A STUDY OF POTENTIAL SECURITY AND SAFETY VULNERABILITIES IN CYBER-PHYSICAL SYSTEMS

by

Ali Al-Hashimi

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

Approved:

Ryan Gerdes, Ph.D. Major Professor	Rajnikant Sharma, Ph.D. Committee Member
Jake Gunther, Ph.D. Committee Member	Don Cripps, Ph.D. Committee Member
David Geller, Ph.D. Committee Member	Richard S. Inouye, Ph.D. Vice Provost for Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2020

ABSTRACT

A Study of Potential Security and Safety Vulnerabilities in Cyber-Physical Systems

by

Ali Al-Hashimi, Doctor of Philosophy

Utah State University, 2020

Major Professor: Ryan Gerdes, Ph.D.
Department: Electrical and Computer Engineering

The objective of this dissertation is to study the performance of two examples of Cyber-Physical Systems (CPS) which operate in adversarial environments, wherein it is possible to modify the operation of one or multiple functionalities of the CPS and induce harmful impacts. In literature, such damaging actions are referred to as attacks. From a security perspective, the study and research of potential attacks on CPSs means defining possible vulnerabilities in the latter, that could be exploited by attackers, and suggesting counter-measures to deter or lessen the impacts of such attacks.

First, we study the behavior of vehicular platoons (CPS example 1) and whether it is possible to attack the sensors, with which each platooned vehicle is equipped, with False-Data Injection (FDI), or for an attacker to control one of the vehicles in the platoon and thereby generate sudden accelerating/decelerating movements. For the study concerned with vehicular sensor attacks, we consider a string of vehicular platoons driving in one direction. Several previous studies show that the automation system of the platooned vehicles cannot handle attacks against its sensors and, as a consequence, will revert control back to a human driver and effectively disband the platoon. Although such an action is meant to reduce the attack impacts, one or more of the following non-attacked platoons may induce unexpected behavior that, according to our results, lead to collisions. For that

reason, we suggest two mitigation solutions to be engaged by the non-attacked platoons, the goal being to eliminate, if possible, or at least reduce the number of collisions. One solution is centralized and formulated using the Model Predictive Control (MPC) technique. The other solution is decentralized, heuristic in nature and requires fewer computations when compared to the first solution.

Next, we focus on FDI attacks against vehicular sensors. For this part, we will employ the optimal control-based reachability analysis in order to determine which conditions allow such attacks to induce collisions and at which relative speeds. We conducted the analysis for FDI attacks against a single range or range-rate sensor, both of them (on one car), and against two range or range-rate sensors, either on the same car or two different cars. In all cases, the results showed the possibility of inducing collisions as a result of FDI attacks and at high relative speeds. Finally, we study the behavior of a single vehicular platoon, wherein the attacker is able to control one of the vehicles. A previous study indicate that such an attack may cause accidents, suggesting a mitigation scheme based on the sliding mode control technique. Although the suggested mitigation succeeded in reducing the collisions significantly, the movement of the intact vehicles is still influenced by the attacker-controlled vehicle. For this reason, we modify the suggested mitigation. Our modification will eventually lead to disbanding the platoon and, hence, releasing the non attacked vehicles from the control of the attacker.

The second direction of this dissertation is to study the behavior of Heating, Ventilating, and Air Conditioning (HVAC) systems (CPS example 2), used in smart buildings to regulate the indoor temperature, while suffering attacks on their temperature sensors. First, we formulate an MPC-based controller to track a desired temperature in each zone of the building. The formulated controller uses readings from temperature sensors, installed at various sections of the building, in the decision-making process and generates the appropriate control commands, or the required amount of air flow rate to each zone. Then, to deter potential attacks against the temperature sensors, such as manipulating their measurements, we suggest two Moving Target Defense (MTD) technique based algorithms. An

important factor that facilitates such attacks is the fact that the MPC controller is static in nature and, thus, attackers can easily induce predictable impacts, such as misleading the controller and, as a result, causing occupants' discomfort. Therefore, our suggested algorithms continuously select a subset of the installed sensors and feed their measurements to the MPC controllers. Furthermore, an optimal observer is employed in order to estimate the other temperatures of which sensors are not selected. As a result, the impacts of attacking the HVAC system's sensors are reduced.

(181 pages)

PUBLIC ABSTRACT

A Study of Potential Security and Safety Vulnerabilities in Cyber-Physical Systems

Ali Al-Hashimi

The work in this dissertation focuses on two examples of Cyber-Physical Systems (CPS), integrations of communication and monitoring capabilities to control a physical system, that operate in adversarial environments. That is to say, it is possible for individuals with malicious intent to gain access to various components of the CPS, disrupt normal operation, and induce harmful impacts. Such a deliberate action will be referred to as an attack. Therefore, some possible attacks against two CPSs will be studied in this dissertation and, when possible, solutions to handle such attacks will also be suggested.

The first CPS of interest is vehicular platoons wherein it is possible for a number of partially-automated vehicles to drive autonomously towards a certain destination with as little human driver involvement as possible. Such technology will ultimately allow passengers to focus on other tasks, such as reading or watching a movie, rather than on driving. In this dissertation three possible attacks against such platoons are studied. The first is called "the disbanding attack" wherein the attacker is capable of disrupting one platoon and also inducing collisions in another intact (non-attacked) platoon vehicles. To handle such an attack, two solutions are suggested: The first solution is formulated using Model Predictive Control (MPC) optimal technique, while the other uses a heuristic approach. The second attack is False-Data Injection (FDI) against the platooning vehicular sensors is analyzed using the reachability analysis. This analysis allows us to validate whether or not it is possible for FDI attacks to drive a platoon towards accidents. Finally, mitigation strategies are suggested to prevent an attacker-controlled vehicle, one which operates inside a platoon and drives unpredictably, from causing collisions. These strategies are based on

sliding mode control technique and once engaged in the intact vehicles , collisions are reduced and eventual control of those vehicles will be switched from auto to human to further reduce the impacts of the attacker-controlled vehicle.

The second CPS of interest in this dissertation is Heating, Ventilating, and Air Conditioning (HVAC) systems used in smart automated buildings to provide an acceptable indoor environment in terms of thermal comfort and air quality for the occupants For these systems, an MPC technique based controller is formulated in order to track a desired temperature in each zone of the building. Some previous studies indicate the possibility of an attacker to manipulate the measurements of temperature sensors, which are installed at different sections of the building, and thereby cause them to read below or above the real measured temperature. Given enough time, an attacker could monitor the system, understand how it works, and decide which sensor(s) to target. Eventually, the attacker may be able to deceive the controller, which uses the targeted sensor(s) readings and raises the temperature of one or multiple zones to undesirable levels, thereby causing discomfort for occupants in the building. In order to counter such attacks, Moving Target Defense (MTD) technique is utilized in order to constantly change the sensors sets used by the MPC controllers and, as a consequence, reduce the impacts of sensor attacks.

To my parents
Abdulkareem and Lubna

To my siblings
Ammar, Mayassa, and Amnah

ACKNOWLEDGMENTS

There are many people, whom I want to mention their names here, who have helped me in numerous ways on my path towards the completion of my Ph.D. First, and foremost, I would like to thank my supervisor Dr. Ryan M. Gerdes for all the time and effort he invested in me throughout the course of my study. His deep insights and positive manner have always been helpful and encouraging. He has been very patient with me and a great encourager and supporter. Then, my special thanks go to my committee members, Dr. Sharma, Dr. Gunther, Dr. Cripps, and Dr. Geller for their support and help, particularly for their patience in reading my dissertation draft. Also, I should specially thank Dr. Sharma for his consistent support and encouragement as well. Furthermore, I would like to thank and express my gratitude to Dr. Thidapat Chantem for her support and help with my research and also for the valuable guidance in writing my research.

In addition, my sincere thanks go to Dr. Ming Li, Dr. Madher Al-Findee, Dr. Mohammad Shekaramiz, Dr. Soodeh Dadras, Marwan Ahmed, Teif Al-Daneen, Waled Al-Dulaimi, Mohammed Al-Edhari, Salam Al-Rubaye, Leila Amanzadeh, Daniel Dunn, Abbas Hommadi, Frost Mitchell, Samuel Mitchell, Pratham Oza, David Petrizze, Imran Sajjad, Mingshun Sun, and many more for their support, friendship, and encouragement. Also, I want to thank Tricia Brandenburg and Heidi Harper from the ECE department.

Finally, I would like to take this space to express most sincere gratitude first to Gregory, Kathryn, and Scott Zaborski, I am blessed to be a part of your family, and last, but not least, to my mom, dad, and siblings for their consistent help and, in particular, moral support in my academic journey and other aspects of my life.

Ali Al-Hashimi

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

ACC	Adaptive cruise control
CACC	Cooperative adaptive cruise control
CPS	Cyber-physical system
DoS	Denial of service
FDI	False-data injection
GA	Genetic algorithm
HJ	Hamilton-Jacobi
HVAC	Heating, ventilating, and air conditioning
IDM	Intelligent drive model
LQR	Linear quadratic regulator
MPC	Model predictive control
MTD	Moving Target Defense
PID	Proportional integral derivative
ROS	Robot Operating systems
UAV	Unmanned aerial vehicle
VAV	Variable air flow volume
V2V	Vehicle-to-vehicle
V2I	Vehicle-to-infrastructure

CHAPTER 1

INTRODUCTION

The focus of this dissertation is the analysis of the behavior of a Cyber-Physical System (CPS) operating in an adversarial environment wherein it could be possible for a malicious individual, which we will refer to henceforth as an attacker, to gain access to various components of the CPS, disrupt their normal operation, and induce harmful impacts. Two examples of CPSs will be studied in this dissertation. The first CPS studied is vehicular platoons travelling on a highway. These are vulnerable to attacks against one of the automation system functionalities, such as local sensors, which can cause platoons to slow down and stop, resulting in an inefficient use of the road and greater fuel consumption. In more serious cases, such disruption may cause a number of the platooned vehicles to collide with each other, potentially leading to the loss of lives. Second, Heating, Ventilating, and Air Conditioning (HVAC) systems are studied, wherein the measurements of installed sensors can be manipulated to feed incorrect data to the corresponding controllers, and as a result, temperature of the controlled thermal zone will increase or decrease, thereby leading to loss of thermal energy and discomfort among the building's occupants.

In general, a CPS may be defined as an integration of sensing, communication, and computation capabilities in order to monitor and control a physical process. Dependence upon CPS applications is growing steadily in applications such as transportation, smart buildings, energy and power grids, and manufacturing. Clearly, many, if not all, CPS applications are safety-critical, and any failure in their operation could lead to permanent damage to the physical process under control and/or the people depending on them. As a result, security of CPS applications has been the topic of a number of studies and references therein [1, 2, 3], including defining possible vulnerabilities in a specific CPS, that could be exploited by an attacker, analyzing the consequences of potential attacks, and suggesting countermeasures against attacks if possible.

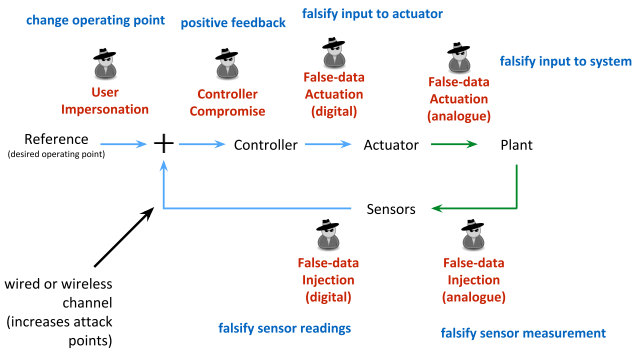


Fig. 1.1. Possible threats on cyber physical systems.

