effect of follower vehicles position and velocity estimates on the ACC controller inputs, $(u, \dot{u})$, generated by our method at attack time points outperform the results of the traditional Kalman filter and the Robust Kalman filter. Most significantly, unlike the other two methods, the position estimates from our algorithm ensures that the relative distance values are always positive, which indicate that the leader and the follower vehicles never collide at any time. At all other time points (when there is no attack), our method performs as well as the optimal Kalman Filter. We also calculated the estimation error during FDI attack and found that the error produced by our algorithm for followers’ position (5.75 m) and followers’ velocity (4.64) m/sec measurements were again less than the Robust Kalman filter (RKF): 10.97 m (followers’ position) and 9.40 m/sec (followers’ velocity). Consequently, since the minimum stopping distance, $d_r = 5 m$, the likelihood of our algorithm preventing a collision is higher than the RKF.
Figure 4.5: False Data Injection attack: (a,b) comparison of true and corrected values of relative distance and relative velocity by Kalman Filter (KF), Robust Kalman Filter (Robust KF) and Our Filter.

- **Case 3: Denial of Service (DoS) attack**

  In the case of DoS attack, we assume that a self-screening jammer is used to jam the sensor signals after \( k = 10 \). Similar to FDI attack, we consider a time frame of (0-1.5) and the DoS attack occurs at time points such as \( 0.1, 0.28 - 0.3, 0.55 - 0.57, 0.97 - 1.0, 1.23, 1.24 \) and 1.48. Again, the \( \chi^2 \) detector is able to detect the attack. Subsequently, we compare the performance of the traditional Kalman filter, the Robust Kalman filter, and our filter. From Figure 4.6a and 4.6b, we observe that the estimates produced by our algorithm for follower vehicles position and velocity during attack time points, make the the ACC controller inputs, \((u, \dot{u})\), much closer to the actual values. Again, the position estimates from our method ensures that the relative distance \( (u) \)
is always positive during attack, thereby preventing possible accident. Our method also performs as well as the optimal Kalman filter at other time points. We also calculated the estimation error during DoS attack and found that our method still performed better by producing less error 4.54 m (followers’ position) and 4.94 m/sec (followers’ velocity) than the Robust Kalman filter 10.27 m (followers’ position) and 12.08 m/sec (followers’ velocity). Again, the likelihood of our algorithm in preventing a collision is higher than the RKF as the followers’ position error is less than the minimum stopping distance, $d_r = 5\, m$.

![Comparison of true and estimated value of relative distance](image)

![Comparison of true and estimated value of relative velocity](image)

Figure 4.6: DoS attack: (a,b) comparison of true and corrected values of relative distance and relative velocity by Kalman Filter (KF), Robust Kalman Filter (Robust KF) and Our Filter.
4.7 Discussions

We first comment on the choice of $\eta$ of the $\chi^2$ detector. While, Theorem 4.5.3 guarantees that the estimation error is always bounded, we remark that the bound $c_3$ depends on $\eta$, that is, smaller $\eta$ leads to a smaller $c_3$ and an algorithm more resilient to attacks. However, if $\eta$ is very small, then $\Pr(\xi^2 < \eta/4)$ would be small and Theorem 4.5.2 implies that there will be many false positive, resulting in sensor measurements $y_i$ being dropped during estimation and causing our estimator to not perform well. Thus, it is important to choose the threshold $\eta$ carefully.

The idea behind the proposed algorithm can be generalized to various settings. For example, there has been studies on sparse attacks, where only a few sensors are attacked at each time $k$. The sparse attack requires an adversary to inject fake data in few sensor measurements and the output equation for such an attack can be given by:

$$y'_k = y_k + \Gamma_k y^{af}_k$$

where, $y^{af}_k$ is the injected malicious data and $\Gamma_k \in \mathbb{B}^{q \times q}$ is a Binary diagonal matrix whose $i$th diagonal entry when $[\Gamma_k]_{ii} = 1$, indicates the attack on sensors $i \in \{1 \ldots q\}$ (where, $q$ is the total number of sensors) and $[\Gamma_k]_{ii} = 0$, shows its absence.. The attack scenario can be analyzed with the given Bernoulli model: $\Pr([\Gamma_k]_{ii} = 1) = 0, \forall i = 1, 2, \ldots, q, \ k < k_f$ and $\Pr([\Gamma_k]_{ii} = 1) = p_{af}, \forall i = 1, 2, \ldots, q, \ k \geq k_f$, where $p_{af}$ is the probability of successfully injecting data after time $k_f$. For this setting, we may first use a detector to find out the sensors that has been attacked, and use the rest of the “correct” sensors and Bayes rules to obtain a good estimation of the states.

In general, our proposed algorithm performs well against DoS and FDI attacks, which was corroborated by the experimental results. However, it is not designed for attack such as replay whose magnitude is small and can escape detection by the $\chi^2$ detector. For such an attack, alternate approach such as [63] has been developed and it is a future direction to analyze and generalize the
algorithm in this work for stealthy attacks.

4.8 Summary

We have proposed a novel attack resilient filter that can recursively estimate states within an error bound, when sensors of the system are compromised. Our approach leverages Bayesian interpretation of the Kalman filter and combines it with the $\chi^2$ detector to ensure safety of CPS against Denial of Service and False Data Injection attacks. The computational complexity of our method is $O(\max(n, q)^3)$, which is same as that of the Kalman filter and it performs better than the standard and the Robust Kalman filters during attack as was shown in the car-following case study. In future, we plan to improve our current analysis on estimation errors to sparse attack and also intend to develop new detection and estimation procedures for covert and zero-dynamic attacks.
CHAPTER 5: DISTRIBUTED RESILIENT STATE ESTIMATION

5.1 Overview

In this chapter, we model the distributed CPS as a linear time-invariant system. Malicious attack corrupts the sensor measurements of some agents of the system. Consequently, we develop a \textit{passive} attack resilient distributed filter and provide theoretical guarantees. We show the asymptotic convergence of estimation error to zero when there is no attack and we provide an upper bound on the error during attack. The unique features of our method are:

- Our method can recursively estimate states of a distributed CPS without considering the number of agents that were compromised in the system. This particular characteristic separates our method from the rest and is effective in scenarios where adversaries might exist in greater numbers.

- Our method can \textit{approximately} (within an error bound) reconstruct the system states and its performance is not related to the magnitude of the input that can be injected by an adversary in sensor data.

5.2 Preliminaries

5.2.1 Notations

We assume that there are \( n \) agents, \( X \triangleq \{1, 2, \ldots, n\} \), in the distributed CPS, whose communication with each other can be described by an undirected graph \( G = (\mathcal{V}, \mathcal{E}) \). In the graph, nodes are the number of agents, \( \mathcal{V} = X \), and edges, \( \mathcal{E} = \mathcal{V} \cup \mathcal{V} \), represent communication between them. Here, \((i, j) \in \mathcal{E}\) is a bi-directional edge between \( i \) and \( j \), that enable them to send and receive messages among themselves, but not simultaneously. We assume that every agent in \( X \) has a self loop i.e. \((i, i) \in \mathcal{E}\) for all \( i = 1, 2, \ldots n \). Neighborhood of \( i \) is defined as the set of nodes that
are adjacent to it i.e. $N(i) = \{i\} \cup \{j \in V : (i,j) \in E\}$ and with whom it can communicate. Furthermore, we assume that each agent has an observer for estimating the state of the system. In this chapter, we use the words agent and node interchangeably.

### 5.2.2 System and Measurement Models without Attack

We model the dynamics of the distributed CPS as a linear time-invariant (LTI) system, which is described by the following equation:

$$x_{k+1} = Ax_k$$  \hspace{1cm} (5.1)

where, $x_k \triangleq [x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(n)}] \in \mathbb{R}^n$ is the state vector at time $k \in \mathbb{N}$ of the distributed system and $A \triangleq [A_{i,j}] \in \mathbb{R}^{n \times n}$ is the system matrix, with $A_{i,j}$ representing block matrix of $i$ and its neighbors $j$.

In the distributed system, each agent measures the system state at time $k$, which is given by

$$y_k^{(i)} = C^{(i)}x_k$$

where, $y_k^{(i)} \in \mathbb{R}^q$ is the measurement from $q$ sensors of the agent and $C^{(i)} \in \mathbb{R}^{q \times n}$ is the observation matrix. For analytical convenience, we represent the aggregated measurement vectors and observation matrices as

$$y_k = Cx_k$$  \hspace{1cm} (5.2)

where, $y_k \triangleq [y_k^{(1)}, y_k^{(2)}, \ldots, y_k^{(n)}]$ and $C \triangleq [C^{(1)}, C^{(2)}, \ldots, C^{(n)}]$. In our model, we assume that each agent estimates the system state, $x_k$, at each time-step $k$ based on the measurements gathered from its neighbors and its own. Also, an agent (good or malicious) is assumed to transmit the same information to all its neighbors. This assumption appears in many practical scenarios such as in
vehicular ad-hoc network.

5.2.3 Measurement Model with Attack

We consider an insider attack, where an adversary has complete control over a set of nodes $\mathcal{V}_a \subset \mathcal{V}$ of the communication network. Such an attacker has knowledge of the observation matrices, $C^{(j)}$, of its neighbors, system matrix $A$, and the communication topology. With these information, they can influence the state of the system without affecting the message scheduler of the network. The reason for considering such an strong adversary model is to show that our resilient estimator can withstand the worst-case scenario.

The attack is carried out by manipulating the sensor data of the compromised agents and can be represented by the following equation:

$$y^{(i),a}_k = C^{(i)} x_{k-1} + a^{(i)}_k$$  \hspace{1cm} (5.3)

where, $a^{(i)}_k$ is the attack vector and $y^{(i),a}_k$ is the corrupted output of agent $i$. Such malicious measurements effect the state estimate of the agent which, when used by its neighbors affect their estimate as well. Consequently, the attack influences the state estimate of the distributed system.

We provide the following definition of a compromised agent

**Definition 5.2.1.** Compromised Agent: An agent $i$ is compromised at time $k \in \mathbb{N}$ if its attack vector $a^{(i)}_k \neq 0$.

As the agents are completely controlled by an adversary, we do not make any assumption on the number of sensors that were manipulated. Also, unlike the $f$-adversarial attack model of [41], where they consider an upper bound on the number of adversarial neighbors of an agent, we do not make any such consideration. However, we acknowledge that the performance of our estimator degrades within an error bound as the number of compromised neighbors of an agent increases.
5.2.4 Problem Definition

Given a linear time-invariant distributed system of $n$ agents with a linear measurement model and an undirected communication graph $G$, design a filter that can estimate system states such that

$$\lim_{k \to \infty} \| \hat{x}^{(i)}_k - x_k \| \to 0, \ \forall i \in \mathbb{R}^n$$

when there is no attack and the estimation errors are bounded when sensor measurement of a subset of nodes $V_a \subset V$ are compromised by an insider attack.

To build such an estimator, we make the following assumptions:

- Matrix $(A, C)$ is detectable of the system. This assumption is in line with the assumption made in $[35, 41]$, where they state it as necessary condition for solving the distributed estimation problem with asymptotic guarantees.
- Each agent shares their estimated state information with neighbors via a secure communication channel. Thus, we do not consider any attack on the network.
- We assume that the agents cannot detect the attack inputs in the sensor measurements and thus accepts the corrupted state estimates from its neighbors.

5.3 Secure Distributed Estimation Method

In this section, we first propose an estimator (6.25) in Section 6.5.1 for no attack scenario and its convergence is proved in Theorem 6.5.2. Based on its optimization interpretation, we propose a novel estimator based on $\ell_1$ norm optimization in (5.8) and analyze its properties in Section 6.5.2. The proofs are provided in Section 6.5.3.

5.3.1 Distributed estimation without attack

In this section, we investigate the attack free case for the following model,

$$x_{k+1} = Ax_k, \quad y^{(i)}_k = C^{(i)} x_k$$  \hfill (5.4)
We assume that the estimation error covariance matrix, $P^{(i)}$, is chosen according to the following equation,

$$
P^{(i)} = \left( \frac{1}{d_i} \sum_{j \in N^{(i)}} (A P^{(i)} A^T + \Sigma^{(i)}_w)^{-1} \right)
+ C^{(i)} (\Sigma^{(i)}_v)^{-1} C^{(i)}$$  \hspace{1cm} (5.5)

where, $N^{(i)} = \{i\} \cup \{\text{neighbors of } i \text{ in } G\}$ and $d_i = |N^{(i)}|$ is the total number of neighbors of node $i$. Now, our proposed estimator has the following prediction rules,

$$
P^{(i)}_k = A P^{(i)}_k A^T + \Sigma^{(i)}_w$$ \hspace{1cm} (5.6)

$$
\hat{x}^{(i)}_k = P^{(i)}_k \left( \frac{1}{d_i} \sum_{j \in N^{(i)}} P^{(j)}_k (A \hat{x}^{(j)}_{k-1})^{-1} \right)
+ C^{(i)} (\Sigma^{(i)}_v)^{-1} y^{(i)}_k$$  \hspace{1cm} (5.7)

where, $\hat{x}^{(i)}_k$ is the state estimate and $P^{(i)}_k$ is a priori estimate covariance of agent $i$. This estimator is motivated from the Distributed Kalman Filter, which has been studied in many works such as [34, 36–38, 64]. While $\Sigma^{(i)}_v$ and $\Sigma^{(i)}_w$ are commonly used to denote the covariance of the noise in the system, here they have no physical meaning and are treated as parameters for developing the algorithm. In principle, they can be chosen to be any positive definite matrices and in our simulations we let them to be the identity matrices $I$.

To apply this estimator, we need to ensure that a solution to (6.23) exist. Thus, we give the following theoretical guarantee:

**Theorem 5.3.1.** If the graph $G$ is connected, $A$ is full-rank, $(A, C)$ is observable, and $\Sigma^{(i)}_v$ is full rank for all $1 \leq i \leq n$, then there exist $\{P^{(i)}_k\}_{i=1}^n$ that satisfy (6.23).

**Proof.** In the proof, both $A \succ B$ and $B \preceq A$ mean that $A - B$ is positive semidefinite.

59
Here, we let \( P_{0}^{(i)} = 0 \) for all \( 1 \leq i \leq n \) and show that for the sequence \( P_{k}^{(i)} \) generated by

\[
P_{k|k-1}^{(i)} = AP_{k-1}^{(i)}A^T + \Sigma_{w}^{(i)}
\]

\[
P_{k}^{(i)} = \left( \frac{1}{d_i} \sum_{j \in N^{(i)}} P_{k|k-1}^{(i)} - 1 + C_{i}^{T} \Sigma_{u}^{(i)}^{-1} C_{i} \right)^{-1}
\]

the limit, \( \lim_{k \to \infty} P_{k}^{(i)} \), exist and it is a positive definite matrix for all \( 1 \leq i \leq n \). If this is true, then \( P_{k}^{(i)} = \lim_{k \to \infty} P_{k}^{(i)} \) is a solution to (6.23).

We will first show that \( P_{1}^{(i)}^{-1} \) is bounded below by a positive definite matrix. For \( k = 1 \), we have

\[
P_{1}^{(i)}^{-1} \succeq C_{i}^{T} \Sigma_{u}^{(i)}^{-1} C_{i},
\]

which is positive semidefinite with range being the row space of \( C^{(i)} \), i.e., \( \{ C^{(i)} T z : z \in \mathbb{R}^q \} \).

If \( j \in N^{(i)} \) (and by definition \( i \in N^{(j)} \)), then

\[
P_{2}^{(i)}^{-1} \succeq \frac{1}{d_i} \left( AP_{1}^{(j)} A^T + \Sigma_{w}^{(j)} \right)^{-1} + C_{i}^{T} \Sigma_{u}^{(i)}^{-1} C_{i}
\]

\[
+ \frac{1}{d_i} \left( AP_{1}^{(i)} A^T + \Sigma_{w}^{(i)} \right)^{-1},
\]

which is a positive semidefinite matrix of range \( \{ C^{(i)} T z_1 + AC^{(i)} T z_2 + AC^{(i)} T z_3 : z_1, z_2, z_3 \in \mathbb{R}^q \} \). By applying the same procedure to time \( k = 3, 4, \ldots \), we verify that for sufficiently large \( k \), the range of \( P_{k}^{(i)}^{-1} \) can be given by the linear combination of \( \bigoplus_j \{ A^{l(j)} C_{j}^{T} z \} \) for some positive integer \( r \). When \( (A, C) \) is observable, this range is \( \mathbb{R}^n \) and as a result, \( P_{k}^{(i)}^{-1} \) is larger than a positive definite matrix with full rank. This suggests that \( P_{k}^{(i)} \) is bounded by a positive definite matrix from above.
In addition, by induction it can be shown that \( P_k^{(i)} \) is strictly increasing in the sense that

\[
P_0^{(i)} \ll P_1^{(i)} \ll P_2^{(i)} \ll \cdots .
\]

Since, the sequence is bounded above, its limit exist. In addition,

\[
P_k^{(i)} \cong \left( \frac{1}{d_i} \sum_{j \in N^{(i)}} \Sigma_w^{(i)} -1 + C_i^T \Sigma_v^{(i)} -1 C_i \right)^{-1}
\]

Thus, its limit is positive definite. \( \square \)

This proof shows that the covariance matrices of DKF converges when they are initialized as zero matrices. In comparison, there are few works on the convergence of the covariance matrices of DKF: [65] proves the convergence of the covariance in probabilistic terms and [36] analyzes an estimator that is defined differently. We remark the observability assumption is also used in the convergence of the covariance in the standard Kalman filter and in this sense it is the optimal assumption we could expect.

Following is the main result of this estimator:

**Theorem 5.3.2.** Under the assumptions in Theorem 6.5.1, the algorithm (6.25) converges to the correct solution in the sense that for all \( 1 \leq i \leq n \), \( \lim_{k \to \infty} \| \tilde{x}_k^{(i)} - x_k \| \to 0. \)

The result described here is called the “omniscience property” in [35, 41], which is proved under the same setting as Theorem 6.5.2, but for different algorithm.

We remark that while the condition “\((A, C)\) is observable” is more restrictive than the condition in [35] that “\((A, C)\) is detectable”, in practice the difference could be addressed using the idea of decomposing the system \((A, C, x)\) into two parts corresponding to stable and unstable eigenvalues of \(A\). Note that for \(x\) the stable part converges to zero, thus it is sufficient to investigate the subsystem of \((A, C, x)\) that is associated with unstable eigenvalues of \(A\). More specif-
ically, let $A = U \text{diag}(S_1, S_2) U^{-1}$ be the Jordan transformation of $A$, where $U$ is the similarity transformation matrix, $S_1$ is a square matrix that contain all Jordan blocks with stable eigenvalues and $S_2$ consist of all Jordan blocks with unstable eigenvalues. Then, with $\tilde{x}_k = U^{-1} x_k$ and $\tilde{x}_k = [\tilde{x}_{1,k}, \tilde{x}_{2,k}]$, the state evolution of (6.22) is equivalent to the following equations:

$$
\tilde{x}_{k+1,1} = S_1 \tilde{x}_{k,1}, \tilde{x}_{k+1,2} = S_2 \tilde{x}_{k,2},
$$

Now, we have $\|\tilde{x}_{k,1}\| \to 0$ as $k \to \infty$. Thus, it is sufficient to estimate $\tilde{x}_{k,2}$. To have the “omni-science property” of the estimation of $\tilde{x}_{k,2}$ from $y_k^{(i)} = C^{(i)} U \tilde{x}_k \approx C^{(i)} U_2 \tilde{x}_{k,2}$ ($U_2$ is a submatrix of $U$ corresponding to the component $S_2$), Theorem 6.5.2 implies that is sufficient to have the observability of $(S_2, C U_2)$. Applying the “Eigenvalue assignment” from [66, Table 15.1], it can be shown that the observability of $(S_2, C U_2)$ is equivalent to the detectability of $(A, C)$.

### 5.3.2 Distributed estimation with attack

In this section, we investigate the case with attack, which is given by the following model,

$$
x_{k+1} = A x_k, \ y_k^{(i),a} = C^{(i)} x_k + a_k^{(i)}
$$

and we propose the following estimator based on optimization:

$$
\hat{x}_k^{(i)} = \arg \min_{\hat{x}_k} \lambda \left\| \Sigma_u^{(i)} \frac{1}{2} (y_k^{(i),a} - C^{(i)} x_k) \right\| \\
+ \frac{1}{d_i} \sum_{j \in N^{(i)}} (x_k - A \hat{x}_k^{(j)\perp})^T P^{(j)} (x_k - A \hat{x}_k^{(j)\perp}) (5.8)
$$

This method is motivated from the estimator (6.25) as follows:
First, (6.25) can be considered as the following optimization problem

\[
\hat{x}_k^{(i)} = \arg \min_{x_k} (y_k^{(i),a} - C^{(i)} x_k) \Sigma_v^{(i)} (y_k^{(i),a} - C^{(i)} x_k)^T \\
+ \frac{1}{d_i} \sum_{j \in N^{(i)}} (x_k - A \hat{x}_k^{(j)}) P_l^{-1} (x_k - A \hat{x}_k^{(j)})
\]  

(5.9)

To make an optimization-based estimator more robust to attacks, a commonly used strategy is to use optimization with \(\ell_1\) norm on the terms affected by attack [31]. We apply a similar strategy, where we replace \((y_k^{(i),a} - C^{(i)} x_k) \Sigma_v^{(i)} (y_k^{(i),a} - C^{(i)} x_k)\) in (5.9) with its square root (with a scalar \(\lambda\)), which is similar to giving a smaller penalty on the attacked measurement \(y_k^{(i),a}\). This procedure makes our algorithm more resilient to attacks. The optimization problem in (5.8) does not have an explicit solution, but it can be solved efficiently as it is a convex optimization problem.

To analyze this estimator, we consider two scenarios:

1) All agents are benign and the system operates normally.
2) Some agents are compromised.

We provide the following theoretical guarantee for the first scenario. It suggests that when the initial estimation errors \(\hat{e}_0^{(i)}\) are not too large, the algorithm holds the “omniscience property”.

**Theorem 5.3.3.** Under the assumptions of Theorem 6.5.1, if the initial estimation errors \(\hat{e}_0^{(i)}\) satisfy the following condition

\[
\{ x : x^T P_l^{(i)} x = \hat{e}_0^{(i)} \Sigma_v^{(i)} \hat{e}_0^{(i)} \} \\
\subseteq \{ x : \| \Sigma_v^{(i)} \hat{e}_0^{(i)} \| \leq 2\lambda \},
\]  

(5.10)

then, for the first scenario without attack, the sequence produced by (5.8) converges to the correct solution i.e. for all \(1 \leq i \leq n\), \(\lim_{k \to \infty} \| \hat{x}_k^{(i)} - x_k \| \to 0\).

For the second scenario, the following theorem suggest that no matter how large are the magnitudes of the attacks, the output of the algorithm does not deviate too much compared to the
attack free scenario. It also suggests that when the number of attacks on agents are finite, the estimation error given by (5.9) is bounded.