Fig. 1.1 shows possible vulnerabilities of a typical CPS that could be exploited in order to disturb the normal operation of CPS and generate harmful impacts. We can see that a physical process (plant) utilizes a controller to regulate, or track, a predefined reference (operation) point. The controller receives the reference point and current measurements from the sensors, processes this data according to the employed control technique, and then generates control commands to the actuators. Each one of those components can be compromised (attacked), resulting in different consequences for the CPS [3]. The impacts of attacking the sensing functionality of vehicular platoons and HVAC systems will be considered in this dissertation.

1.1 CPS1: Vehicular platoons

Vehicular platooning is an automation technology wherein a number of vehicles are grouped together to follow each other closely and safely without human intervention. This technology has been shown to provide a safe and comfortable experience that ultimately allows passengers to focus on tasks other than driving [4, 5]. Platooning also enables

vehicles to safely navigate at a closer distance than is possible with human-driven vehicles, thereby improving traffic throughput and reducing congestion [6, 7]. Additionally, studies have shown that platooning can help improve fuel consumption [8]. In general, a vehicular platoon includes a leading vehicle (leader) responsible for following a specified trajectory to the destination and setting the speed by which the whole platoon travels, and following vehicles (followers) which share the same control strategy that describes how they react to changes in the leader's behavior (e.g. the leader accelerates or decelerates).

1.1.1 Control of vehicular platoons: a survey

The idea of organizing a number of vehicles in platoons was studied by a number of research groups [9]. The platooning objective is to combine multiple vehicles and design the proper controllers to regulate and maintain a desired separation and speed. A large body of literature already exists addressing how to achieve that objective for homogeneous platoons, in which every vehicle uses the same control law [10]. Moreover, various spacing policies which define the desired separation are proposed in order to implement control laws that regulate the relative spacing of either the front of vehicle (unidirectional control) or both the front and rear of the vehicle (bidirectional control) [10]. This is achieved with solely locally-sensed information [11] or the addition of (V2V) communication [12]. The work of [13] proposes appropriate communication schemes to transmit messages between adjacent vehicles and also suggests a protocol that helps with the process of combining two platoons. Additionally, [14] shows that it is feasible to establish vehicle-to-infrastructure (V2I) communication in order to exchange vehicles' information with road units designed for that purpose.

Vehicular platooning is an example of CPS as implementing such a system requires communication, computation, and sensing capabilities in order to maintain predefined inter-vehicle separation and relative speed among the platooned vehicles. Adaptive Cruise Control (ACC) and Cooperative Adaptive Cruise Control (CACC) are the most well-known longitudinal control strategies used to form and maintain platoons, by implementing the selected spacing policy. ACC operation requires locally available information, specifically

the range (relative spacing) and range-rate (relative speed) gathered from sensors available to the vehicle (e.g., RADAR, LIDAR, or cameras), to generate the appropriate acceleration commands needed to maintain a preset inter-vehicle separation and speed. CACC, on the other hand, is an extension of ACC which incorporates vehicle-to-vehicle (V2V) communication, so that vehicles may exchange state information and intentions (e.g., alerting other vehicles to changes in acceleration), and is thereby able to achieve smaller inter-vehicle separations [10].

1.1.2 Security of vehicular platoons: a survey

Vehicular platoon security has been the focus of extensive research in literature. Many of the presented attacks are either insider attacks, in which one or multiple vehicle in the platoon are compromised to facilitate implementing the attack, or outsider attacks, wherein certain automation system components/functionalities are targeted from outside the platoon. In [15], the authors use CACC technique to form the platoon and then they present a number of insider attacks that target the vehicles' controllers and could disturb the formation. One of these attacks, for example, induces collisions at high speeds by exploiting the CACC structure through sending false information to the following vehicles. The authors also suggest a detection scheme for such attacks, based on requiring each platooned vehicle to model expected behavior of the preceding vehicles and compare that behavior with observed behavior. Such a scheme could lead to detecting abnormalities. Further insider attack work is presented in [16], wherein platooned vehicles use ACC with a bidirectional control algorithm to form a platoon. The attacker is then able to control one of the vehicles and modify its controller's gains such that generated acceleration commands modify the behavior of the attacked vehicle and induce instability in the entire platoon. Automated vehicle operating with the presence of an insider attacker is also discussed in [17]. In this work, the efficiency of the platoon is degraded when a malicious vehicle causes the surrounding vehicles to increase energy consumption unnecessarily. This is achieved by implementing an optimally calculated sequence of accelerating and decelerating commands by the attacker controlled vehicle.

In [18], the authors employ ACC and CACC control schemes to show that multiple attacker vehicles can operate within the platoon, modify their controllers, and coordinate their behavior in order to produce instability in a large traffic of automated vehicles. These attacks were able to induce traffic jams, passengers' discomfort, and an increased risk of collisions.

The studies conducted in [19] and [20] present how it is possible to detect and mitigate insider attacks mounted against vehicular platoons. In the former study, the authors consider an ACC with a bidirectional control-based platoon, wherein an attacker has the same capabilities as is given in [16]. The authors propose a low-pass filter detection scheme combined with a sliding-mode control based mitigation strategy in order to handle the attacker controlled vehicle misbehavior and reduce the risk of accidents. In the latter study [20], the authors consider a CACC with a unidirectional control-based platoon, wherein the insider attacker can cause Denial-of-Service (DoS) by broadcasting legitimate messages at a higher rate when compared to other platooned vehicles, in order to saturate the inter-vehicle communication channels. Furthermore, the authors suggest a detection strategy based on using a system node, or a low powered computer that does not need high computational capabilities.

On the other hand, there are other works which investigate external attacks on vehicular platoons wherein the target is local range and range-rate sensors or inter-vehicle-communication channels. The authors of [13] attempt to gain an understanding of the possible impacts of an outsider attack on a CACC-based platoon. In their study, a number of attacks against inter-vehicle communication channels are defined, including a falsification attack, wherein the attacker alters the contents of the broadcast messages used to implement the CACC control law, a spoofing attack, wherein the attacker pretends to be a vehicle in the platoon and sends inaccurate messages to other vehicles in the platoon, and a replay attack, wherein the attacker receives a transmitted message at a certain time and sends it back to its original destination at another time, thus creating a chance for hazardous effects, as the message contains old information. Furthermore, this study also

considers attacks against the vehicle hardware at the manufacturing level such as tampering with or installing a faulted sensor, which would lead to feeding incorrect information to the controller. The simulation results presented in [13] show that platoon stability can be affected, and as a result, the passenger safety may be compromised.

Another work which considers an external attack on a CACC-based vehicular platoon is presented in [21]. In this study, a wireless inter-vehicle channel suffers a jamming attack, wherein data transmission is severed between various vehicles in the platoon, launched from a drone flying above. The considered jamming attacks aim to disturb the string stability of the platoon, a characteristic which ensures that relative spacing errors attenuate along the platoon. In their study, the authors defined the best location for launching a jamming attack that would result in the highest spacing error propagation. Simulation results for that study show that attacker success produces string instability for the platoon by jamming communications between the lead and follower vehicles in the platoon.

The authors of [22] investigated the effects of attacking CACC based vehicular platoons with jamming and false data injection attacks. For this study, they used three of the CACC existing controllers to implement platoons. In order to generate the right commands, the first controller used constant distance for its spacing policy, the second used states estimation to predict the acceleration of the preceding vehicle if it were not received, and the third controller used state, position and speed, and information of the preceding and lead vehicles of the platoon. The reason for using three different controllers is to quantify their performance in the presence of the aforementioned attacks. According to the results presented in this study, all tested CACC controllers are unreliable when subjected to the above attacks; however, there are differences in terms of how each controller is affected by said attacks.

In [23], another study was conducted on a CACC-based vehicular platoon, wherein it was possible to manipulate the measurements provided by local RADAR, a sensor which measures the range-rate, and LIDAR, a sensor which measures the range, equipped on each of the platooned vehicles. In addition, this work also considers a case wherein an attacker-

controlled vehicle within the platoon reports false acceleration data to the following vehicle, information which is needed for CACC operation of the following vehicle, through the inter-vehicle communication. Although detection was not considered, the authors of this work have proposed two mitigation schemes: the first relies on the physical properties of the platoon, whereby newly-obtained data are compared with previously-obtained data and in the case that a certain threshold is violated, an attack may be in progress. The second scheme is based on building a behavioral model over time by using gathered data and then comparing that model with observed behavior. According to the given simulation results, the second scheme has shown better performance in terms of the detection rate of all the mounted attacks.

1.2 CPS2: HVAC systems

A smart building is a general term used to describe any building or a structure that utilizes an automation system to supervise and control important functionalities/subsystems, such as security, fire and flood safety, lighting, heating, cooling, and ventilating of the building. Designing a smart building, or even upgrading an old building to a smart one, requires installing sensors and actuators on the building's subsystems, such as fire alarms, water pumps, doors, and/or heating/cooling units. In addition, dedicated controllers are implemented to collect and analyze data from sensors and generate the appropriate commands to the actuators of each subsystem.

Smart buildings provide two major benefits: First, such buildings are characterized by an efficient use of energy, since the subsystems' controllers can provide optimal control, rather simple (classical) on/off control. Second, such buildings increase the comfort level of their occupants, which, in turn, may lead to more productivity within corporate offices. Heating, Ventilating, and Air Conditioning (HVAC) systems are one of the control systems implemented in smart buildings. Such systems can provide thermal comfort in residential or commercial smart buildings, while at the same time consuming less energy. For the purpose of designing an appropriate controller that achieves certain goals, detailed information about the heat dynamics of the building under consideration is needed. Acquiring an accurate

building model is helpful in the decision-making process of the controller, especially when the control strategy is highly dependent on the model of the process under control.

1.2.1 Control of HVAC: a survey

Optimal control of an HVAC system usually involves formulating multi-variable complex optimization problems. For that reason, most control algorithms employed for HVAC systems are simply on/off controllers. However, optimal control algorithms were also suggested for the HVAC system as such algorithms have shown the ability to reduce energy consumption. The authors of [24, 25] have developed a thermal model with the purpose of designing an optimal controller for an HVAC system within a building. In their model, the building is divided into a number of thermal zones, each of which could consist of single or multiple rooms, and within which, each zone is assumed to have sensors installed to measure the current temperatures. Then, the authors propose a hierarchical control algorithm which consists of two levels. The high level receives the current measurements and desired temperature, as set by occupants of each zone in the building. The high level uses LQR control technique to solve an optimization problem aiming to minimize energy consumption and improve the comfort level of the occupants. The outputs of the high level are the optimally-required amounts of air mass flow for each zone. The low level is simply a number of PID controllers for each zone, each of which implement air mass flow as calculated by the LQR.

Model Predictive Control (MPC) is a promising control strategy capable of operating a building's HVAC system in an optimal way while also satisfying state and input constraints, such as room air temperature and air mass flow rate. In [26], the authors used a grey-box approach to develop a nonlinear thermal model of a building. Then, the authors estimated the model parameters and validated the resulting model using recorded historic data (based on buildings on campus where the authors work). Next, the model was linearized around an operating point and then descritized, for the purpose of MPC controller design. The suggested MPC controller ensured the minimization of total energy (air mass flow) consumption and utilized the linearized model for future prediction. Performance of

the suggested MPC controller was compared with that of a simple on/off controller and the results showed a reduction in the air mass flow rate (input) throughout the day. Due to the weather conditions of the building considered, the designed MPC was intended only for heating. Another work considered also used MPC to reduce energy consumption but in this case, it was utilized for the purpose of chilling (cooling) a building with a linear thermal model [27, 28].

1.2.2 Security of HVAC: a survey

The authors of [29] defined possible vulnerabilities in the automation systems that employ HVAC technology. Such vulnerabilities include the attacker's capability to gain physical access either to the controllers, by guessing the correct password and shutting down the whole system, or to the interconnection between the HVAC's critical components, such as actuators, and, as a result, generating negative impacts. To counter such vulnerabilities, the authors suggested a neural network-based intrusion detection mechanism. Another vulnerability in HVAC systems defined by the authors of [30] is inaccurate measurement in the temperature or air flow rate sensors. By exploiting such vulnerability, the targeted sensors could produce measurements that are either below (negative bias) or above (positive bias) the real value of the measured quantities. Similarly, a wavelet neural work was also suggested and trained in order to diagnose faulty sensor(s).

In [31], several threats against HVAC systems were defined, including manipulating the set points, feeding a sensor either a constant false measurement or varying measurements that still fit within the bounds of the sensor measurements, or sending harmful commands to the actuators. In the same work, a system model-based detection method was also suggested. Similar to the above mentioned works, in this chapter we will focus on potential attacks against the temperature sensors of HVAC systems that result in incorrect measurements. Furthermore, we also suggest countermeasures to reduce the impacts generated by such attacks.

1.3 Analysis of threats against CPSs

As mentioned earlier, a number of attacks have been defined as possible threats against CPSs. One such attack is false-data injection. We will define this attack and discuss related previous works that make mention of this particular strategy. Furthermore, we will describe the reachability analysis utilized in this dissertation to quantify the impacts of the aforementioned attack.

1.3.1 False-data injection attacks

False Data Injection (FDI) attacks are carried out by an adversary with the capability to access and manipulate measurements provided by one or a set of a CPS sensors and consequently cause misbehavior in the decision-making process of the CPS, ultimately compromising the operation of the controller. Compared with DoS, FDI attacks are designed carefully such that their detection becomes more difficult as data is still available from the sensors, just the correct data. Furthermore, this false data is not determined randomly, as this data is predetermined to achieve certain attack goals. FDI attacks have been analyzed in literature with the goal of defining them for various examples of CPSs and also providing the conditions and guarantees necessary for successful FDI attacks.

The authors of [32] studied the effects of FDI attacks on a subset of sensors equipped for a linear system, a system assumed to have a state estimator. Their work assumes that the attacker has a full knowledge of the system and controller dynamics. This study explains the necessary and sufficient conditions by which the FDI attack cannot be easily detected. The analysis conducted in [33] focuses on the FDI attacks against the state estimation process in electrical power grids. In this work, the attacker was assumed aware of the configuration of the attacked power grid, which made it possible to design two attack scenarios that easily passed bad measurements detectors, as these are usually installed in power systems, and thereby produce false state estimation that could destabilize the grid.

As mentioned earlier, a platooning CPS controller requires measurements from on-board sensors for its operation. Existing work has demonstrated that most on-board sensors in automated vehicles, such as LIDAR or cameras, are vulnerable to FDI attacks executed

at a distance. For example, [34] presents external jamming and spoofing attacks that can be carried out against ultrasonic sensors and cameras, and experimental results even showed a possibility of malfunctioning a Tesla vehicle. Also, it was proven possible to falsify the readings of a vehicle's RADAR [35], LIDAR, and/or cameras [36], and, as a result, disrupt the behavior of the automated vehicle. Therefore, we will formulate an FDI attack that could be mounted against one of the vehicular sensors in order to show the possible harmful impacts that could result.

1.3.2 Reachability analysis

Reachability analysis defines the reachable set of a dynamic system, or the set of all system states that can be attained within a finite time. Considering the physical bounds and performance constraints of system states and inputs, reachability analysis helps us verify whether from a given initial point, a system can eventually reach a given final point. For this purpose, reachability can be applied in real world applications where safety must be determined, such as collision avoidance problems in airplanes [37] and vehicles [38], or controller design for the platooning of unmanned aerial vehicles (UAV) [39, 40].

Various methods have been proposed for obtaining reachable sets for different classes of systems. Some of these methods are based on ellipsoidal techniques [41, 42] which calculates outer elliptical bounds around the reachable set of a linear system with physical bounds on its input vector. This method has been applied in problems such as determining algorithms for collision avoidance in UAVs [43], or determining new artificial physical bounds for a system's actuators (inputs), in order to restrict the states that can be reached and hence limit the impacts of potential attacks on that system [44]. Other methods, generally known as Hamilton-Jacobi (HJ) reachability, are based on finding the solution of a Hamilton-Jacobi-Bellman partial differential equation [37, 45]. HJ reachability has been used to solve various problems, such as auto landing of an aircraft [46], the interaction of two air-crafts for automated aerial refueling [47], and path planning for UAVs [48]. Finally, another method suggested for determining the reachable set is based on optimal control theory, wherein the final states' points are included in the formulation of an optimization problem. This

problem, in turn, calculates the appropriate control required to drive the system states toward those final states, while also considering states and input constraints [49]. This method has been applied in problems such as determining a safe landing area for a moon lander [50] and suggesting an alternate trajectory for vehicles, in order to make tracking possible and avoid collision with other vehicles [51].

1.4 Mitigation of attacks against CPSs: Moving Target Defense

Moving Target Defense (MTD) has been suggested as a countermeasure that aims to decrease the attacker's ability to influence the attacked CPS. The MTD mechanism utilizes a switching structure in order to alter the behavior of the CPS in terms of its actuators or sensors. As a result, the MTD mechanism is proactive in nature and could preemptively guarantee that most attacks would fail to induce harmful impacts. Previous work has shown that MTD has been employed in computer security [52]. For example, the authors of [53] proposed an MTD algorithm to protect the privacy of Internet Protocol version 6 users. Their algorithm repeatedly changed the addresses of both the sender and receiver, such that the attacker was prevented from identifying the two communication hosts. Similarly, the authors of [54] developed an MTD algorithm for mutating the IP address, thus creating a high chance of unpredictability while maintaining the original configuration of the address.

MTD has also been used in the context of control theory. In [55], the authors implemented an MTD mechanism by introducing additional states related to the original states of the control system, each with time-varying dynamics. While the new states are known to the control system operator, they remain hidden from the attacker, and because their dynamics change constantly, the attacker cannot identify them, and hence, the attack influence is deterred. Also, the authors of [56] formulated a zeros-sum game theoretic framework to aid with designing an MTD strategy for a vulnerable system. They also developed a feedback mechanism that would allow the system to monitor its states and decide whether to add stochastic dynamics as a part of the suggested MTD, such that the attack surface of the system would be decreased.

1.5 Organization

In Chapter 2, similar to the security-related works discussed in Section 1.1.2, we present a possible vulnerability in vehicular platoons and analyze its impacts on platoon safety. However, ours is the first work that considers the effect that the presence of human control in the platoon can produce. Specifically, we try to answer the following: "What happens if control of multiple vehicles transitions to humans, due to disruption of the automated systems?" or "What happens if a passenger assumes command of a vehicle after observing irregular motion behavior, owing to an already mounted attack?". Naturally, once a human driver controls the vehicle, they will first apply brakes in an attempt to slow down the vehicle [57]. While such an action is helpful in avoiding accidents, it may also generate instability in the following non-attacked platoons and lead to collisions.

In Chapter 3, we are concerned with the safety of a vehicular platoon operating in an adversarial setting where it is possible to target one, or more, of the platoon's vehicle's sensors with an FDI attack. Particularly, we are interested in defining the set of final states that the platoon can reach as a result of experiencing a manipulation in the measurements obtained from one or more of the locally-equipped sensors. For that purpose, we will use the optimal control-based reachability approach to determine the reachable (final) set of states, since it allows us to include attacker's capabilities, physical limits on the vehicle's acceleration and speed, and resolution and physical limits of the attacked sensor(s) in the problem formulation as constraints. Furthermore, this approach requires a prior definition of the final states of interest. For that reason, and because we are primarily concerned with the safety of the platoon, we will focus on unsafe states, which can be translated collisions between two or more vehicles in the platoon and at different speeds of impact. Regardless of the type of equipped sensors, from this point on we will refer to the sensors measuring relative distance and speed as range and range-rate sensors, respectively.

In Chapter 4, we utilize MPC technique to formulate an optimal controller that aims to achieve an acceptable temperature tracking, of a desired set-point, in each zone of the building. To develop such a controller, a model of the process under consideration (the

smart building) is needed. For that purpose, we employ a thermal model which captures heat storage and transfer between connected spaces of the building, as well as the influence of outside temperature. On the other hand, we consider a possible threat against the HVAC system by manipulating the measurements of the temperature sensors installed at various sections of HVAC-equipped buildings. An important factor that facilitates such attacks against temperature sensors is the fact that the MPC controller, which uses those sensors, is static in nature and thus, attackers can easily induce predictable impacts. Therefore, in this chapter we suggest MTD technique-based algorithms which aim to add unpredictability to the system by constantly changing the sensor sets used by the MPC controllers and thereby reduce the impacts of potential attacks.

In Chapter 5, we study the behavior of a single vehicular platoon where one of the platooned vehicles is controlled by an attacker. The latter is able to modify the platooning controller of the seized vehicle and, hence, produce sudden accelerating/decelerating movements that can lead to collisions within the platoon. A previous study suggested a sliding mode controller which uses only local vehicular sensor information without the need for inter-vehicle communications, to mitigate the impacts of the aforementioned attack. The suggested controller is also assisted with decentralized attack detection. Simulation results from that study demonstrate that collisions are eliminated, or significantly reduced in certain cases. However, the same results also indicate that the intact vehicles concede platooning and start following the attacker. For instance, the lead vehicle, even if not attacked, will no longer follow the reference trajectory of its platooning goals, once it detects an attack in the following vehicles. Therefore, we will modify the suggested mitigation controller such that collisions are also reduced and the control of intact vehicles will eventually switch from auto to human, thereby disbanding the platoon so the attacker can have no more influence.

REFERENCES

[1] A. A. Cardenas, S. Amin, and S. Sastry, "Secure control: Towards survivable cyber-physical systems," in *2008 The 28th International Conference on Distributed Computing Systems Workshops*, June 2008, pp. 495–500.

[2] A. Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, and S. Sastry, "Challenges for securing cyber physical systems," in *Workshop on Future Directions in Cyber-physical Systems Security*. DHS, July 2009. [Online]. Available: http://chess.eecs.berkeley.edu/pubs/601.html

[3] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—a survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, Dec 2017.

[4] E. Coelingh and S. Solyom, "All aboard the robotic road train," *IEEE Spectrum*, vol. 49, pp. 34–49, 2012.

[5] L. Alvarez and R. Horowitz, "Traffic flow control in automated highway systems," California PATH Research Report, 1997. [Online]. Available: www.sartre-project.net

[6] J. Carbaugh, D. N. Godbole, and R. Sengupta, "Safety and capacity analysis of automated and manual highway systems," *Transportation Research Part C: Emerging Technologies*, vol. 6, no. 1, pp. 69 – 99, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X98000096

[7] W. Ren and D. Green, "Continuous platooning: a new evolutionary operating concept for automated highway systems," in *American Control Conference (ACC), 1994*, June 1994.

[8] K.-Y. Liang, J. Mårtensson, and K. H. Johansson, "Fuel-saving potentials of platooning evaluated through sparse heavy-duty vehicle position data," *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 1061–1068, 2014.

[9] C. Bergenhem, S. Shladover, E. Coelingh, C. Englund, and S. Tsugawa, "Overview of platooning systems," in *Proceedings of the 19th ITS World Congress, Oct 22-26, Vienna, Austria (2012)*, 2012.

[10] R. Rajamani, *Vehicle Dynamics and Control*, ser. Mechanical Engineering Series. Springer, 2011. [Online]. Available: https://books.google.com/books?id=eoy19aWAjBgC

[11] D. Yanakiev and I. Kanellakopoulos, "A simplified framework for string stability analysis in ahs," in *Proceedings of the 13th IFAC World Congress, 1996*, 1996, pp. 177–182.

[12] S. Oncu, J. Ploeg, N. van de Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Network-aware analysis of string stability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1527–1537, Aug 2014.

[13] M. Amoozadeh, A. Raghuramu, C. Chuah, D. Ghosal, H. M. Zhang, J. Rowe, and K. Levitt, "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 126–132, June 2015.

[14] C. Chou, C. Li, W. Chien, and K. Lan, "A feasibility study on vehicle-to-infrastructure communication: Wifi vs. wimax," in *2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, May 2009, pp. 397–398.