**Theorem 5.3.4.** Consider the optimization problem (5.8). For different values of $y^{(i)\alpha}$, the norm of the difference of the estimated value $\hat{x}_k^{(i)}$ is at most

$$
\lambda d_i \left\| \sum_{j \in N^{(i)}} P^{(j)} (y^{(i)\alpha})^{-1} \Sigma_u^{(i)} \right\|^2.
$$

(5.11)

### 5.3.3 Proof of Main Results

#### 1. Proof of Theorem 6.5.2

We specify the estimation error as $e_k^{(i)} = \hat{x}_k^{(i)} - x_k$ and show that,

$$
e_k^{(i)T} P^{(i)} e_k^{(i)} \leq \frac{1}{d_i} \sum_{j \in N^{(i)}} e_{k-1}^{(j)T} P^{(j)} e_{k-1}^{(i)}.
$$

(5.12)

Applying (5.9), we have

$$
e_k^{(i)} = \arg \min_{e_k} f(e_k),$$

where

$$
f(e_k) = (C^{(i)} e_k)^T \Sigma_u^{(i)} (C^{(i)} e_k) + \frac{1}{d_i} \sum_{j \in N^{(i)}} (e_k - A e_{k-1}^{(j)})^T P^{(j)} (e_k - A e_{k-1}^{(j)}).
$$

(5.13)

Using the fact that $\nabla f(e_k)|_{e_k = e_k^{(i)}} = 0$, we have

$$
(C^{(i)} e_k^{(i)})^T \Sigma_u^{(i)} (C^{(i)} e_k^{(i)}) + \frac{1}{d_i} \sum_{j \in N^{(i)}} (e_k^{(i)} - A e_{k-1}^{(j)})^T P^{(j)} (e_k^{(i)} - A e_{k-1}^{(j)}) = 0.
$$

(5.14)
Combining (6.32) with \( f(e^{(i)}_k) \geq 0 \) gives,

\[
\frac{1}{d_i} \sum_{j \in N^{(i)}} (Ae^{(j)}_{k-1})^T \mathbf{P}^{(j)-1}_j A e^{(j)}_{k-1} \\
\geq (C^{(i)}e^{(i)}_k)^T \Sigma^{(i)}_{\nu}^{-1}(C^{(i)}e^{(i)}_k) + \frac{1}{d_i} \sum_{j \in N^{(i)}} e^{(i)}_k P^{(j)-1}_j e^{(i)}_k \\
= e^{(i)}_k P^{(i)-1}_i e^{(i)}_k. \tag{5.15}
\]

Since, \( P^{(i)-1}_i - A^T \mathbf{P}^{(j)-1}_j A = P^{(i)-1}_i - A^T (A P^{(i)} A^T + \Sigma^{(i)}_{\nu})^{-1} A = P^{(i)-1}_i - (P^{(i)} + A^{-1} \Sigma^{(i)}_{\nu} A^{-T})^{-1} \)

is positive semidefinite, (6.33) implies (6.30), and (6.30) implies that

\[
\max_{1 \leq i \leq n} e^{(i)}_k P^{(i)-1}_i e^{(i)}_k
\]

does not increase as a function of \( k \) and thus, it converges. However, it remains to be proven that it converges to zero. If this is not the case, then (6.33) achieves the equality \( (Ae^{(j)}_{k-1})^T \mathbf{P}^{(j)-1}_j A e^{(j)}_{k-1} = e^{(j)}_k P^{(j)-1}_j e^{(j)}_k \) and it implies that \( e^{(j)}_{k-1} = 0 \) for all \( j \in N^{(i)} \). Combining it with \( Ae^{(j)}_{k-1} = e^{(i)}_k \) (which follows from the equality \( f(e^{(i)}_k) = 0 \)), we get \( e^{(i)}_k = 0 \).

2. Proof of Theorem 6.5.3

We follow the proof of Theorem 6.5.2 and differentiate the objective function of (5.8), which gives us

\[
\lambda \frac{(C^{(i)}e^{(i)}_k)^T \Sigma^{(i)}_{\nu}^{-1}(C^{(i)}e^{(i)}_k)}{\|\Sigma^{(i)}_{\nu}^{\frac{1}{2}}(C^{(i)}e^{(i)}_k)\|} + \frac{1}{d_i} \sum_{j \in N^{(i)}} (e^{(i)}_k - A e^{(j)}_{k-1})^T \mathbf{P}^{(j)-1}_j e^{(i)}_k = 0. \tag{5.16}
\]
As a result

\[
\frac{1}{d_i} \sum_{j \in N(i)} (e_k^{(i)} - A e_{k-1}^{(j)})^T P_{(j)}^{-1} (e_k^{(i)} - A e_{k-1}^{(j)}) \geq 0
\]

implies that

\[
\frac{1}{d_i} \sum_{j \in N(i)} (A e_{k-1}^{(j)})^T P_{(j)}^{-1} A e_{k-1}^{(j)} \geq 2 \lambda \left\| \frac{1}{2} (C(i)e_k^{(i)}) \right\| + \frac{1}{d_i} \sum_{j \in N(i)} e_k^{(i)} P_{(j)}^{-1} e_k^{(i)}
\]

\[
= e_k^{(i)} P_{(i)}^{-1} e_k^{(i)},
\]

(5.17)

if \( \left\| \frac{1}{2} (C(i)e_k^{(i)}) \right\| \leq 2 \lambda \).

Using the assumption on the initialization (6.28), it can be proved that \( \max_{1 \leq j \leq n} (A e_{k-1}^{(j)})^T P_{(j)}^{-1} A e_{k-1}^{(j)} \)

is decreasing and following the proof of Theorem 6.5.2, it converges to zero. Therefore, the theorem is proved.

3. Proof of Theorem 6.5.4

First we introduce the following lemma.

**Lemma 5.3.5.** When \( A \) is a square matrix and \( Q \) is positive definite, then the minimizer of \( x^T Q x + \lambda \left\| A x - a \right\| \), \( \hat{x} \), satisfies the following

\[
\left\| \hat{x} \right\| \leq \frac{\lambda}{2} \left\| Q^{-1} A \right\|.
\]

**Proof.** The gradient of the objective function \( x^T Q x + \lambda \left\| A x - a \right\| \) of the minimizer should be zero, i.e.

\[
2Q \hat{x} + \lambda A \frac{A \hat{x} - a}{\left\| A \hat{x} - a \right\|} = 0.
\]

So, \( \hat{x} = -\frac{\lambda}{2} Q^{-1} A \frac{A \hat{x} - a}{\left\| A \hat{x} - a \right\|} \) and \( \left\| \hat{x} \right\| \leq \frac{\lambda}{2} \left\| Q^{-1} A \right\| \).

Based on Lemma 6.5.5, we have the following result:
For any $a_1, a_2$, the minimizers of

$$(x - x_0)^TQ(x - x_0) + \lambda \|Ax - a_1\|$$

and

$$(x - x_0)^TQ(x - x_0) + \lambda \|Ax - a_2\|$$

are at most $\lambda \|Q^{-1}A\|$ apart.

Now, we can prove the theorem. Note that the optimization problem (5.8) leads to:

$$e_k^{(i)} = \arg \min_{e_k} \lambda \left\| \sum_{i} \frac{1}{2} (a_k^{(i)} - C^{(i)} e_k) \right\|$$

$$+ \frac{1}{d_i} \sum_{j \in N^{(i)}} (e_k - Ae_k^{(j)})^T P^{-1}(j) (e_k - Ae_k^{(j)}). \tag{5.18}$$

Thus, even for different attack vectors $a_k^{(i)}$, the difference of their solutions are bounded above by (6.29).

### 5.4 Experimental Results

In this section, we use a numerical example to demonstrate the effectiveness of our resilient state estimation approach against sensor attacks on the nodes of distributed CPS. We represent the dynamics of the system using a linear time-invariant model given by (5.1). The sensor measurement equations of an agent with and without attacks are given by (5.2) & (5.3). We generate a random system matrix, $A$, which is nondegenerate and random output matrices for all the agents, $\{C^{(i)}\}_{i=1}^n$. The dimension of the state $x$ is 20 and the number of sensors per agent is 14.

The undirected communication graph of the system, as shown in Figure 5.1, consists of 10 randomly located agents and 12 edges. Different shades of color are representative of nodes with different degrees. We simulate the distributed system, communication graph, and the sensor attack
on nodes in MATLAB.

![Communication network of a distributed system of 10 agents and 12 edges](image)

Figure 5.1: Communication network of a distributed system of 10 agents and 12 edges

We first evaluate our algorithm over the attack free scenario. Figure 5.2 compares estimation error of all the agents over a time frame of 50 (0 : 50). Along the Y-axis is normalized state estimation error of all the agents. We observe that the estimation errors converge within 10 seconds.

![Performance of resilient distributed estimation algorithm in the attack free case](image)

Figure 5.2: Performance of resilient distributed estimation algorithm in the attack free case.
Next, we consider the scenario where an adversary corrupts sensor measurements of nodes 1, 2, 3, 5, 8 & 10 after a certain time point. In case of the attack, malicious data of random value are added to all the sensor output of the compromised nodes. We assume that the attack vector \( \{a_i^k\}_{i=1,2,3,5,8,10} \) are injected after time point \( k = 20 \) into the nodes and the probability of its occurrence at any time after initiation at the nodes is \( p_a = 0.9 \). For instance, in our experiment, the attack occurs at time points such as 22, 25, 32, 35, 37 and 45. Figure 5.3 provides comparison of estimation error among all nodes for all the normalized states. We observe that the estimation errors are high for the neighbors, \( \{3, 4, 8\}, \{9\}, \{1, 8\}, \{6, 7\}, \{1, 3, 6, 7, 10\}, \{8, 9\} \), of the compromised nodes 1, 2, 3, 5, 8, 10, respectively. Note that the estimates obtained from our filter at attack time points does not get unbounded. At all other time points (when there is no attack), our method performs as well as in the attack free case.

![Figure 5.3: Performance of resilient distributed estimation algorithm during attack. Agents under attack are marked with (a). Estimation error is bounded and small.](image)

We also observe the performance of the non resilient estimator (5.9) during attack, as is shown in Figure 5.4. We see that by replacing \((y_k^{(i),a} - C^{(i)}x_k)^T\Sigma^{-1}(y_k^{(i),a} - C^{(i)}x_k)\) in (5.9) with its square root (with a scalar \( \lambda \)), our solution becomes more resilient and bounded during attacks.
Figure 5.4: Performance of non resilient distributed estimation algorithm during attack. Agents under attack are marked with (a). Estimation error is large.

5.5 Summary

We have proposed a novel passive attack resilient distributed state estimation algorithm that can recursively estimate states within an error bound without restricting the number of agents that can be compromised. We show using a numerical example that the estimation error of our method asymptotically convergence to zero when there is no attack and has an upper bound during attack. In future, we plan to improve our current analysis on estimation errors to stochastic systems and also intend to develop new attack detection procedures.
CHAPTER 6: RESILIENT STATE ESTIMATION IN VEHICLE PLATOON

6.1 Overview

In this chapter, we model the longitudinal dynamics of vehicle’s of a platoon using a third order state space model. Malicious attack corrupts the measurements of some sensors of a vehicle. Consequently, we develop a passive attack resilient distributed filter and provide theoretical guarantees. We show the asymptotic convergence of estimation error to zero when there is no attack and we provide an upper bound on the error during attack.