[15] B. DeBruhl, S. Weerakkody, B. Sinopoli, and P. Tague, "Is your commute driving you crazy?: a study of misbehavior in vehicular platoons," in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 2015, p. 22.

[16] S. Dadras, R. M. Gerdes, and R. Sharma, "Vehicular platooning in an adversarial environment," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, ser. ASIA CCS '15. New York, NY, USA: ACM, 2015, pp. 167–178. [Online]. Available: http://doi.acm.org/10.1145/2714576.2714619

[17] R. M. Gerdes, C. Winstead, and K. Heaslip, "Cps: an efficiency-motivated attack against autonomous vehicular transportation," in *Proceedings of the 29th Annual Computer Security Applications Conference*. ACM, 2013, pp. 99–108.

[18] D. D. Dunn, S. A. Mitchell, I. Sajjad, R. M. Gerdes, R. Sharma, and M. Li, "Regular: Attacker-induced traffic flow instability in a stream of semi-automated vehicles," in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, June 2017, pp. 499–510.

[19] I. Sajjad, D. D. Dunn, R. Sharma, and R. Gerdes, "Attack mitigation in adversarial platooning using detection-based sliding mode control," in *Proceedings of the First ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy*, ser. CPS-SPC '15. New York, NY, USA: ACM, 2015, pp. 43–53. [Online]. Available: http://doi.acm.org/10.1145/2808705.2808713

[20] M. Kaur, A. Rayamajhi, M. Rahman, J. J. Martin, M. Chowdhury, and G. Comert, "Towards secure infrastructure-based cooperative adaptive cruise control," *CoRR*, vol. abs/1809.05119, 2018.

[21] A. Alipour-Fanid, M. Dabaghchian, H. Zhang, and K. Zeng, "String stability analysis of cooperative adaptive cruise control under jamming attacks," in *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, Jan 2017, pp. 157–162.

[22] R. van der Heijden, T. Lukaseder, and F. Kargl, "Analyzing attacks on cooperative adaptive cruise control (cacc)," in *2017 IEEE Vehicular Networking Conference (VNC)*, Nov 2017, pp. 45–52.

[23] M. Jagielski, N. Jones, C.-W. Lin, C. Nita-Rotaru, and S. Shiraishi, "Threat detection for collaborative adaptive cruise control in connected cars," in *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, ser. WiSec '18. New York, NY, USA: ACM, 2018, pp. 184–189. [Online]. Available: http://doi.acm.org/10.1145/3212480.3212492

[24] M. Maasoumy Haghighi, "Modeling and optimal control algorithm design for hvac systems in energy efficient buildings," Master's thesis, EECS Department, University of California, Berkeley, Feb 2011. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2011/EECS-2011-12.html

[25] M. Maasoumy, A. Pinto, and A. Sangiovanni-Vincentelli, "Model-based hierarchical optimal control design for hvac systems," *ASME 2011 Dynamic Systems and Control Conference and Bath/ASME Symposium on Fluid Power and Motion Control, DSCC 2011*, vol. 1, 01 2011.

[26] M. Maasoumy and A. Sangiovanni-Vincentelli, "Total and peak energy consumption minimization of building hvac systems using model predictive control," *IEEE Design Test of Computers*, vol. 29, no. 4, pp. 26–35, Aug 2012.

[27] Y. Ma, F. Borrelli, B. Hencey, A. Packard, and S. Bortoff, "Model predictive control of thermal energy storage in building cooling systems," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, Dec 2009, pp. 392–397.

[28] Y. Ma, F. Borrelli, B. Hencey, B. Coffey, S. Bengea, and P. Haves, "Model predictive control for the operation of building cooling systems," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 796–803, May 2012.

[29] C. B. Jones and C. Carter, "Trusted interconnections between a centralized controller and commercial building hvac systems for reliable demand response," *IEEE Access*, vol. 5, pp. 11 063–11 073, 2017.

[30] Z. Du, X. Jin, and Y. Yang, "Fault diagnosis for temperature, flow rate and pressure sensors in vav systems using wavelet neural network," *Applied Energy*, vol. 86, no. 9, pp. 1624 – 1631, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261909000233

[31] D. C. Wardell, R. F. Mills, G. L. Peterson, and M. E. Oxley, "A method for revealing and addressing security vulnerabilities in cyber-physical systems by modeling malicious agent interactions with formal verification," *Procedia Computer Science*, vol. 95, pp. 24 – 31, 2016, complex Adaptive Systems Los Angeles, CA November 2-4, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050916324619

[32] Y. Mo and B. Sinopoli, "False data injection attacks in control systems," *Preprints of the 1st Workshop on Secure Control Systems*, 01 2010.

[33] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 21–32. [Online]. Available: http://doi.acm.org/10.1145/1653662.1653666

[34] C. Yan, W. Xu, and J. Liu, "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle," *DEF CON*, vol. 24, 2016.

[35] R. Chauhan, R. M. Gerdes, and K. Heaslip, "Attack against an fmcw radar," in *Proceedings of Embedded Security in Cars Conference*, 2014.

[36] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," *Black Hat Europe*, vol. 11, p. 2015, 2015.

[37] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, July 2005.

[38] I. Xausa, R. Baier, M. Gerdts, M. Gonter, and C. Wegwerth, "Avoidance trajectories for driver assistance systems via solvers for optimal control problems," in *20th International Symposium on Mathematical Theory of Networks and Systems*, Melbourne, Australia, 2012, cD-ROM, Paper No. 294, 8 pages,. [Online]. Available: https://hal.inria.fr/hal-00712878

[39] M. Chen, Q. Hu, C. Mackin, J. F. Fisac, and C. J. Tomlin, "Safe platooning of unmanned aerial vehicles via reachability," *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 4695–4701, 2015.

[40] M. Chen, Q. Hu, J. F. Fisac, K. Akametalu, C. Mackin, and C. J. Tomlin, "Reachability-based safety and goal satisfaction of unmanned aerial platoons on air highways," 2016.

[41] A. Kurzhanski and P. Varaiya, "On ellipsoidal techniques for reachability analysis. part i: External approximations," *Optimization Methods and Software*, vol. 17, no. 2, pp. 177–206, 2002.

[42] A. A. Kurzhanskiy and P. Varaiya, "Ellipsoidal techniques for reachability analysis of discrete-time linear systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 1, pp. 26–38, Jan 2007.

[43] Y. Zhou and J. S. Baras, "Reachable set approach to collision avoidance for uavs," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 5947–5952.

[44] S. H. Kafash, J. Giraldo, C. Murguia, A. A. Cardenas, and J. Ruths, "Constraining attacker capabilities through actuator saturation," in *2018 Annual American Control Conference (ACC)*, June 2018, pp. 986–991.

[45] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-jacobi reachability: A brief overview and recent advances," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec 2017, pp. 2242–2253.

[46] A. M. Bayen, M. Mitchell, M. M. K. Oishi, and C. J. Tomlin, "Aircraft autolander safety analysis through optimal control-based reach set computation," 2006.

[47] J. Ding, J. Sprinkle, S. S. Sastry, and C. J. Tomlin, "Reachability calculations for automated aerial refueling," in *2008 47th IEEE Conference on Decision and Control*, Dec 2008, pp. 3706–3712.

[48] M. Chen, S. Bansal, J. F. Fisac, and C. J. Tomlin, "Robust sequential trajectory planning under disturbances and adversarial intruder," *IEEE Transactions on Control Systems Technology*, pp. 1–17, 2018.

[49] R. Baier and M. Gerdts, "A computational method for non-convex reachable sets using optimal control," in *2009 European Control Conference (ECC)*, Aug 2009, pp. 97–102.

[50] Y. E. Arslantaş, T. Oehlschlägel, and M. Sagliano, "Safe landing area determination for a moon lander by reachability analysis," *Acta Astronautica*, vol. 128, pp. 607 – 615, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094576516307846

[51] M. Gerdts and I. Xausa, "Avoidance trajectories using reachable sets and parametric sensitivity analysis," in *System Modeling and Optimization*, D. Hömberg and F. Tröltzsch, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 491–500.

[52] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*, 1st ed. Springer Publishing Company, Incorporated, 2011.

[53] M. Dunlop, S. Groat, W. Urbanski, R. Marchany, and J. Tront, "Mt6d: A moving target ipv6 defense," in *2011 - MILCOM 2011 Military Communications Conference*, Nov 2011, pp. 1321–1326.

[54] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "Openflow random host mutation: Transparent moving target defense using software defined networking," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN '12. New York, NY, USA: ACM, 2012, pp. 127–132. [Online]. Available: http://doi.acm.org/10.1145/2342441.2342467

[55] S. Weerakkody and B. Sinopoli, "Detecting integrity attacks on control systems using a moving target approach," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 5820–5826.

[56] Q. Zhu and T. Başar, "Game-theoretic approach to feedback-driven multi-stage moving target defense," in *4th International Conference on Decision and Game Theory for Security - Volume 8252*, ser. GameSec 2013. New York, NY, USA: Springer-Verlag New York, Inc., 2013, pp. 246–263. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-02786-9_15

[57] R. Zheng, K. Nakano, S. Yamabe, M. Aki, H. Nakamura, and Y. Suda, "Study on emergency-avoidance braking for the automatic platooning of trucks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1748–1757, Aug 2014.

CHAPTER 2

THE DISBANDING ATTACK: EXPLOITING HUMAN-IN-THE-LOOP CONTROL IN VEHICULAR PLATOONING

Due to advances in automated vehicle technology and inter-vehicle communication, vehicular platoons have attracted a growing interest by academia and industry alike, as they can produce safe driving, regularize traffic flow, and increase throughput. Research has demonstrated, however, that when platoons are placed in an adversarial environment they are vulnerable to a variety of attacks that could negatively impact traffic flow and produce collisions and/or injuries. In this chapter, we consider an attack that seeks to exploit human-in-the-loop control of compromised vehicles that are part of a platoon. Specifically, we demonstrate that should a human operator need to suddenly take control of a platooned vehicle significant upstream effects, which threaten the safety of passengers in other vehicles, may be induced. To counter this so-called disbanding attack, we present an optimal centralized mitigation approach. Due to scalability, security, and privacy concerns such an approach may not be practical in reality so we propose a decentralized mitigation algorithm that reduces excessive speed changes and coordinates inter-platoon behaviors to minimize the impact of the attack. Our algorithm is compared to the aforementioned optimal centralized approach and is shown to produce nearly equivalent results while requiring far fewer resources. Experimental results on a hardware testbed show that our countermeasure permits graceful speed reductions and can avoid collisions.

2.1 Introduction

Vehicular platooning is an automation technology wherein a number of vehicles are grouped together to follow each other closely and safely. This technology has been shown to provide a safe and comfortable experience that will ultimately allow passengers to focus on tasks other than driving [1]. It also enables vehicles to safely navigate at a closer dis-

tance than it is possible with human-driven vehicles, thereby improving traffic throughput and reducing congestion [2], as well as helping to improve fuel consumption [3]. Vehicle platooning is an example of a cyber-physical system (CPS), as it requires an integration of computation, communication, and monitoring capabilities to control a physical process. Adaptive Cruise Control (ACC) and Cooperative Adaptive Cruise Control (CACC) are the most well-known control strategies used to form and maintain platoons. ACC operation relies on locally-available information to generate appropriate acceleration commands in order to maintain a preset inter-vehicle separation and speed (longitudinal control). CACC, on the other hand, is an extension of ACC that employs vehicle-to-vehicle (V2V) communication, so that vehicles may exchange state information and intentions (e.g., alerting other vehicles to changes in acceleration), and is able to achieve smaller inter-vehicle separations [4].