6.2 System Model

We consider a longitudinal platoon of \( n \) cooperative adaptive cruise control (CACC) equipped vehicles without attack. In Fig.6.1, \( c^{(i)} \) is the position of \( i \)-th vehicle in the traffic, where \( i = 0, \ldots, n \) and \( d^{(i)} = (c^{(i-1)}(t) - c^{(i)}(t)) \) is actual distance between vehicle \( i-1 \) and its follower \( i \). The active sensors and the vehicle to vehicle (V2V) communication devices on the cars are responsible for gathering these measurements. We assume that the leader car of the platoon has index \( i = 0 \), and followers of the platoon have index \( i = 1, \ldots, n \). These vehicles should maintain a desired distance \( d^{(i)} \) to its preceding vehicle, which in our case is proportional to the headway time (\( \tau_h = 3 \) sec) between the vehicles, minimum stopping distance (\( d_r = 5 \) m), and speed of the follower vehicle (\( v^{(i)} \)).

\[
d^{(i)}(t) = d_r + \tau_h v^{(i)}(t), \quad 1 \leq i \leq m \quad (6.1)
\]

According to [61], the spacing-policy (6.1) improves road efficiency, safety and disturbance
attenuation. Now, the objective of the local control law is to regulate the following distance and velocity errors to zero in the platoon

\[ e^{(i)}(t) = d^{(i)}(t) - d^{r,(i)}(t) \]  \hspace{1cm} (6.2) \\
\[ \dot{e}^{(i)}(t) = v^{(i-1)}(t) - v^{(i)}(t) - \tau^{i}_h a^{(i)}(t) \]  \hspace{1cm} (6.3) 

Subsequently, a platoon of homogeneous vehicles (the dynamics and controller governing all the vehicles are identical) is considered and longitudinal dynamics of the \( i \)-th follower vehicle is described by the following set of linear equations: [67]

\[ \dot{d}^{(i)}(t) = v^{(i-1)} - v^{(i)} \]
\[ \dot{v}^{(i)}(t) = a^{(i)} \]
\[ \dot{a}^{(i)}(t) = -\frac{1}{\tau} + \frac{1}{\tau} u^{(i)} \]  \hspace{1cm} (6.4) 

\( d^{(i)}(t), \ v^{(i)}(t), \ a^{(i)}(t) \) are distance, velocity, and acceleration of the \( i \)-th vehicle and \( \tau = 1.008 \) is a constant that represent inertial delay of vehicles longitudinal dynamics and is assumed to be identical for all vehicle.

6.2.1 Modeling Communication Network

We model the vehicle-to-vehicle communication of the platoon with the help of a graph. We assume that the platoon includes \( n \) follower vehicles and one leader (0), \( X = \{0\} \cup \{1, 2, \ldots, n\} \). The information flow among followers and leader is given by a directed graph \( G = (\mathcal{V}, \mathcal{E}) \). In the graph, nodes represent the leader and followers, \( \mathcal{V} = X \), whose dynamics are given by (6.4) and edges, \( \mathcal{E} = \mathcal{V} \cup \mathcal{V} \), represent communication between them. Here, \((i,j) \in \mathcal{E}\) is a unidirectional edge between \( i \) and \( j \), that enable \( i \) to send messages to \( j \). Neighborhood of \( i \in X \setminus \{0\} \) is defined as the set of nodes that are adjacent to it i.e. \( N^{(i)} = \{i\} \cup \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\} \) and with
whom it can communicate. In our case, the preceding car of a follower is its neighbor. To describe the information exchange among followers, we use an adjacency matrix, \( M = [m^{(i)(j)}] \in \mathbb{R}^{n \times n} \), where,

\[
m^{(i)(j)} = \begin{cases} 
1 &: (i, j) \in \mathcal{E} \\
0 &: (i, j) \notin \mathcal{E}
\end{cases}
\]

Furthermore, we model the communication between the leader and its followers with a directed spanning tree, which is a sub-graph of \( G \). As such, there exists a directed path via which leader exchanges its information with every follower. In this chapter, we use the words vehicle and node interchangeably.

### 6.2.2 Closed-loop Platoon Dynamics

The CACC equipped vehicles of the platoon have distributed controller that rely on information of neighbors, given by the set \( N^{(i)} \), to compute the desired acceleration. The controller uses a combination of DSRC based feed forward input, \( u_{ff}^{(i)} \), and sensor measurement based feedback input, \( u_{fb}^{(i)} \), to obtain the following control input of the plant:

\[
u^{(i)} = u_{fb}^{(i)} + u_{ff}^{(i)}
\]

(6.5)

The radar and internal sensors of the CACC unit measures the follower and preceding vehicles distance and velocity, which are used to calculate inter-vehicular errors, \( e^{(i)}, \dot{e}^{(i)} \). Subsequently, a proportional and derivative (PD) feedback controller operates on the errors to generate the following input:

\[
u_{fb}^{(i)} = k_p e^{(i)} + k_d \dot{e}^{(i)}
\]

(6.6)

where, \( k_p \) and \( k_d \) are the proportional and derivative gains of the linear controller, respectively.
Here, \( e^{(i)} \) (given by (6.2)) and \( \dot{e}^{(i)} \) (given by (6.3)) are relative distance and velocity errors respectively.

The feed forward controller uses acceleration data of the directly preceding vehicle \( \dot{a}^{(i-1)} \) and the leader \( \dot{a}^{(0)} \) (obtained by DSRC) to improve vehicle following and string stability performance. The feed forward control input is given by the following equation:

\[
\dot{u}_{ff}^{(i)} = -\frac{1}{\tau_h^{(i)}} u^{(i)}_{ff} + \frac{1}{\tau_h^{(i)}} (\ddot{a}^{(i-1)}) + \frac{1}{\tau_h^{(i)}} (\ddot{a}^{(0)})
\]  

(6.7)

In the case where the vehicles are behaving correctly then \( \dot{a}^{(i-1)} = a^{(i-1)} \) and \( \dot{a}^{(0)} = a^{(0)} \) during the update periods. However, in general we do not assume true values of accelerations via DSRC in order to account for malicious behavior.

Now, the closed-loop dynamics of CACC equipped vehicle is obtained by combining the longitudinal dynamics model (6.4) with the feedback control law (6.6) and the feed forward control law (6.7). By choosing the state variable as \( x^{(i)T} = [d^{(i)} \ v^{(i)} \ a^{(i)} \ u^{(i)}_{ff}] \in \mathbb{R}^4 \), the state space representation of the \( i \)-th CACC equipped vehicle in an \( n \)-vehicle string is given by the following time-invariant linear equations

\[
\dot{x}^{(i)} = A^{(i), (i)} x^{(i)} + A^{(i), (i-1)} x^{(i-1)} + B_s^{(i)} u^{(i)} + B_c^{(i)} \dot{a}^{(i-1)} + B_{\dot{a}}^{(i)} \ddot{a}^{(0)}
\]  

(6.8)

\[
y^{(i)} = C^{(i)} x^{(i)}
\]  

(6.9)

where, \( y^{(i)} \) is the output equation. We replace the control command, \( \dot{u}^{(i-1)} \) and \( \dot{u}^{(0)} \), received via
DSRC with \( \hat{a}^{(i-1)} \) and \( \hat{a}^{(0)} \) respectively. Also, the matrices of the above equation are

\[
A^{(i), (i)} = \begin{bmatrix}
0 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & -\tau^{-1} & 0 \\
0 & 0 & 0 & -\frac{1}{\tau h}
\end{bmatrix},
\]

\[
B^{(i)} = \begin{bmatrix}
0 \\
0 \\
\tau^{-1} \\
0
\end{bmatrix},
\]

\[
A^{(i), (i-1)} = \begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix},
\]

\[
B^{(i)}_{c} = \begin{bmatrix}
0 \\
0 \\
\frac{1}{\tau h} \\
0
\end{bmatrix},
\]

\[
C^{(i)} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix},
\]

where, \( B^{(i)}_{s} \) is the transformation vector corresponding to the input \( u^{(i)} \), which is generated using locally generated sensor data and \( B^{(i)}_{c} \) is the vector for the input (acceleration) of the preceding vehicle \( \hat{a}^{(i-1)} \) and leader \( \hat{a}^{(0)} \), which are received by the \( i \)th vehicle via DSRC.

The leader i.e. car 0 has a unique control law, which is \( u^{(0)} = u_{r} \), where \( u_{r} \) is the reference desired acceleration of the platoon. It is assumed that the leader receives \( u_{r} \) in real-time and no prediction is made on its value. Subsequently, the leader dynamics can be represented by the following equation:

\[
\dot{x}^{(0)} = A^{(0)} x^{(0)} + B^{(0)}_{s} u_{r} \tag{6.10}
\]

where,

\[
A^{(0)} = \begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & -\tau^{-1} & 0 \\
0 & 0 & 0 & 0
\end{bmatrix},
\]

\[
B^{(0)}_{s} = \begin{bmatrix}
0 \\
0 \\
\tau^{-1} \\
0
\end{bmatrix},
\]

For the closed loop dynamics of the homogeneous platoon, we aggregate the state and
output vectors of all the vehicles as

\[ X = \{ x^{(0)T}, x^{(1)T}, x^{(2)T}, \ldots, x^{(n)T} \} \]  \hspace{1cm} (6.11)

\[ Y = \{ y^{(0)T}, y^{(1)T}, y^{(2)T}, \ldots, y^{(n)T} \} \]  \hspace{1cm} (6.12)

We also define the collective inputs to the system as

\[ U^T = [u^{(0)}, u^{(1)}, \ldots, u^{(n)}] \]  \hspace{1cm} (6.13)

where, each vehicle \( i \) receives its input values \( u^{(i)}, u^{(0)} \) and \( u^{(i-1)} \) via sensors and DSRC.

Now, the homogeneous platoon of cars interconnected by a given information topology can be represented in the following compact form:

\[ \dot{X} = AX + BU \]  \hspace{1cm} (6.14)

\[ Y = CX \]  \hspace{1cm} (6.15)

where,

\[ A = \begin{bmatrix}
A_s^{(0)} & 0 & 0 & \ldots & 0 \\
A^{(2),(1)} & A^{(2),(2)} & 0 & \ldots & 0 \\
0 & A^{(3),(2)} & A^{(3),(3)} & \ldots & \vdots \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & A^{(n),(n-1)} & A^{(n),(n)}
\end{bmatrix} \]
\( B = \begin{bmatrix} B_s^{(0)} & 0 & 0 & \ldots & 0 \\ B_c^{(1)} & B_s^{(1)} & 0 & \ldots & 0 \\ B_c^{(2)} & B_c^{(2)} & B_s^{(2)} & \ldots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ B_c^{(n)} & 0 & \ldots & B_c^{(n)} & B_s^{(n)} \end{bmatrix} \),

\( C = I^{(n)} \otimes C^{(i)} \), \hspace{1cm} (6.16)

\( I^{(n)} \) is the identity matrix and symbol \( \otimes \) is the Kronecker product. The \( A \) matrix is derived from the adjacency matrix, \( M \).

From (6.14), we observe that platoon dynamics are a function of vehicles longitudinal dynamics \((A, B)\), the network topology \(M\), the distributed feedback and feed-forward control laws \((u_{fb}^{(i)}, u_{ff}^{(i)})\) and the spacing policy \(d_r^{(n)}\).
6.3 Attack Strategies

We consider two variants of sensor data manipulation attack and assume that the attacks do not corrupt measurements of all the follower vehicles after its initiation. In the first type, we assume that the adversary has control over followers of the platoon and it transmits wrong sensor data to the vehicle’s controller via its internal network. We define such an attack as False Data Injection (FDI), whose goal is to make the vehicle behave incorrectly. Consequently, such an attack can lead to destabilization of the platoon. The impact of such an attack on the output equation of the system can be given as:

\[ Y'_k = Y_k + \Gamma_k Y_{af} \]  \hspace{1cm} (6.17)

where, \( Y_{af} \) is the malicious injected data vector. The attack scenario can be analyzed with the given Bernoulli model: \( \mathbb{P}([\Gamma_k]_{i(i)} = 1) = 0, \forall i = 1, 2, \ldots, n, \ k < k_f \) and \( \mathbb{P}([\Gamma_k]_{i(i)} = 1) = p_{af}, \forall i = 1, 2, \ldots, n, \ k \geq k_f \), where \( p_{af} \) is the probability of successfully injecting false data after time \( k_f \).