The Society of Automotive Engineers (SAE) and the National Highway Traffic Safety Administration (NHTSA) have defined levels of vehicular automation. Based on their criteria, vehicle manufacturers have been able to produce vehicles at level 2 capabilities, including BMW, Ford, and General Motors, or level 3, such as Tesla [5]. In level 2, an automated vehicle is able to generate both longitudinal (accelerating/decelerating) and lateral (steering) control commands. This level also requires the human to monitor the road and retain readiness to assume control if needed. Level 3 provides more automated functionalities in terms of generating control commands and monitoring the driving environment; however, it also requires a human driver to be available to assume control [6]. Platooning without human oversight, a level 4 technology, is not yet a reality due to a lack of robustness in V2V communications, the cost and number of sensors required to monitor the environment, and unresolved questions regarding unexpected maneuvers on the part of other vehicles on the road [7]. As a result, the current platooning automation technology falls in the category of level 2 or level 3, and human attention is still required in the platooned vehicles in case humans need to take control.

Transition of control is defined as the process of switching control from the vehicle's

automation to a human driver for cases when the automated system cannot handle certain situations; e.g., a vehicle emerging from a side road abruptly and merging onto a highway without notice, oncoming traffic turning left to enter a side road and crossing an automated vehicle's path, a car parking on the road and partially blocking the roadway [5, 8], or a technical failure in one or more components of the vehicle's automation system [9]. Such failures could stem from deliberate manipulation of the automated system components such as sensors, actuators, or inter-vehicle communication [10]. A number of previous studies analyzed human driver behaviors post transition of control and results have shown that some drivers apply maximum deceleration to handle certain situations, e.g., avoiding collision with preceding vehicles [11, 12]. These studies also determined the time required to ensure a safe transition [8, 13].

A platooning CPS (typically) employs a distributed controller that uses information from both local sensors and other vehicles, information obtained through inter-vehicle communications or connections to external networks [14]. As a result, a platooning CPS has a large attack surface upon which an attacker could induce disruptive and/or fatal behaviors [15, 16, 17, 18, 19]. Attacks mounted against a platooning CPS can lead to the disruption of the steady-state operation (i.e., desired inter-vehicle separation and relative speed) and thereby produce harmful effects, such as collisions or uncomfortable acceleration/deceleration, which could, in turn, lead to further disruption, such as chronic traffic jams. Also, attacks on platooned vehicles could induce a transition of control which might disband (dissolve) the platoon, as the latter would no longer be automated nor in compliance with platooning control laws. While the security of platooning CPS has been studied from many perspectives, so far the exploitation of the human element has been left unexplored.

In this chapter, we examine, from an adversarial perspective, the after-effects of automated vehicles transitioning their control to humans. Particularly, we are interested in analyzing the upstream effects of all vehicles in a platoon transitioning control to human operators (a process we refer to as platoon disbanding) due to a system failure resulting from an attack. Although disbanding may seem a sensible fail-safe solution to prevent

attackers from achieving their objectives, we will show that transition of control can still be leveraged to undermine the operation of surrounding vehicles, cause collisions, and/or induce massive congestion. The main contributions of this chapter are:

- We study the effect of a "disbanding attack" that involves transition of control of multiple vehicles in a platoon. We illustrate the harmful impacts such an attack can induce, especially in the case of causing upstream (non-attacked) platoons to experience slowdowns and collisions.
- We define a disbanding attack by formulating it as an optimization problem wherein the objective is to maximize the deviation in vehicle speeds as a proxy for slowdowns and increased chances of colliding, by selecting both platoon(s) to be disbanded and time(s) of disbanding.
- To mitigate the aftermath of such an attack, we formulate an optimal solution using a Model Predictive Control (MPC) technique. However, as the optimal approach is not scalable in practice, as it is centralized and information and communication intensive, we also propose a heuristic algorithm to be used locally by vehicles of intact (non-disbanded) platoons. Our findings indicate that our algorithm produces nearly equivalent results in terms of reducing speed changes and avoiding accidents.
- We also demonstrate the validity of the above attack and the suggested heuristic countermeasures using experiments on a hardware testbed consisting of a motion capture system and small mobile robots acting as vehicles.

2.1.1 A motivating example

Let us consider a scenario wherein multiple vehicular platoons are traveling in the same direction on a highway. Although they may not be heading to the same destination, platoons still follow one another in order to reap platooning benefits of optimizing traffic flow and reducing congestion. Consider that while the platoons operate at a steady-state, a malicious party utilizes an existing external attack techniques [18, 20] in order to destabilize

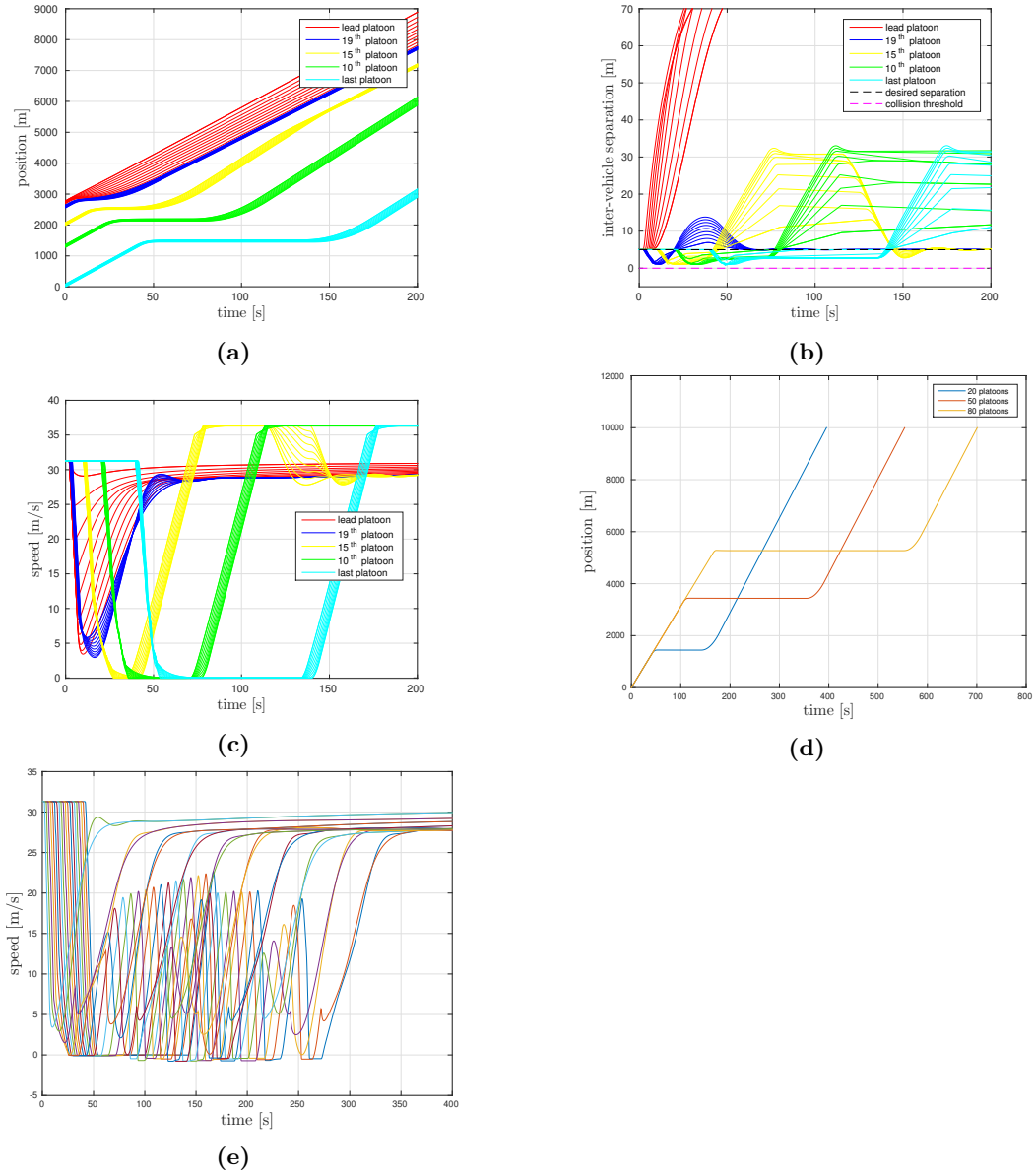


Fig. 2.1. a) Position profiles of the platoons shown in the legend. The lead platoon started disbanding at t = 2 s. b) Inter-vehicle separation profiles of the platoons shown in the legend. The lead platoon started disbanding at t = 2 s. c) Speed profiles of the platoons shown in the legend. The lead platoon started disbanding at t = 2 s. d) Position profile of the rear vehicle in the platoons formation, whose size is shown in the legend. e) Speed profiles of the rear vehicles belonging to a twenty-platoon formation when multiple platoons start disbanding at different time instances.

the formation. For example, the attacker might install units on the roadside that can jam the sensors of multiple vehicles or modify sensor measurements so that the targeted vehicles

start behaving irregularly [21]. At this point, either the automation system would suffer a failure and inform the driver, perhaps via sounding an auditory alarm [11], or the attack would be detected by either a mechanism designed for such a purpose or by a passenger who observes erratic behavior in the vehicle's motion. In any case, the driver must assume control of the vehicle and apply the brakes [9]. As a result, the attacked platoon would be effectively disbanded as the vehicles would no longer comply with platooning laws, and the mounted attack would fail to achieve its goals. However, intact upstream platoons, which were not the goal of the mounted external attack, would also exhibit unexpected behavior as a result of disbanding, which would create, at minimum, discomfort for passengers, or, at maximum, collision.

Fig. 2.1a shows the position profiles of selected platoons, out of 20, whose indices are shown in the legend. Each platoon includes 10 vehicles. The lead platoon (red), which constitutes 5% of the total number of vehicles, transitions its control after being attacked at t = 2 s. We can see that the lead platoon begins disbanding when the inter-vehicle separations, shown in Fig. 2.1b, are no longer 5 m (the desired separation) and the platoon manages to avoid accidents. Also, Fig. 2.1c indicates that the vehicles of the disbanded platoon initially slow down before speeding up. In response to the disbanding of the lead platoon, we can see in Fig. 2.1c that the following (still automated) 19^{th} intact platoon (blue) also begins to slow down. In addition, Fig. 2.1b shows that the inter-vehicle separation of the 19^{th} platoon is also affected as it decreases when slowing down happens, but not below 0, and then starts increasing to above 10 m when speeding up occurs, eventually reaching 5 m after almost 1 minute.

The same effect induced in the 19^{th} platoon will propagate throughout the remainder of the following platoons. For example, the 15^{th} platoon (yellow) began decelerating until all vehicles completely stopped, as shown in Fig. 2.1c, for almost 30 seconds. Then the platoon's lead vehicle began accelerating, reaching a maximum speed of 36 m/s in order to decrease the gap with respect to the preceding 16^{th} platoon (not shown in plots), before it eventually slowed down upon approach of the preceding platoon, after almost 2 minutes

(these actions of accelerating/decelerating result from the response of adopted automation control laws to the behavior of the preceding platoon). We can see the same behavior in Fig. 2.1b where the inter-vehicle separation of the 15^{th} platoon decreased, increased, and then settled at 5 m. The same pattern also shows on the 10^{th} (green) and last (cyan) platoons, but in these cases, longer times were needed to regain the inter-vehicle separations and speeds. For this specific disbanding attack, 10 minutes were needed in order for all of the affected platoons to re-establish (recover) the desired separations and speeds. Furthermore, Fig. 2.1d shows the absolute position of the last vehicle in the traffic stream for various platoons with the lead platoon disbanded. We can see that as the number of platoons increases, the vehicle stops for a longer time before resuming movement. Furthermore, the string of 20, 50, and 80 platoons needed 10, 25, and 43 minutes, respectively, to recover. In summary, we can see in these plots that disbanding one platoon could cause the following platoons to respond irregularly, such that they stop-and-go, which, in turn, creates discomfort for passengers, not to mention traffic jams, inefficient use of the road, and greater fuel-wasting.

Alternatively, when aware of such effects, an attacker could then target more than one platoon systematically and produce even worse impacts, such as multiple stop-and-go behaviors, which inevitably lead to passenger discomfort, greater fuel wasting, and increased collisions. For example, an attacker can induce disbanding by targeting every other platoon, out of twenty, at regular intervals, with 30 s increments (Fig. 2.1e). For the speed profiles shown in Fig. 2.1e, 65%, 45%, and 37.5% of the intact platoons were forced to stop-and-go once, twice, and three times, respectively. As a result, 55% of the vehicles, in the intact platoons, suffered collisions.

2.1.2 Related work

The objective of vehicular platooning is to combine multiple vehicles and design the proper controllers to maintain a desired separation and speed [22]. A large amount of literature addresses how to achieve that objective. Also, various spacing policies have been proposed for implementing control laws that regulate the relative spacing, either in front of the vehicle (unidirectional control), or on both the front and rear of the vehicle (bidirectional

control) [4]. This is achieved by using either locally-sensed information or with the addition of (V2V) communication [23]. Communication schemes have been proposed [24] to transmit messages between adjacent vehicles. In addition, it has been found that establishing vehicle-to-infrastructure (V2I) communication is feasible in order to exchange vehicle information with road units designed for that purpose [25]. In this work, we adopt a proportional-derivative controller from [23] to form our platoons with the presence of a forward-looking V2V communication in order to implement our suggested attack mitigation (Section 2.4).

Vehicular platoon security has been the focus of extensive research in literature. For example, [17] presents a number of insider attacks that target vehicles' CACC controllers. It suggests detection schemes for those attacks. Another insider attack work is [16], wherein the attacker's controlled vehicle is able to modify its controller's gains such that generated commands induce instability in the entire platoon. [15] shows that it is possible for a malicious vehicle in the platoon to increase energy consumption unnecessarily in neighboring vehicles by misbehaving. In [19], it is shown that multiple attacker vehicles can operate within the platoon and coordinate their behavior in order to produce instability that could lead to accidents. Alternately, other works investigate external attacks wherein local range and range-rate sensors are targeted to misinform the vehicle of the surrounding vehicles' information in order to negatively impact road efficiency and passenger comfort and safety [18, 20]. Similar to the security-related works above, we also present a possible vulnerability in vehicular platoons and analyze its impacts on platoon safety. However, ours is the first work that considers the effect the presence of human control in the platoon can produce. Specifically, we try to answer "what happens if control of multiple vehicles transition to humans because of disruption of their automated systems?" or "what happens if a passenger decides to assume command of a vehicle after observing irregular behavior in its motion, owing to an already mounted attack?". Naturally, once a human driver starts controlling the vehicle, brakes will be applied in an attempt to slow the vehicle [9]. While such an action is helpful in avoiding accidents, it will also generate instability in the following non-attacked platoons that could lead to collisions.

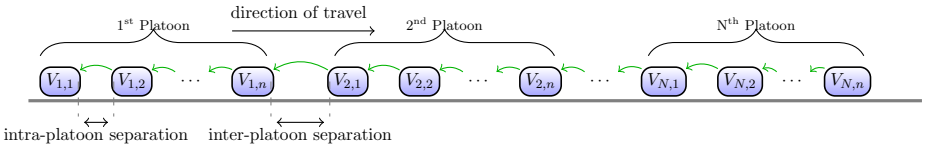


Fig. 2.2. A stream of n-vehicle N platoons. Green arrows represent the flow of transmitted information.

2.1.3 Organization

Section 5.5 explains the vehicular platooning control laws and describes the threat model. Section 2.3 discusses different optimal attack scenarios and analyze their impacts. Section 2.4 presents effective attack countermeasures. Experimental results are presented in Section 2.5. Conclusions are given in Section 2.6.

2.2 System Model

The modeling of platoon dynamics and control as well as the attack mechanism are discussed in this section.

2.2.1 Vehicle and platoon model

We consider N homogeneous platoons, where every vehicle uses the same control law, with n vehicles in each (lead vehicle is indexed as n while the last vehicle is indexed as 1) as shown in Fig. 2.2. Each vehicle is equipped with front and back range and range-rate sensors, to measure the corresponding relative distances and speeds, and implements an upper-level controller (responsible for determining the commanded (desired) acceleration) and a lower-level controller (uses the desired acceleration to determine throttle and brakes commands). The latter is expected to achieve the desired acceleration with some delay due to its finite bandwidth [4, 14]. We will focus on the upper-level controller since the attacker can easily affect it (e.g., through attacks on sensors). The following model is used

to simulate the dynamics of each j^{th} vehicle in the i^{th} platoon

$$\begin{bmatrix} \dot{x}_{i,j}(t) \\ \dot{v}_{i,j}(t) \\ \dot{a}_{i,j}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & \frac{-1}{\tau} \end{bmatrix} \begin{bmatrix} x_{i,j}(t) \\ v_{i,j}(t) \\ a_{i,j}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{bmatrix} u_{i,j}(t) \tag{2.1}$$

where x, v a, and u refer to the vehicle's absolute position, velocity, acceleration, and commanded acceleration, respectively, and τ is a time constant used to model the actuator's delay. In this work, vehicles in a platoon use a bidirectional control technique [23] which has two major benefits: First, it is able to guarantee platoon string stability, which maintains proper traffic flow [4, 23]. Second, it does not need any (V2V) transmitted information to generate control commands. However, such a wireless communication is established and will be used to inform vehicles of attack detection and to transmit data in the mitigation process (Section 2.4) though at a data rate far lower than that required to maintain V2V-enabled platoons. For the last vehicle in the i^{th} platoon, we have

$$u_{i,1}(t) = k_p\big(x_{i,2}(t) - x_{i,1}(t) - x_d\big) + k_d\big(v_{i,2}(t) - v_{i,1}(t)\big) \tag{2.2}$$

where k_p and k_d are the controller's proportional and derivative gains, respectively, and x_d is a constant denoting inter-vehicle desired separation. For the other vehicles in the i^{th} platoon, except the leader, we have

$$\begin{aligned} u_{i,j}(t) = k_p\big\{\big(x_{i,j+1}(t) - x_{i,j}(t) - x_d\big) - \big(x_{i,j}(t) - x_{i,j-1}(t) - x_d\big)\big\} \\ + k_d\big\{\big(v_{i,j+1}(t) - v_{i,j}(t)\big) - \big(v_{i,j}(t) - v_{i,j-1}(t)\big)\big\} \end{aligned} \tag{2.3}$$

A different control structure is adopted for the platoons' lead vehicles since we expect that the platoon may encounter other platoons as they travel on the road. Lead vehicles attempt to maintain a desired separation and speed, with respect to a preceding vehicle, by using a

control law given by [4]

$$u_{i,n}(t) = k_p\big(x_{i+1,1}(t) - x_{i,n}(t) - h.v_{i,n}(t)\big) + k_d\big(v_{i+1,1}(t) - v_{i,n}(t)\big) \tag{2.4}$$

where h is a time headway constant. Also, each lead vehicle is equipped with a transitional controller which is engaged in cases it encounters a slowly moving vehicle or a slowly driving platoon on the road. Interested readers are referred to [4] for more details on transitional controllers. W are interested in studying the effect of control transition. Therefore, we will adopt the Intelligent Driver Model (IDM) [26], which can predict human driving behavior, to simulate the dynamics of control transitioned vehicle(s). The commanded acceleration of the disbanded platoon vehicles is calculated using

$$\begin{aligned} u_{i,j}(t) &= u_{\max}\big\{1 - \big(v_{i,j}(t)/v_d\big)^4 - \big(s^*(t)/(x_{i,j+1}(t) - x_{i,j}(t))\big)^2\big\} \\ s^*(t) &= r_0 + v_{i,j}(t)\big\{\ h + (v_{i,j}(t) - v_{i,j+1}(t)/(2\sqrt{u_{\min}u_{\max}})\big\} \end{aligned} \tag{2.5}$$

where v_d is the desired velocity, $u_{\min}$, $u_{\max}$ are minimum and maximum acceleration, respectively, and r_0 is the minimum inter-vehicle separation (a vehicle cannot move if the separation is smaller than r_0). Finally, we assume that all vehicles are equipped with a collision-avoidance technique where $u_{\min}$ will be applied when the following condition is true [4, 24]

$$x_{i,j+1}(t) - x_{i,j}(t) \leq r_0 + \big(v_{i,j}^2(t) - v_{i,j+1}^2(t)\big)/2u_{\min} \tag{2.6}$$

2.2.2 Threat model

The aim of a disbanding attack in a multi-platoon scenario is to cause harm to the vehicles in some of the platoons by targeting one or more vehicle(s) in a different platoon and disrupting their automation. More specifically, this type of attacks relies on compromising some aspect of a vehicle's automation system so as to force the vehicle to abandon automated operation, i.e., transition of control, and hence cause the platoon to which it belongs to disband. The action of disbanding will then impact upstream platoons. As stated earlier,

the level of automation provided by the currently available automation technology is still not highly autonomous. Therefore, it is still expected that human drivers will need to take control of the automated vehicles during certain situations.

One possible attack vector that could be leveraged to compromise a vehicle's automation, and force a transition of control, is to target the vehicle's front and/or rear facing sensors, those that are relied upon to perceive the relative distance and speed of neighboring vehicles. Existing work has demonstrated that LIDAR, RADAR, camera, and ultra-sonic sensors, the most commonly-used sensors in automated vehicles for these purposes, can be jammed or spoofed and that such attacks can be targeted, easy to carryout, accomplished at a distance, and mounted against multiple vehicles at once [21, 27, 28, 29].

To demonstrate the impacts of the disbanding attack in our study, we assume the attacker has the capability to target the sensors of either one or multiple automated vehicles belonging to one or more platoons. Also, we assume that the mounted attack succeeds in degrading the sensing functionality of the automation system(s) employing the targeted sensor(s). We consider two possible scenarios resulting from this attack. In the case wherein a sensor of a single vehicle in a platoon is targeted and its automation compromised, the vehicle will utilize V2V communications and alert the other vehicles in that platoon so that they begin to transition their control[1]. In the case of targeting the sensor(s) of all vehicles in a platoon, the automation systems of those vehicles will suffer the disruption of the sensors operation, become unable to handle the current situation, and begin the process of transition of control. In either case, the automated vehicles are forced to transition their control in an attempt to mitigate the attack and avoid accidents, effectively disbanding the platoon.

Although the process of disbanding a platoon can help with avoiding accidents, the resulting action of braking will cause upstream effects on intact (non-attacked) platoons. Those effects pose a threat to the safety of these platoons, resulting in sudden and excessive velocity changes that could lead to collisions. Disbanding attacks are extremely effective,

[1]Disbanding (dissolving) a platoon when one vehicle reverts to manual control has been recommended in actual platooning systems [30].

as attack-resilient platooning controllers tend to ignore human intervention in the design process.

2.3 Human-in-the-loop Attacks

In this section, the disbanding attack is formulated as an optimization problem in order to find an optimal attack scenario. Then, the simulation setup to carry out such an attack is explained.

2.3.1 Finding an optimal disbanding attack

Given the attacker's capabilities and platoon dynamics as described in Section 2, the goal of the attacker is to find which platoon(s) and at which time(s) vehicles' sensors must be attacked, to induce disbanding, such that the velocity deviation of all intact vehicles is maximized. To assess the impacts of disbanding attacks on the simulated platoons, we use the following metrics

- Average velocity error (deviation): which describes the non-attacked platoons' slowing down as a result of disbanding another platoon(s). For the j^{th} vehicle in the i^{th} platoon, the average velocity error is defined as

$$E_v = \frac{1}{|T_s|} \sum_{k=1}^{|T_s|} \frac{|v_{i,j}(t_k) - v_d|}{v_d} * 100 \tag{2.7}$$

  where T_s is the attack window (in seconds), and v_d is desired speed. Since we are considering platoons, equation (2.7) is modified into the following

$$E_v = \frac{1}{N * n * |T_s|} \sum_{i=1}^{N} \sum_{j=1}^{n} \sum_{k=1}^{|T_s|} \frac{|v_{i,j}(t_k) - v_d|}{v_d} * 100 \tag{2.8}$$

  which calculates E_v for all vehicles $(N * n)$ throughout T_s.