In the second variant, we consider the attacker is present outside the target vehicle. As such the adversary uses tools such as Jammer to blind the sensors (radar, LIDAR, or ultrasonic) of the targeted follower. We define such an attack as Denial of Service (DoS), whose impact on the system can be captured by the following attacked output signal, \( Y'_k \):

\[ Y'_k = Y_k - \Gamma_k (Y_k - Y_{\tau}) \]  \hspace{1cm} (6.18)

where, \( \tau \) is the time at which last good measurement, \( Y_{\tau} \), was obtained and \( \Gamma_k \in \mathbb{B}^{q \times q} \) is a Binary diagonal matrix whose \( i \)th diagonal entry when \( [\Gamma_k]_{i(i)} = 1 \), indicates the DoS attack on vehicle \( i \in n \) and \( [\Gamma_k]_{i(i)} = 0 \), shows its absence. Such an attack on vehicles can be represented using the following Bernoulli model: \( \mathbb{P}([\Gamma_k]_{i(i)} = 1) = 0, \forall i = 1, 2, \ldots, n, \ k \leq \tau \), and \( \mathbb{P}([\Gamma_k]_{i(i)} = 1) = p_{aj}, \forall i = 1, 2, \ldots, n, \ k > \tau \), where \( p_{aj} \) is the probability of successfully
jamming the sensor data after time \( \tau \). Now, an adversary can carry out any one of these attacks, but not simultaneously.

### 6.4 Problem Description

Given a homogeneous platoon of \( n \) vehicles and a leader represented by linear time-invariant system model, linear measurement model, and directed communication graph \( G \), design a filter that can estimate system states such that \( \lim_{k \to \infty} \| \hat{X}_k^{(i)} - X_k \| \to 0, \forall i \in \mathbb{R}^n \) when there is no attack and the estimation errors are bounded when measurement of a subset of sensors of a vehicle of the platoon are compromised by an attack.

### 6.5 Methodology

In this section, we first propose an estimator (6.25) in Section 6.5.1 for no attack scenario and its convergence is proved in Theorem 6.5.2. Based on its optimization interpretation, we propose a novel estimator based on lasso in (6.26) and analyze its properties in Section 6.5.2. The proofs are provided in Section 6.5.3.

#### 6.5.1 Distributed estimation without attack

Following from Bayesian perspective of the Kalman filter with restricted information to the observations of \( i \) and the estimations of the neighbors of vehicle \( i \), we assumes that \( y_k^{(i)} = C^{(i)} X_k + v_k^{(i)} \) with \( v_k^{(i)} \sim N(0, \Sigma_v^{(i)}) \) and \( X_{k+1} = AX_k + w_k^{(i)} \) with \( w_k^{(i)} \sim N(0, \Sigma_w^{(i)}) \), the estimator can be
described by

\[
P_{k|k-1}^{(i)} = A P_{k-1}^{(i)} A^T + \Sigma^{(i)}
\]

\[
P_k^{(i)} = \left( \frac{1}{d_i} \sum_{j \in N^{(i)}} P_{k|k-1}^{(i)} + C_i^T \Sigma_v^{(i)} C_i \right)^{-1},
\]

\[
\hat{X}_k^{(i)} = P_k^{(i)} \left( \frac{1}{d_i} \sum_{j \in N^{(i)}} P_{k|k-1}^{(j)} A \hat{X}_k^{(j)} + C_i^T \Sigma_v^{(i)} y_k^{(i)} \right).
\]

In this section, we investigate the attack free case for the following model,

\[
X_{k+1} = AX_k, \quad y_k^{(i)} = C^{(i)} X_k
\]

We assume that the estimation error covariance matrix, \( P^{(i)} \), is chosen according to the following equation,

\[
P^{(i)} = \left( \frac{1}{d_i} \sum_{j \in N^{(i)}} (A P^{(i)} A^T + \Sigma^{(i)} - 1) \right.

\[
\left. + C^{(i)} T \Sigma_v^{(i)} - 1 C^{(i)} \right)^{-1}
\]

where, \( N^{(i)} = \{ i \} \cup \{ \text{neighbors of } i \text{ in } G \} \) and \( d_i = |N^{(i)}| \) is the total number of neighbors of node \( i \). Now, our proposed estimator has the following prediction rules,

\[
P_{|}^{(i)} = A P^{(i)} A^T + \Sigma^{(i)}
\]

\[
\hat{X}_k^{(i)} = P_k^{(i)} \left( \frac{1}{d_i} \sum_{j \in N^{(i)}} P_{|}^{(j)} A \hat{X}_k^{(j)} + C^{(i)} T \Sigma_v^{(i)} - 1 y_k^{(i)} \right)
\]

where, \( \hat{X}_k^{(i)} \) is the state estimate and \( P_{|}^{(i)} \) is an a priori estimate covariance of vehicle \( i \). This
estimator is motivated from the Distributed Kalman Filter, which has been studied in many works such as [34, 36–38, 64]. While $\Sigma_v^{(i)}$ and $\Sigma_w^{(i)}$ are commonly used to denote the covariance of the noise in the system, here they have no physical meaning and are treated as parameters for developing the algorithm. In principle, they can be chosen to be any positive definite matrices and in our simulations we let them to be the identity matrices $I$.

To apply this estimator, we need to ensure that a solution to (6.23) exist. Thus, we give the following theoretical guarantee:

**Theorem 6.5.1.** If the graph $G$ is connected, $A$ is full-rank, $(A, C)$ is observable, and $\Sigma_v^{(i)}$ is full rank for all $1 \leq i \leq n$, then there exist $\{P^{(i)}\}_{i=1}^n$ that satisfy (6.23).

**Proof.** In the proof, both $A \succeq B$ and $B \preceq A$ mean that $A - B$ is positive semidefinite.

Here, we let $P^{(i)}_0 = 0$ for all $1 \leq i \leq n$ and show that for the sequence $P^{(i)}_k$ generated by

$$P^{(i)}_{k|k-1} = AP^{(i)}_{k-1}A^T + \Sigma_v^{(i)}$$

$$P^{(i)}_k = \left( \frac{1}{d_i} \sum_{j \in \mathcal{N}^{(i)}(i)} P^{(i)-1}_{k|k-1} + C^{(i)T}\Sigma_v^{(i)-1}C^{(i)} \right)^{-1}$$

the limit, $\lim_{k \to \infty} P^{(i)}_k$, exist and it is a positive definite matrix for all $1 \leq i \leq n$. If this is true, then $P^{(i)} = \lim_{k \to \infty} P^{(i)}_k$ is a solution to (6.23).

We will first show that $P^{(i) -1}_1$ is bounded below by a positive definite matrix. For $k = 1$, we have

$$P^{(i) -1}_1 \succeq C^{(i)T}\Sigma_v^{(i)-1}C^{(i)}$$

which is positive semidefinite with range being the row space of $C^{(i)}$, i.e., $\{C^{(i)}z : z \in \mathbb{R}^q\}$. 

81
If \( j \in N^{(i)} \) (and by definition \( i \in N^{(i)} \)), then

\[
P^{(i)}_2^{-1} \geq \frac{1}{d_i} (A \Sigma^{(j)} W_i + \Sigma^{(i)} W_i)^{-1} + C_i \Sigma^{(i)} W_i^{-1} C_i^{-1} \\
+ \frac{1}{d_i} (A \Sigma^{(i)} W_i + \Sigma^{(i)} W_i)^{-1},
\]

which is a positive semidefinite matrix of range \( \{ C^{(i)} T z_1 + A C^{(i)} T z_2 + A C^{(i)} T z_3 : z_1, z_2, z_3 \in \mathbb{R}^q \} \).

By applying the same procedure to time \( k = 3, 4, \ldots \), we verify that for sufficiently large \( k \), the range of \( P^{(i)}_k^{-1} \) can be given by the linear combination of \( \bigoplus_j \{ A^{l(j)} C^{(i)} T z \} \) \( \forall j \) such that there exists a path from \( j \) to \( i \) of length \( l(j) \). It can be shown that for sufficiently large \( k \), the set contains the range of \( A^r C^T, A^{r+1} C^T, A^{r+2} C^T, \ldots \) for some positive integer \( r \). When \( (A, C) \) is observable, this range is \( \mathbb{R}^n \) and as a result, \( P^{(i)}_k^{-1} \) is larger than a positive definite matrix with full rank. This suggests that \( P^{(i)}_k \) is bounded by a positive definite matrix from above.

In addition, by induction it can be shown that \( P^{(i)}_k \) is strictly increasing in the sense that

\[
P^{(i)}_0 \preceq P^{(i)}_1 \preceq P^{(i)}_2 \preceq \cdots.
\]

Since, the sequence is bounded above, its limit exist. In addition,

\[
P^{(i)}_k \succ \left( \frac{1}{d_i} \sum_{j \in N^{(i)}} \Sigma^{(i)} W_i^{-1} + C_i \Sigma^{(i)} W_i^{-1} C_i^{-1} \right)^{-1}
\]

Thus, its limit is positive definite.

This proof shows that the covariance matrices of DKF converges when they are initialized as zero matrices. In comparison, there are few works on the convergence of the covariance matrices of DKF: [65] proves the convergence of the covariance in probabilistic terms and [36] analyzes an estimator that is defined differently. We remark the observability assumption is also used in the convergence of the covariance in the standard Kalman filter and in this sense it is the optimal
assumption we could expect.

Following is the main result of this estimator:

**Theorem 6.5.2.** Under the assumptions in Theorem 6.5.1, the algorithm (6.25) converges to the correct solution in the sense that for all $1 \leq i \leq n$, $\lim_{k \to \infty} \| \hat{X}_k^{(i)} - X_k \| \to 0$.

The result described here is called the “omniscience property” in [35, 41], which is proved under the same setting as Theorem 6.5.2, but for different algorithm.

We remark that while the condition “$(A, C)$ is observable” is more restrictive than the condition in [35] that “$(A, C)$ is detectable”, in practice the difference could be addressed using the idea of decomposing the system $(A, C, X)$ into two parts corresponding to stable and unstable eigenvalues of $A$. Note that for $X$ the stable part converges to zero, thus it is sufficient to investigate the subsystem of $(A, C, X)$ that is associated with unstable eigenvalues of $A$. More specifically, let $A = U \text{diag}(S_1, S_2) U^{-1}$ be the Jordan transformation of $A$, where $U$ is the similarity transformation matrix, $S_1$ is a square matrix that contain all Jordan blocks with stable eigenvalues and $S_2$ consist of all Jordan blocks with unstable eigenvalues. Then, with $\tilde{X}_k = U^{-1}X_k$ and $\tilde{X}_k = [\tilde{X}_{1,k}, \tilde{X}_{2,k}]$, the state evolution of (6.22) is equivalent to the following equations:

$$
\tilde{X}_{k+1,1} = S_1 \tilde{X}_{k,1}, \quad \tilde{X}_{k+1,2} = S_2 \tilde{X}_{k,2},
$$

Now, we have $\| \tilde{X}_{k,1} \| \to 0$ as $k \to \infty$. Thus, it is sufficient to estimate $\tilde{X}_{k,2}$. To have the “omniscience property” of the estimation of $\tilde{X}_{k,2}$ from $y_k^{(i)} = C^{(i)} U \tilde{X}_k \approx C^{(i)} U_2 \tilde{X}_{k,2}$ ($U_2$ is a submatrix of $U$ corresponding to the component $S_2$), Theorem 6.5.2 implies that is sufficient to have the observability of $(S_2, C U_2)$. Applying the “Eigenvalue assignment” from [66, Table 15.1], it can be shown that the observability of $(S_2, C U_2)$ is equivalent to the detectability of $(A, C)$. 

83
6.5.2 Distributed estimation with attack

In this section, we investigate the case with attack, which is given by the following model,

\[ X_{k+1} = AX_k, \quad y_k^{(i),a} = C^{(i)}X_k + a_k^{(i)} \]

and we propose the following estimator based on optimization:

\[
\hat{X}^{(i)} = \arg \min_{X_k, A_k^{(i)}} (y_k^{(i),a} - C^{(i)}X_k - a_k^{(i)})^T \Sigma^{(i)}_{\nu}^{-1}
\]

\[
(y_k^{(i),a} - C^{(i)}X_k - a_k^{(i)}) + \lambda \left\| a_k^{(i)} \right\|_1
\]

\[
\frac{1}{d_i} \sum_{j \in N^{(i)}} (X_k - A\hat{X}^{(j)}_{k-1})^T P^{(j)}_{\nu}^{-1}(X_k - A\hat{X}^{(j)}_{k-1})
\]

(6.26)

This method is motivated from the estimator (6.25) as follows:

First, (6.25) can be considered as the following optimization problem

\[
\hat{X}^{(i)} = \arg \min_{X_k} (y_k^{(i),a} - C^{(i)}X_k)^T \Sigma^{(i)}_{\nu}^{-1}(y_k^{(i),a} - C^{(i)}X_k)
\]

\[
\frac{1}{d_i} \sum_{j \in N^{(i)}} (X_k - A\hat{X}^{(j)}_{k-1})^T P^{(j)}_{\nu}^{-1}(X_k - A\hat{X}^{(j)}_{k-1})
\]

(6.27)

To make an optimization-based estimator more robust to attacks, a commonly used strategy is to use optimization with \( \ell_1 \) norm on the terms affected by attack [31]. We apply a similar strategy, where we replace \((y_k^{(i),a} - C^{(i)}X_k)^T \Sigma^{(i)}_{\nu}^{-1}(y_k^{(i),a} - C^{(i)}X_k)\) in (6.27) with a penalized term (with a scalar \( \lambda \)) on attacked values and subtract the attack value, \( a_k^{(i)} \), from the attacked measurement \( y_k^{(i),a} \). This procedure makes our algorithm more resilient to attacks. The optimization problem in (6.26) does not have an explicit solution, but it can be solved efficiently as it is a convex optimization problem.

To analyze this estimator, we consider two scenarios:
1) All vehicles are benign and the system operates normally.

2) Some vehicles are compromised.

We provide the following theoretical guarantee for the first scenario. It suggests that when the initial estimation errors $e_0^{(i)}$ are not too large, the algorithm holds the “omniscience property”.

**Theorem 6.5.3.** Under the assumptions of Theorem 6.5.1, if the initial estimation errors $e_0^{(i)}$ satisfy the following condition

$$
\{ X : X^T P^{(i)} - 1 X = e_0^{(i)T} P^{(i)} - 1 e_0^{(i)} \}
\subseteq \{ X : \| \Sigma^{(i)} - \frac{1}{2} C^{(i)} X \| \leq 2 \lambda \},
\tag{6.28}
$$

then, for the first scenario without attack, the sequence produced by (6.26) converges to the correct solution i.e. for all $1 \leq i \leq n$, $\lim_{k \to \infty} \| \hat{X}_k^{(i)} - X_k \| \to 0$.

For the second scenario, the following theorem suggest that no matter how large are the magnitudes of the attacks, the output of the algorithm does not deviate too much compared to the attack free scenario. It also suggests that when the number of attacks on vehicles are finite, the estimation error given by (6.27) is bounded.

**Theorem 6.5.4.** Consider the optimization problem (6.26). For different values of $y_k^{(i)}, a_k$, the norm of the difference of the estimated value $\hat{X}_k^{(i)}$ is at most

$$
\lambda d_i \left\| \left( \sum_{j \in N^{(i)}} P^{(j)} - 1 \right)^{-1} \Sigma^{(i)} - \frac{1}{2} C^{(i)} \right\|.
\tag{6.29}
$$

6.5.3 **Proof of Main Results**

1. **Proof of Theorem 6.5.2**
We specify the estimation error as $e_k^{(i)} = \hat{X}_k^{(i)} - X_k$ and show that,

$$e_k^{(i) T} P_k^{(i) -1} e_k^{(i)} \leq \frac{1}{d_i} \sum_{j \in N(i)} e_k^{(j) T} P_k^{(j) -1} e_k^{(j)}.$$  \hfill (6.30)

Applying (6.27), we have $e_k^{(i)} = \text{arg min}_{e_k} f(e_k)$, where

$$f(e_k) = (C^{(i)} e_k) T \Sigma_v^{(i)} -1 (C^{(i)} e_k) + \frac{1}{d_i} \sum_{j \in N(i)} (e_k - A e_{k-1}^{(j)}) T P_k^{(j) -1} (e_k - A e_{k-1}^{(j)})$$ \hfill (6.31)

Using the fact that $\nabla f(e_k) |_{e_k = e_k^{(i)}} = 0$, we have

$$\begin{align*}
(C^{(i)} e_k^{(i)}) T \Sigma_v^{(i)} -1 (C^{(i)} e_k^{(i)}) + \frac{1}{d_i} \sum_{j \in N(i)} (e_k^{(i)} - A e_{k-1}^{(j)}) T P_k^{(j) -1} e_k^{(i)} &= 0. \\
\text{(6.32)}
\end{align*}$$

Combining (6.32) with $f(e_k^{(i)}) \geq 0$ gives,

$$\begin{align*}
\frac{1}{d_i} \sum_{j \in N(i)} (A e_{k-1}^{(j)}) T P_k^{(j) -1} A e_{k-1}^{(j)} \\
\geq (C^{(i)} e_k^{(i)}) T \Sigma_v^{(i)} -1 (C^{(i)} e_k^{(i)}) + \frac{1}{d_i} \sum_{j \in N(i)} e_k^{(i) T} P_k^{(j) -1} e_k^{(i)} \\
= e_k^{(i) T} P_k^{(i) -1} e_k^{(i)}. \\
\text{(6.33)}
\end{align*}$$

Since, $P_k^{(i) -1} - A^T P_k^{(j) -1} A = P_k^{(i) -1} - A^T (A P_k^{(i) A^T + \Sigma_v^{(i)}})^{-1} A = P_k^{(i) -1} - (P_k^{(i)} + A^{-1} \Sigma_v^{(i)} A^{-1 T})^{-1}$
is positive semidefinite, (6.33) implies (6.30), and (6.30) implies that

$$\max_{1 \leq i \leq n} e_k^{(i)} P^{(i)-1} e_k^{(k)}$$

do not increase as a function of $k$ and thus, it converges. However, it remains to be proven that it converges to zero. If this is not the case, then (6.33) achieves the equality $(Ae_{k-1}^{(j)})^T P_{|j}^{(j)-1} Ae_{k-1}^{(j)} = e_k^{(j)} P^{(j)-1} e_k^{(j)}$ and it implies that $e_k^{(j)} = 0$ for all $j \in N^{(i)}$. Combining it with $Ae_{k-1}^{(j)} = e_k^{(i)}$ (which follows from the equality $f(e_k^{(i)}) = 0$), we get $e_k^{(i)} = 0$.

2. Proof of Theorem 6.5.3

We follow the proof of Theorem 6.5.2 and differentiate the objective function of (6.26), which gives us

$$\lambda \frac{(C^{(i)} e_k^{(i)})^T \Sigma_v^{(i)-1} (C^{(i)} e_k^{(i)})}{\|\Sigma_v^{(i)-\frac{1}{2}} (C^{(i)} e_k^{(i)})\|} + \frac{1}{d_i} \sum_{j \in N^{(i)}} (e_k^{(i)} - Ae_{k-1}^{(j)})^T P_{|j}^{(j)-1} e_k^{(i)} = 0. \quad (6.34)$$

As a result

$$\frac{1}{d_i} \sum_{j \in N^{(i)}} (e_k^{(i)} - Ae_{k-1}^{(j)})^T P_{|j}^{(j)-1} (e_k^{(i)} - Ae_{k-1}^{(j)}) \geq 0$$

implies that

$$\frac{1}{d_i} \sum_{j \in N^{(i)}} (Ae_{k-1}^{(j)})^T P_{|j}^{(j)-1} Ae_{k-1}^{(j)} \geq 2 \lambda \frac{(C^{(i)} e_k^{(i)})^T \Sigma_v^{(i)-1} (C^{(i)} e_k^{(i)})}{\|\Sigma_v^{(i)-\frac{1}{2}} (C^{(i)} e_k^{(i)})\|} + \frac{1}{d_i} \sum_{j \in N^{(i)}} e_k^{(i)} P_{|j}^{(j)-1} e_k^{(i)} = e_k^{(i)} P^{(i)-1} e_k^{(i)}, \quad (6.35)$$

if $\|\Sigma_v^{(i)-\frac{1}{2}} (C^{(i)} e_k^{(i)})\| \leq 2 \lambda$.  

87
Using the assumption on the initialization (6.28), it can be proved that \( \max_{1 \leq j \leq n} (A e^{(j)}_{k-1})^T P^{(j)-1} A e^{(j)}_{k-1} \) is decreasing and following the proof of Theorem 6.5.2, it converges to zero. Therefore, the theorem is proved.

3. Proof of Theorem 6.5.4 First we introduce the following lemma.

Lemma 6.5.5. When \( A \) is a square matrix and \( Q \) is positive definite, then the minimizer of \( X^T Q X + \lambda \| AX - a \| \), \( \hat{X} \), satisfies the following

\[
\| \hat{X} \| \leq \frac{\lambda}{2} \| Q^{-1} A \|.
\]

Proof. The gradient of the objective function \( X^T Q X + \lambda \| AX - a \| \) of the minimizer should be zero, i.e.

\[
2Q \hat{X} + \lambda A \frac{A \hat{X} - a}{\| AX - a \|} = 0.
\]

So, \( \hat{X} = -\frac{\lambda}{2} Q^{-1} A \frac{A \hat{X} - a}{\| AX - a \|} \) and \( \| \hat{X} \| \leq \frac{\lambda}{2} \| Q^{-1} A \| \). \( \square \)

Based on Lemma 6.5.5, we have the following result:

For any \( A_1, A_2 \), the minimizers of

\[
(X - X_0)^T Q (X - X_0) + \lambda \| AX - a_1 \|
\]

and

\[
(X - X_0)^T Q (X - X_0) + \lambda \| AX - a_2 \|
\]

are at most \( \lambda \| Q^{-1} A \| \) apart.
Now, we can prove the theorem. Note that the optimization problem (6.26) leads to:

\[
\hat{e}_k^{(i)} = \arg\min_{e_k} \lambda \left\| \sum_{v}^{(i)} - \frac{1}{2} (a_k^{(i)} - C^{(i)} e_k) \right\| + \frac{1}{d_i} \sum_{j \in N^{(i)}} (e_k - A e_k^{(j)})^T P_j^{(j)} (e_k - A e_k^{(j)}).
\]

Thus, even for different attack vectors \(a_k^{(i)}\), the difference of their solutions are bounded above by (6.29).

### 6.6 Experimental Results

In this section, we consider a platoon of five vehicles to demonstrate the effectiveness of our resilient state estimation approach against sparse sensor attacks. We represent the dynamics of vehicles of the platoon using a linear time-invariant model given by (6.8). The sensor measurement equations of a car with and without attacks are given by (6.9) and (6.17) and (6.18) respectively. The system matrix, \(A\), the input transformation matrix, \(B\), and the output matrix, \(C\), of the homogeneous platoon is given by (6.16). The dimension of the state, \(x\), of each vehicle is four and each of them is equipped with three sensors.