- Collisions: although each vehicle is assumed to be equipped with a collision-avoidance algorithm, crashes between some of the intact vehicles can still occur according to

our simulations as a result of disbanding. Therefore, we will indicate whether the considered attack scenario involves collisions or not.

Let $\boldsymbol{p_d}$ be a vector of indices of platoons to be disbanded, and $\boldsymbol{t_d}$ a vector of times of disbanding. The attacker will solve

$$\begin{aligned} \underset{\boldsymbol{p_d},\boldsymbol{t_d}}{\text{maximize}} \quad & E_v = f(\boldsymbol{p_d}, \boldsymbol{t_d}) \\ \text{subject to} \quad & 1 \leq \boldsymbol{p_d} \leq N \\ & 1 \leq \boldsymbol{t_d} \leq T_s \\ & \boldsymbol{p_d}(i_1) \neq \boldsymbol{p_d}(i_2) \text{ for } i_1, i_2 = 1, \ldots, \text{ no. of targeted platoons} \end{aligned} \tag{2.9}$$

Equation (2.9) is interpreted as follows: given a number of targeted platoon(s), the attacker seeks the best values for $\boldsymbol{p_d}$ and $\boldsymbol{t_d}$ such that the highest value for the cost function E_v will result. The constraints of the problem ensure that values of $\boldsymbol{p_d}$ and $\boldsymbol{t_d}$ are within bounds and the same platoon cannot be disbanded twice (in case of multi platoon disbanding). We used the Genetic Algorithm (GA) Toolbox in MATLAB to solve equation (2.9).

2.3.2 Simulation setup

For the theoretical results presented in this work, we used MATLAB to simulate a string of platoons, using the control structures and dynamics from Section 2.2.1. Table 2.1 displays the data used in all subsequent simulations. In previous work, the value of τ was selected to be either 0.1 s [31] or 0.5 s [4]. To generalize the problem, we also simulated values in-between for (τ). To produce realistic simulations, all vehicles' velocities are ensured to be below or equal to a maximum value and all vehicles move only forward (no negative velocities). Also, the acceleration is bounded within minimum and maximum values. Since the vehicles' responses to initial separations and velocities may result in some overshoot before reaching the steady-state, all simulations were started at the steady-state so that that transient response will not interfere with the attack impacts.

2.3.3 Results

Two different cases of disbanding attacks are shown. In Fig. 2.3a, the attacker seeks disband the lead platoon, and in Fig. 2.3b, the attacker seeks to disband the lead two platoons, out of 10 total platoons. Results are shown in terms of the absolute speed of the last/rear vehicles of intact platoons (legends are removed to reduce visual clutter). We can see clearly that disbanding results in slowing down, hence deviating from the desired speed of 31 m/s, and even stopping completely. That behavior is captured by calculating E_v using (2.8) which is equal to 29.57% and 43.69% for Fig. 2.3a and 2.3b, respectively. For Fig. 2.3c and 2.3d, the total number of platoons (N) is varied between 2 and 10, as shown on the x-axis, and the actuator's delay (τ) is varied between 0.1 s to 0.5 s with an increment of 0.1 s, and the time headway (h) is selected equal to 1.5 s.

For each value of N, the solution of (2.9) indicated that the optimal disbanding attack always occurs by disbanding the lead platoon and at at time equal to 1 s (the beginning of the attack window). Fig. 2.3c shows the optimal (maximum) average velocity error (E_v) for disbanding the lead platoon and for different values of τ. We can see clearly that more severe attack impacts are induced as the total number of platoon increases. We have already assumed in Section 2.2.1 that all vehicles are equipped with an appropriate

Table 2.1. Parameters used in the simulations.

Parameter	Value	Description
N	[2:10]	number of platoons
n	10	number of vehicles per platoon
k_p	1	controller's proportional gain
k_d	5	controller's derivative gain
x_d	{5,4} m	desired inter-vehicle separation
v_d	31 m/s	nominal velocity
h	1.5 s	time headway
τ	{0.1,0.3,0.5} s	time-lag constant
$v_{\max}$	36 m/s	maximum velocity
$v_{\min}$	0 m/s	minimum velocity
$u_{\max}$	1 m/s^2	maximum acceleration
$u_{\min}$	-5 m/s^2	minimum acceleration
r_0	1 m	minimum inter-vehicle separation
T_s	180 s	simulation time

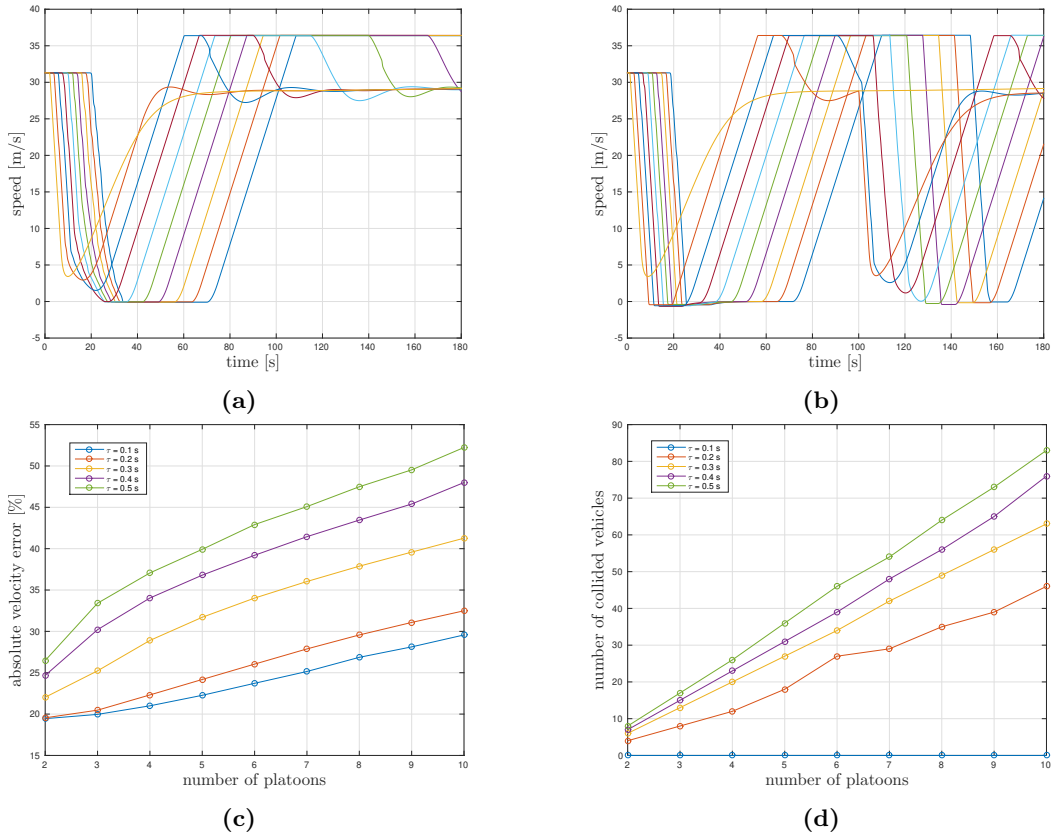


Fig. 2.3. a) Speed profiles of platoons' rear vehicles (N = 10) when the lead platoon started disbanding at t = 2 s. b) Speed profiles of platoons' rear vehicles (N = 10) when the 9^{th} and 10^{th} platoons started disbanding at t = 2 s and 100 s, respectively. c) Average velocity error for optimal single-platoon disbanding cases. d) Number of collided vehicles for optimal single-platoon disbanding cases.

collision-avoidance algorithm. However, simulation results indicate that disbanding attacks can also cause accidents between some of the vehicles in the intact platoons, those which were not the original target of the attack. Fig. 2.3d shows the number of colliding vehicles for each of the optimal disbanding attack cases, as displayed in Fig. 2.3c. We can see that regardless of N, collisions occur when the actuator's delay is greater than 0.1 s, and the total number of accidents increases as the total number of platoons increases.

2.4 Attack mitigation

We propose two approaches each of which proactively adjusts the commanded acceleration profiles of intact platoons' vehicles in an attempt to mitigate attack impacts, by

lessening the velocity deviations and reducing the number of collisions if possible. By executing each of the proposed approaches, the automation of intact platoons is maintained and no transition of control will be initiated.

2.4.1 Optimal mitigation

The mitigation of disbanding attacks impacts is formulated as an optimization control problem. The Model-Predictive Control (MPC) technique is used to find an on-line solution using receding horizon [32]. The MPC based formulation is an optimal control technique that has been used successfully in different applications [33]. It is based on minimizing a cost function (velocity deviation) in order to achieve a certain goal (mitigating disbanding attack impacts), while considering performance and physical constraints (collision-avoidance and speed and acceleration bounds). As such, this optimal approach will be used to compare and evaluate the performance of the other approach, as suggested in Section 2.4.2. However, this approach requires more computation power and more sophisticated infrastructure to perform the required calculations.

Using this approach, the objective is to compute a control sequence which will command each vehicle behind the disbanded platoon, in order to reduce the deviation in velocity and avoid accidents. More specifically, the controller of an intact vehicle will use the current measurements of velocity and acceleration in order to solve

$$\min_{\mathbf{U}} \quad 2\mathbf{M_1}\mathbf{U} + \mathbf{U}^T\mathbf{M_2}\mathbf{U} \tag{2.10}$$

$$\text{s.t.} \quad \mathbf{M_3}\mathbf{U} \leq \mathbf{M_4} \tag{2.11}$$

where $\boldsymbol{U}$ is the resulting control sequence and $\boldsymbol{M_1}$, $\boldsymbol{M_2}$, $\boldsymbol{M_3}$, and $\boldsymbol{M_4}$ are matrices formulated to consider acceleration and physical speed limits, as well as collision avoidance. The complete formulation for this controller is given in Appendix A. While this approach would yield an optimal solution for every time instance, it requires global knowledge of the platoon dynamics. Namely, to perform the calculations needed to produce $\boldsymbol{U}$, to command intact vehicles, speed and acceleration measurements of all related vehicles should be available to

a centralized infrastructure responsible for those calculations, basically a central computer with (V2I) and (I2V), wherein the needed capabilities exist to receive current measurements, perform the required calculations, and transmit the resulting acceleration commands back to the corresponding vehicles. It has been shown that such a communication structure is feasible [25], but not likely to be deployed in the near term, as it presents a single-point of failure. For that reason, in the next section we suggest an efficient heuristic mitigation approach which requires a less sophisticated communication model and produces nearly equivalent results to the optimal approach.

2.4.2 Efficient heuristic mitigation

The goal of this approach is to modify the commanded acceleration of a vehicle by comparing the distance it will cover with the distance that will be covered by the preceding vehicle during a predefined time horizon (t_s). Initially, the acceleration commands of both vehicles are calculated according to the platooning control structures given in Section 2.2.1. Let us consider a vehicle in an intact platoon ($V_{current}$) and a preceding vehicle ($V_{preceding}$), where subscripts ($current$) and ($preceding$) refer to two adjacent vehicles belonging either to the same platoon or to two different adjacent platoons. Each vehicle's dynamics are described by

$$\begin{aligned} \dot{x}_m(t) &= v_m(t), \\ \dot{v}_m(t) &= u_m(t), \end{aligned} \tag{2.12}$$

where $m \in \{current, preceding\}$, $t \in [t_s(1) : \Delta t_s : t_s(end)]$, $t_s(1)$ and $t_s(end)$ are the first and last time samples of the time horizon t_s, and Δt_s is the time increment. Under the assumption that u_m is constant for the duration of t_s and using the forward difference approximation [34], the absolute position and velocity can be calculated as follows

$$\begin{aligned} x_m\big(t_s(k+1)\big) &= x_m\big(t_s(k)\big) + \Delta t_s v_m\big(t_s(k)\big), \\ v_m\big(t_s(k+1)\big) &= v_m\big(t_s(k)\big) + \Delta t_s u_m, \end{aligned} \tag{2.13}$$

Algorithm 1: Heuristic mitigation

Input: $v_m(t_s(1)), u_m(t_s(1))$, for $m \in \{current, preceding\}$ // velocity and commanded acceleration values of current and preceding vehicles.