The directed communication graph of the platoon, as shown in Figure 6.2, consists of five nodes (representing homogeneous vehicles) and eight edges (representing radar and V2V communication). We simulate the vehicle dynamics of platoon, communication graph, and the sensor attack on vehicles in MATLAB.

- **Case 1: Attack free scenario**

We first evaluate our algorithm over the attack free scenario. Figure 6.3 compares estimation error of five vehicles over a time frame of 50 sec \((0 : 50)\). Along the Y-axis is normalized state estimation distance error of all the cars in meters (m). We observe that the estimation errors converge within 10 seconds.
**Case 2: Denial of Service (DoS) attack**

We consider a scenario where an adversary corrupt measurements of internal sensors of a vehicle after a certain time point. In the case of DoS attack, we assume that a self-screening jammer is used to jam two out of the three sensor signals of Car-2 after $k = 20$ sec. We consider a time frame of (0-50) sec and the probability of occurrence of the DoS attack at any time after initiation in car-2 is $p_a = 0.5$. For instance, in our experiment, the attack occurs at time points 21, 23, 27 – 29, 31, 39, 44 – 46 and 50 sec. Figure 6.4 provides comparison of error in estimating distance of all vehicles in the presence (Figure 6.4a) and absence (Figure 6.4b) of our attack resilient algorithm. We observe that the error in estimating distance is high for the neighbor, {3}, of the compromised...
car-2. Also, the error propagates to the neighbor, \( \{4\} \), of car-3. Note that the distance error obtained from our filter, shown in Figure 6.4a, at attack time points does not get unbounded. At all other time points (when there is no attack), our method performs as well as in the attack free case. Thus, the likelihood of our algorithm in preventing a collision is high as cars-(2, 3, 4) distance error \(< 1 \text{ m}\) is less than the minimum stopping distance, \( d_r = 5 \text{ m} \).

(a) Performance of resilient distributed estimator, (6.26), during DoS attack. Car under attack is marked with (a). Distance estimation error is bounded and small

(b) Performance of non resilient distributed estimator, (6.27), during DoS attack. Car under attack is marked with (a). Distance estimation error is large.

Figure 6.4: DoS attack: (a,b) comparison of distance error of attack resilient and non resilient distributed estimation algorithm.
• **Case 3: False Data Injection (FDI) attack**

In case of FDI attack on our experimental system, malicious data of random value are added to the distance and velocity outputs of car-2 at random time points and duration. We assume that the signal attack vector \( y_{ak}^{af} \) is injected into the output data after \( k = 20 \) sec, but the attack does not corrupt all the measurements after its initiation. We consider a time frame of (0-50) sec and the probability of occurrence of the FDI attack at any time after initiation in car-2 is \( p_a = 0.5 \). For instance, in our experiment, the attack occurs at time points such as 21, 23, 27 − 29, 31, 39, 44 − 46 and 50 sec. Figure 6.5 provides comparison of error in estimating distance of all vehicles, with (Figure 6.5a) and without (Figure 6.5b) the implementation of our attack resilient algorithm. We observe that the FDI attack effect distance estimation of the neighbor, \( \{3\} \), of the compromised car-2. However, compared to the DoS attack, the FDI attack has less influence on the distance estimates of the neighbor of car-2. Note that the distance error obtained from our filter, shown in Figure 6.5a, at attack time points does not get unbounded. At all other time points (when there is no attack), our method performs as well as in the attack free case. Thus, the likelihood of our algorithm in preventing a collision is high as cars-(2, 3) distance error (\(< 1 \, m\)) is less than the minimum stopping distance, \( d_r = 5 \, m \).

### 6.7 Summary

We have proposed a novel *passive* attack resilient distributed state estimation algorithm that can recursively estimate states within an error bound without restricting the number of sensors that can be compromised of a vehicle in the platoon. We consider a homogeneous platoon of five vehicles and demonstrate that the estimation error of our method asymptotically convergence to zero when there is no attack and has an upper bound during sparse Denial of Service (DoS) and False Data Injection (FDI) attacks on sensors of a vehicle. In future, we plan to improve our current analysis to stochastic systems and also intend to develop new attack detection procedures.
Figure 6.5: FDI attack: (a,b) comparison of distance error of attack resilient and non resilient distributed estimation algorithm.

(a) Performance of resilient distributed estimation algorithm during FDI attack. Car under attack is marked with (a). Distance estimation error is bounded and small.

(b) Performance of non resilient distributed estimation algorithm during FDI attack. Car under attack is marked with (a). Distance estimation error is large.
CHAPTER 7: QUANTIFYING TRUST IN AUTONOMOUS SYSTEM

7.1 Overview

In this chapter, we use a statistical probabilistic formal verification method for certifying controllers autonomous system as trustworthy and show its resiliency against attacks. In our approach, we assume that the sensors of the system are being compromised by an attacker via spoofing or jamming attacks. As a result, the control and decision making unit of the system get degraded sensor data, which makes the system behave abnormally. We also assume that the control and decision making unit, the actuators, and the diagnostics and fault management sub-system of the execution unit cannot be modified by an attacker. However, these units/sub-systems may be affected by internal and environmental disturbances. With these assumptions, we first estimate a set of safe states for the vehicle. This estimation is done at the diagnostics and fault management sub-system of the execution unit. Then, the safe states are used to monitor the outputs of the control and decision unit. When a suspicious state is detected by this sub-system, it alerts the reactive control sub-system, which is also in the execution unit. This system takes the necessary actions to keep the autonomous system within the realms of safety. As the reactive control sub-system is prone to sensor noise and environmental disturbance, we verify whether it satisfies the safety requirements with certain probability value. Based on the verification results we assign performance measurements to different controllers of the execution unit of the autonomous system. As 100% certification of autonomous system is nearly impossible, we adopt a probabilistic rating mechanism to assign confidence values to different control units of the system. Thus, we quantify trust in the system by adding ratings of all these controllers. We demonstrate our approach on an autonomous ground vehicle under sensor spoofing attacks.
7.2 Attack Model

In this chapter, we consider remote attacks, targeting data acquisition unit of the vehicle, which comprises of sensors [1, 16, 22, 68]. An autonomous vehicle has many sensors (radar, lidar, Global Positioning System (GPS), camera, ultrasonic, and inertial measurement unit (IMU)) for collecting data of its internal and external environment [27, 69]. Compromising the data gathered by these sensors can impact the decisions made by the motion control unit of the vehicle.

To degrade sensor data quality, an attacker has to carry out spoofing or jamming attack remotely. Following the attack models described in [1, 16, 22, 68], we assume that the attacker target the hardware layer (external sensors) and has limited knowledge of sensors firmware or software. We also assume that they have limited resources to carry out the remote spoofing attack. Furthermore, the attack can be mounted on the target vehicle while its stationary or in motion. By carrying out these attacks, the attacker intends to cause physical damage to the vehicle.

Based on the remote spoof attacks carried out in [1, 16, 22, 68], we can state that the lidar, GPS, radar, camera, and ultrasonic sensors of the vehicle can be compromised. We assume that the sensor fusion unit (shown in Figure 7.1), which gets processed data of compromised sensors, cannot rectify the malicious data. Other assumptions are that the adversary does not have access to the diagnostic and fault management sub-system of the vehicle and they cannot directly modify the control and decision unit.

7.3 Framework for Quantifying Trust in Autonomous System

In this section, we describe our framework for quantifying trust and its various components. At the beginning, we briefly describe the functional software architecture of the autonomous ground vehicle under consideration. Then, we outline the requirements for safe state estimation. Subsequently, we describe the method of our choice for modeling autonomous vehicle under adversarial attack. We verify the controller of the autonomous vehicle against probabilistic safety specification. The
confidence interval and confidence percentage obtained from the verification process helps us in quantifying trust on the vehicle.

### 7.3.1 Functional Software Architecture of Autonomous Ground Vehicle

The software architecture of the autonomous ground vehicle (AV) under consideration (See Figure 7.1) is similar to the ones in [27, 70]. The AV has six on-board sensors, global positioning system (GPS), inertial measurement unit (IMU), cameras, radars, lidars, and ultrasonic. Raw data from these sensors are processed into a form which can be understood by the hierarchical sensor fusion unit. The multi-sensor data fusion algorithm of this unit uses different combination of on-board sensor data for object detection and classification. This data is further combined with information from V2V/V2X communications and on-board maps to enhance perception of the vehicle beyond the capabilities of traditional sensors. The output of the sensor fusion unit is a 3-D map of the environment, which is used by the control and decision unit along with information of the current state of the vehicle to generate an optimal obstacle free trajectory\(^1\). Subsequently, the trajectory execution sub-system uses propulsion, steering, and braking components to execute the trajectory generated by the control and decision unit. The diagnostic and fault management (D&F) sub-system estimates safe states of the vehicle. It analyzes the generated trajectory to determine whether the vehicle is operating within the safety domain. In case of any deviation from the expected behavior, it issues command to the reactive control sub-system, which immediately responds by taking action such as braking to avoid collision. The reactive control sub-system operates in parallel with the trajectory execution unit and when a threat is identified, its output overrides normal operation of the vehicle.

In this chapter, we consider an adaptive cruise control unit, whose function is to drive the vehicle in an intended trajectory and avoid collision with other vehicles on the road.

---

\(^1\)Note: In Figure 7.1, the current state refer to a trajectory generated in a previous time step and future state refer to a new trajectory generated by the control and decision unit after analyzing information from sensor fusion unit.
7.3.2 Safe State Estimation

We consider an autonomous ground vehicle in which the D&F sub-system monitors the trajectory generated by the control and decision unit to ensure the vehicle operates safely. To perform monitoring, the D&F sub-system estimate a set of safe states based on the knowledge of overall system dynamics and disturbances arising from internal components and external environment.

We first construct an abstract interval around true sensor measurements using \textit{set-membership} method [71]. Then, we estimate the safe states of the system from these sensor values. The abstract interval is constructed by using manufacturers specifications about precision and accuracy of sensors, as well as physical limitations such as sampling jitter and synchronization error. If these information’s are not present, then an approximate abstract interval is constructed based on empirical data of initial sensor measurements. Smaller the size of the interval, higher is the confidence on the sensor measurements. Anything outside this abstract interval are bad sensor values (due to adversarial attack) and the trajectory of the system should never enter the bad states (measured from the bad sensor values). Information of safe and bad states enables the D&F sub-system to monitor the system.
7.3.3 Modeling of Autonomous Vehicle under Adversarial Attacks

Modeling of systems involving interaction of discrete and continuous dynamics (such as in autonomous ground vehicles) has been successfully carried out using hybrid system [47, 48]. Over the years, several computational frameworks have been developed for representing hybrid system such as hybrid program, hybrid automata (extends the traditional finite state automata by incorporating continuous dynamics), networks of priced timed automata (NPTA), and hybrid petri net. However, these frameworks did not allow randomness in the design. Consequently, the theory of stochastic hybrid system and its modeling formalisms such as Piecewise Deterministic Markov Processes (PMDP), stochastic hybrid automata, and Switched Diffusion Processes (SDP), were developed for capturing variability/uncertainty in the system and performing probabilistic analysis. This enabled modeling and analysis of uncertain autonomous systems.

In this chapter, we use NPTA with stochastic semantics [72] for modeling vehicular system dynamics and representing internal/external disturbances including adversarial attack. Using this approach we model the adaptive cruise control (ACC) system of the ground vehicle, which is shown in our case study.

7.3.4 Verification of Autonomous Ground Vehicle

When the D&F sub-system of the execution unit (See Figure 7.1) identifies a suspicious behavior (state), it issues command to the reactive control sub-system, which then carries out the actions to prevent fatal scenarios. To ensure the reactive control sub-system behave correctly, we verify that it carries out the safe actions within a probability threshold. As specifications for a controller may be violated once in a while due to internal or external disturbances, we resort to a probabilistic method of verification. With the help of this approach, we can generate probabilistic guarantees on whether the system’s state will transition to a safe state within certain time interval.

Model checking is one of the approach for verifying stochastic hybrid systems [73]. In this
approach, a stochastic model $\mathcal{M}$ of a system with an initial state $s_0$ is expressed as a transition system, behavioral specification such as safety property $\phi$ is expressed using bounded linear temporal logic, $\theta \in [0, 1]$ is a probability threshold, and $\triangledown \in \{>, <, \geq, \leq\}$. Using these notations, the probabilistic model checking problem can be formally stated as $\mathcal{M}, s_0 \models P_{\triangledown \theta}(\phi)$. The underlying algorithm of this technique explores the state-space of the model to decide whether $\mathcal{M}$ satisfies $\phi$ with a probability $>, <, \geq,$ or $\leq$ to a certain threshold $\theta$. If a case exists where the model does not satisfy the specification, a counterexample in the form of a trace is produced by the model checker.

In our chapter, we use Statistical probabilistic Model Checking (SMC) to verify the reactive control sub-system of the autonomous ground vehicle against probabilistic safety specifications (generated from knowledge of system dynamics). The specifications used in this tool are time bounded. Such a model checking approach combines randomized simulation (i.e., Monte Carlo simulation), statistical analysis, and model checking, and it is scalable to large designs. However, using large number of digital clocks (we use one clock) in the system model effects efficiency and scalability of this approach.

### 7.3.5 Quantification of Trust in the Autonomous Ground Vehicle

Our probabilistic approach for safety verification of the autonomous ground vehicle requires running the system for certain time units to obtain results. Based on the number of runs, we obtain an overall estimate of the correctness of the system. The results are represented in the form of confidence interval $[\theta - \epsilon, \theta + \epsilon]$ (where, $\theta \in [0, 1]$ is the probability assigned to the safety specification and $\epsilon$ is an approximation parameter) and confidence $1 - \delta$ (where $\delta$ is the confidence parameter). The value of confidence represents the probability of the specification satisfying the system model within the confidence interval for the runs of the system. Higher the value of confidence within the confidence interval, more is the likelihood of the vehicle system model satisfying the safety specification. We write specifications for all the units of the autonomous ground vehicle under consideration and obtain their respective confidence values for a fixed confidence interval. Based
on these values, we assign overall trust to the vehicle.

## 7.4 Case Study

As a case study, we consider adaptive cruise control (ACC) of an autonomous ground vehicle. ACC relies on data from camera, lidar, or radar to measure distance \(d\) between a follower car \((fc)\) and a leader car \((lc)\) on a lane. From distance measurement, velocity and acceleration of the \(fc\) is calculated. The safety requirement for collision avoidance between \(fc\) and \(lc\) is that the distance between them should always be greater than zero i.e. \(d > 0\). Under normal operation of the vehicle, the control and decision unit achieves this condition by adjusting the acceleration of \(fc\). However, under spoofing attack, the sensors will produce wrong measurements of distance \(d\). As a result, the control and decision unit will issue wrong commands to the execution unit, resulting in a possible vehicular collision. To mitigate this issue, diagnostics and fault management sub-system should detect this threat and alert the reactive control sub-system. Subsequently, the later unit should override normal operation and issue the brake command.

In our example, we consider safe distance \(d\) between \(fc\) and \(lc\) is 20 m. When \(d\) is \(< 20\) m between the vehicles, the reactive controller of \(fc\) should brake within 600 msec to avoid collision. We assume the adversary spoofs the lidar signals, thereby making the sensor detect fake vehicles at a distance \(d > 20\) m, when it is actually less than the safe distance. Due to this, the reactive controller will not be able to brake within the desired time, resulting in a collision. However, our safe state estimation and monitoring method in the D&F sub-system will detect this fault and notify the reactive controller within the desired time to prevent collision.

To verify this scenario, we use the UPPAL Statistical Model Checking (SMC) tool Version 4.1. We build a very simple timed automata of \(D&F\ sub-system\) (Figure 7.2a) with two states \((\text{safe} \text{ and} \text{ warning})\). The \text{safe} state indicate that the future state from the control and decision unit conform with the safe state estimated by the D&F sub-system. On the contrary, the
warning state indicate suspicious behavior. In our current set-up we assume that the adversary cannot modify the D&F sub-system directly, but as the estimation algorithm is not completely accurate, we assign uncertainty to the state variable of the automaton. The safe state is initialized with an uncertain variable clk_p uniformly between [1.00, 2.00]. The automaton transitions to the warning state when the value of clk_p is > 1 and it outputs obs_detected!, which is sent to the reactive controller sub-system (Figure 7.2b). From the warning state it transition back to the safe state when clk_p<=4 with an output obs_clear! and the variable clk_p resets to 0. In order to estimate the probability that the clk_p will change its value to > 1 within a simulation time bound of 1000 time units, the following specification will be checked.

\[ Pr[<= 1000](<> clk_p > 1) \]

The result of the tool for D&F automata is: confidence interval [0.9026, 1.00] with 95% confidence after having generated 36 runs. After the obs_detect! signal is received by the reactive control automata (Figure. 7.2b), it transition from Initial state to Command via Intermediate state in < 100 time units. In this transition, the controller sends the command signal_break to the actuator sub-system (Figure 7.2c) as an obstacle has been detected. At the Initial state, internal variable obs is initialized to 0. After transitioning to Command state, the obs value changes to 1. This is shown by the method obs_update(1) in Figure 7.2b. To estimate the probability that the internal variable obs changes value to 1 in < 100 time units, we use the following specification.

\[ Pr[<= 1000](<> time < 100 and obs == 1) \]

The result of the tool for reactive controller automata is: confidence interval [0.9026, 1.00] with 95% confidence after having generated 36 runs. Once the obstacle is clear the automaton transitions to the Initial state via Update state and the internal variable obs value is reset.
to 0, shown in Figure. 2b as obs_update(0). The output of this transition is the command signal_acc!, which is also sent to the actuator sub-system. When the actuator automata (Figure. 7.2c) receives the command signal_break! from the reactive control automata, it should transition from max_velocity state to the break state in < 500 time units. During the same transition, internal variable apply_break changes its value from 0 to 1. This is shown by the method break_update(1) of Figure. 2c. To estimate the probability that the internal variable apply_break changes its value to 1 in < 500 time units, we use the following specification,

\[ Pr[< 1000](<> time < 500 and apply_break == 1) \]

The result of the tool for actuator automata is: confidence interval [0.9026, 1.00] with 95% confidence.
confidence after having generated 36 runs. From the break state the automata transition to stop state in time \( > 10 \) time units. During this transition, variable acc is initialized to an initial value of 60 with an uncertain derivative \( dv \) uniformly between \([1.00, 2.00]\). This uncertainty indicates that the car never comes to a complete halt at the stop state. After receiving the command signal_acc! from the reactive controller, the vehicle transitions back to the acc state. During this transition, the value of internal variable apply_break changes from 1 to 0 as shown by the method break_update(0) of Figure. 2c. This transition indicates that the collision has been avoided and the vehicle resume its motion. All the safety specifications we considered, were satisfied in the confidence interval \([0.9026, 1.00]\) with 95% confidence after having generated 36 runs.

In Table.1 we have provided summary of trusted and untrusted signals of automatons of Figure. 2a, 2b, 2c. Now, based on the results of these automatons, trust on the autonomous ground vehicle under consideration is 95% for a confidence interval of \([0.9026, 1.00]\).

### 7.5 Summary

Trust plays an important role in adoption of autonomous systems such as self-driving vehicles. In this chapter, we have addressed the trust issue in autonomous systems using an estimation method and statistical model checking approach. The estimation method identifies safe states based on a set-membership method and with the help of system dynamic. The statistical model checking approach provide probabilistic guarantees on whether various sub-systems of the vehicle will satisfy safety specifications within some time interval. In our case study, we use the confidence results generated during verification process to assign trust on the adaptive cruise-control unit of the vehicle. In future, we intend to explain our estimation and verification methods in more sophisticated systems.
Table 7.1: Summary of Trusted and Untrusted Signals of Automatons

<table>
<thead>
<tr>
<th>Signals</th>
<th>Untrusted</th>
<th>Trusted</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>obs_detect!</td>
<td>✓</td>
<td></td>
<td>Triggered on encountering bad states</td>
</tr>
<tr>
<td>obs_clear!</td>
<td></td>
<td>✓</td>
<td>Doesn’t depend on bad states</td>
</tr>
<tr>
<td>obs_detect?</td>
<td>✓</td>
<td></td>
<td>Relies on obs_detect!</td>
</tr>
<tr>
<td>signal_break!</td>
<td></td>
<td>✓</td>
<td>Doesn’t depend on bad states</td>
</tr>
<tr>
<td>obs_clear?</td>
<td></td>
<td>✓</td>
<td>Doesn’t depend on bad states</td>
</tr>
<tr>
<td>signal_acc!</td>
<td></td>
<td>✓</td>
<td>Doesn’t depend on bad states</td>
</tr>
<tr>
<td>signal_break?</td>
<td></td>
<td>✓</td>
<td>Doesn’t depend on bad states</td>
</tr>
<tr>
<td>signal_acc?</td>
<td></td>
<td>✓</td>
<td>Doesn’t depend on bad states</td>
</tr>
</tbody>
</table>
CHAPTER 8: CONCLUSION

Trust plays an important role in adoption of autonomous systems such as self-driving vehicles. One way of ensuring trust is to make the system secure against cyber-and physical attacks. Toward this objective, we have proposed several attack resilient algorithm in this thesis.

First, we introduced an active security solution that involved challenge authentication for detection of two types of attacks: the Denial of Service (DoS) and the delay injection, on active sensors of autonomous systems. The recursive least square approach was used for estimation of sensor measurements when it was under attack. With these estimated measurements, safe control inputs of the autonomous CPS were derived, which enabled the system to recover and operate safely in the presence of attacks. A case study was presented to show resiliency of adaptive cruise control system of ground vehicle, that leveraged our proposed solutions to counter attacks. However, the detection method failed when an adversary with adequate resources could sample the incoming signals from active sensors faster than the defender.

Next, we proposed a novel passive security mechanism that involved design of an attack resilient filter that could recursively estimate states within an error bound, when sensors of the system were compromised. Our approach leveraged Bayesian interpretation of the Kalman filter and combined it with the $\chi^2$ detector to ensure safety of CPS against Denial of Service and False Data Injection attacks. The computational complexity of our method is $O(\max(n, q)^3)$, which is same as that of the Kalman filter and it performed better than the standard and the Robust Kalman filters during attack as was shown in the car-following case study.

Having demonstrated our security solutions for a single system, we next developed a novel attack resilient state estimation algorithm for distributed system that could recursively estimate states within an error bound without restricting the number of agents that can be compromised. We show using a numerical example that the estimation error of our method asymptotically converges to zero when there was no attack and has an upper bound during attack. We also extend this general
distributed resilient estimator to a vehicle platoon scenario and show its applicability in estimating states when system is under sparse attacks.

Finally, we formally verify using the statistical model checking approach that controllers of a real-time system such as a self-driving car with security solutions adheres to timing constrains. Our method provide probabilistic guarantees on whether various controllers of the vehicle satisfy safety specifications within some time interval. In our case study, we used the confidence results generated during verification process to assign trust on the adaptive cruise-control unit of the vehicle.
LIST OF REFERENCES