Output: $u^{new}_{current}$, // new commanded acceleration value for current vehicle.

$u^{new}_{current} \leftarrow u_{current}(t_s(1))$;

compute d_m for the interval of t_s using input data;

if $d_{preceding} < d_{current}$ **then**

  $u^{new}_{current} \leftarrow \frac{d_{preceding} - v_{preceding}\big(t_s(end) - t_s(1)\big)}{0.5\big(t_s^2(end) - t_s^1(1)\big)}$;

  if *current vehicle and preceding one will collide during* t_s **then**

    search for $u^{new}_{current}$ within $\big[a_{min}, u_{preceding}\big)$;

where $k = 1, \ldots, |t_s|$. Once the vector $x_m(.)$ is obtained, the distance traveled by vehicle (V_m) during t_s can be calculated as $d_m = x_m\big(t_s(end)\big) - x_m\big(t_s(1)\big)$. Based on the calculated distance travelled by the current vehicle $d_{current}$ and that of the preceding vehicle $d_{preceding}$, we proceed as follows

- If $(d_{preceding} < d_{current})$, then $(V_{current})$ is covering more distance and may collide with a preceding vehicle. Therefore it has to slow down by modifying its commanded acceleration $(u_{current})$. To produce the same traveled distance for $V_{current}$, $u_{current}$ is selected equal to u_{new} which is calculated as

$$u_{new} = \frac{d_{preceding} - v_{current}\big(t_s(end) - t_s(1)\big)}{0.5\big(t_s^2(end) - t_s^2(1)\big)}, \tag{2.14}$$

  Using the new acceleration command, another important consideration is to ensure that the predicted position vectors of $(V_{current})$ and $(V_{preceding})$, calculated using (2.13), will not overlap (collide) during the interval of t_s. If that is the case, then acceleration need to be further modified and selected from the interval $\big[a_{\min} : \Delta a : u_{new}\big)$ where Δa is a suitable acceleration increment. Namely, $u_{current}$ is set equal to the first value smaller than u_{new} within that interval. If the new value produces no collisions, then it is applied. Otherwise, the next value is selected and so on.

- If $(d_{preceding} \geq d_{current})$, then the commanded acceleration $u_{current}$, calculated according to the platooning control laws from Section 2.2.1, is maintained.

The steps of this approach are shown in Algorithm 1. Once the disbanding attack against one of the platoons is detected, as explained in Section 2.2.2, the last vehicle of the disbanded platoon will inform the following lead intact vehicle, using the established inter-vehicle communication. The latter vehicle will calculate its platooning acceleration command and modify it, if needed, using this mitigation approach. Furthermore, it will also inform the following vehicle to implement similar steps. Practically, in order to implement the suggested approach locally on a certain vehicle, the following information should be available: the commanded acceleration of both the current vehicle (measured locally) and the preceding vehicle (transmitted via the already established communication), and the velocity of both the current vehicle (measured locally) and the preceding vehicle (estimated form the measurements of velocity and relative velocity). The process described above will be repeated at the next time instant, using the newly-obtained measurements. $V_{current}$ will reuse the adopted platooning control law once the inter-vehicle distance, with respect to $V_{preceding}$, begins to increase. Finally, it should be noted that our platooning model (Section 2.2.1) requires a far less sophisticated communication model to connect any two neighboring vehicles, performs a decentralized mitigation, and produces equivalent results to the MPC-based mitigation. Hence, it is not only cheaper to implement the heuristic approach rather than the MPC-based one, but the former is also more resilient.

Table 2.2. Results for optimal one-platoon disbanding attack

x_d	τ	E_v [%]			Crash		
[m]	[s]	baseline	mit.1	mit.2	baseline	mit.1	mit.2
5	0.1	29.570	24.283	23.025	No	No	No
	0.3	41.268	25.556	25.182	Yes	No	No
	0.5	52.235	28.482	28.709	Yes	Yes	Yes
4	0.1	27.995	25.063	22.798	Yes	No	No
	0.3	40.115	26.864	24.823	Yes	No	No
	0.5	52.706	29.079	29.742	Yes	Yes	Yes

2.4.3 Results and discussion

Table 2.2 displays the average velocity error E_v collected from different scenarios, involving the optimal single-platoon disbanding attack. Baseline, mit.1, and mit.2 refer to platoons using the control structure from Section 2.2.1, the heuristic mitigation, and the MPC-based mitigation, respectively. For all cases given, the total number of platoons is equal to 10, while the inter-vehicle separation x_d and actuator's delay τ parameters are varied, in order to generate different scenarios.

We can see in Table 2.2 that the baseline control does not perform well against the disbanding attack, since all cases involve accidents (except for $x_d = 5$ m and $\tau = 0.1$ s) and an increase in E_v. On the other hand, it is clear that our approach improves the values of E_v for all attack cases. In addition, collisions are avoided in most attack cases except when τ equals to 0.5 s. Also, the heuristic approach reduces the number of colliding vehicles. For example, the attack case with $x_d = 4$ m resulted in accidents involving 58% and 29% of the total number of vehicles (100) for the baseline and mit.1, respectively. Furthermore, the attack case with ($x_d = 5$ m, $\tau = 0.1$ s), which had no accidents, 80% of the intact vehicles experienced stop-and-go once, due to the use of a collision-avoidance algorithm that functioned by applying maximum deceleration. However, in our approach, and for all attack cases, all intact vehicles slowed down gradually and did not have to come to a complete stop. Using mit.2 also helps with improving the values of E_v and avoiding collisions. By comparison, we can see that the values of E_v for both mit.1 and mit.2 are very equal. In fact, it is clear that our approach improves the results, in terms of lowering

Table 2.3. Results for optimal two-platoon disbanding attack

x_d	τ	E_v [%]			Crash		
[m]	[s]	baseline	mit.1	mit.2	baseline	mit.1	mit.2
5	0.1	38.347	27.056	26.221	Yes	No	No
	0.3	39.839	28.548	29.129	Yes	No	Yes
	0.5	45.004	35.724	38.690	Yes	Yes	Yes
4	0.1	37.069	30.731	29.811	Yes	No	No
	0.3	40.233	33.183	34.868	Yes	Yes	No
	0.5	45.823	38.349	38.914	Yes	Yes	Yes

E_v and avoiding collisions, in some attack cases. Overall, these numbers demonstrate that our heuristic approach produces nearly equivalent results to the optimal MPC approach.

Table 2.3 shows data for E_v collisions for various cases involving two platoons disbanding, wherein the total number of platoons is equal to 10. The optimal attack is found to occur by targeting the 10^{th} (lead) and 9^{th} platoons in the formation at times equal to 2 s and 100 s, respectively within T_s. We can see that the baseline control produces collisions for all attack cases. However, with either mit.1 or mit.2, the reduction in velocity is minimized and crashes are avoided completely in some cases. Also, the results for both mitigation approaches are nearly equivalent. Furthermore, in comparison with Table 2.2, even with mitigation, the two-platoon disbanding attack results in more crashes, which indicates that it is a more severe attack, as compared to disbanding a single platoon.

2.5 Experimental Validation

We validate our proposed mitigation algorithm on a platooning testbed and compare it with the baseline algorithm i.e., regular platoon control law with integrated collision avoidance.

2.5.1 Hardware Setup

Our experimental setup consists of small robots that represent vehicles in a stream of platoons and a motion capture system for tracking as shown in Fig. 2.4. We implemented the disbanding attack and the traveled distance mitigation algorithm on three 3-vehicle platoons, denoted as per the convention shown in Fig. 2.2. The 3^{rd} (leading) platoon disbands and the response of other two platoons is captured with the different algorithms in place.

Each robot is affixed with multiple IR markers which are tracked by the Optitrack motion capture system consisting of 24 IR cameras and the Motive software that enables us to capture the robot positions. This position data is then streamed to a command computer where an interface application utilizing the Robot Operating System (ROS) [35] framework makes the gathered position data for each robot available to our controller application. This

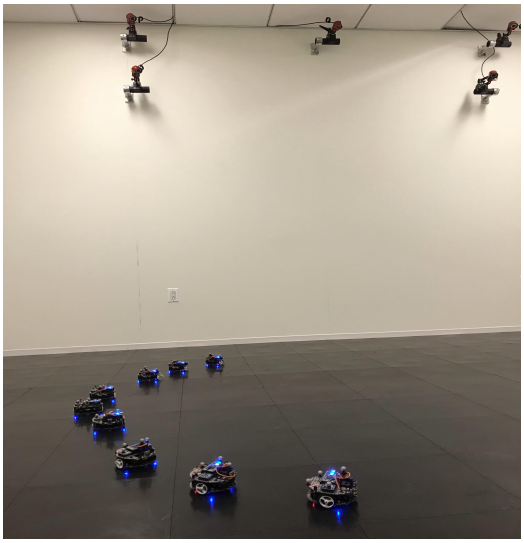


Fig. 2.4. Experimental environment with small robots and motion capture system

application processes the position data and sends control commands accordingly to each robot. The controller application implemented on ROS works in the following manner:

- The raw position data is processed using an Extended Kalman Filter to reduce camera sensor noise and estimate the measured position and velocity.
- *Pure Pursuit Controller* utilizes the extimated positions and circular path coordinates from the experiment environment to calculate the angular velocity command for each vehicle.
- The estimated data of all vehicles is used to calculate the relative distance and velocity between consecutive vehicles. This is then fed to a *High level Controller* which implements the platoon model following the bidirectional control law as explained in Section 2.2.1 and provides desired acceleration values for the robots.
- The mitigation and baseline algorithm then modify the acceleration values from the *High level Controller* in case a disbanding attack is detected.

- As the vehicles only act upon instantaneous velocity commands, these acceleration values along with current measured velocities are used to calculate the desired velocities for each vehicle. The desired linear velocities for the vehicles are achieved using a PI controller which acts as our *Low Level Controller*. This controller calculates the linear velocity commands for each vehicle such that the measured and the desired velocities match.

Each robot consists of a 32-bit ARM-based mbedNXP LPC1768 microcontroller on the Pololu m3pi platform to which the Digi Xbee receivers are interfaced. The corresponding Xbee transmitter is connected to the command computer. These Xbee modules allow us to establish a wireless communication channel using the Zigbee protocol over which the angular and linear velocity commands calculated for each robot using our controller application are then broadcast. The firmware on these robots receive the broadcast messages and calculate the left and right wheel speeds from the received angular and linear velocities as per the differential drive model.

2.5.2 Experimental Results

Fig. 2.5 shows individual velocity profiles for the vehicles under consideration (three platoons with three robots in each). Fig. 2.5a indicates the effect on velocity due to disbanding for the baseline control structure, given in Section 2.2.1, wherein we can see vehicles in the last platoon not only slow down suddenly, but one of them stops, in response to the disbanding of the lead platoon. Fig. 2.5b and 2.5c give the velocity profiles when intact robots use the traveled distance mitigation approach, wherein it can be seen that the speed of vehicles in second and third platoon slow down gradually and then begin to accelerate. This mitigation approach was tested with $t_s = 0.5$s and 1s, respectively. The point labeled as A in Fig. 2.5a, 2.5b, and 2.5c indicate that the platoons are in a steady state. Point B marks the time at which the attack on the lead platoon is emulated, causing all of its vehicles to disband and suddenly decelerate. Deceleration patterns of the vehicles after point B for the baseline structure clearly indicate a sudden drop in velocities for the

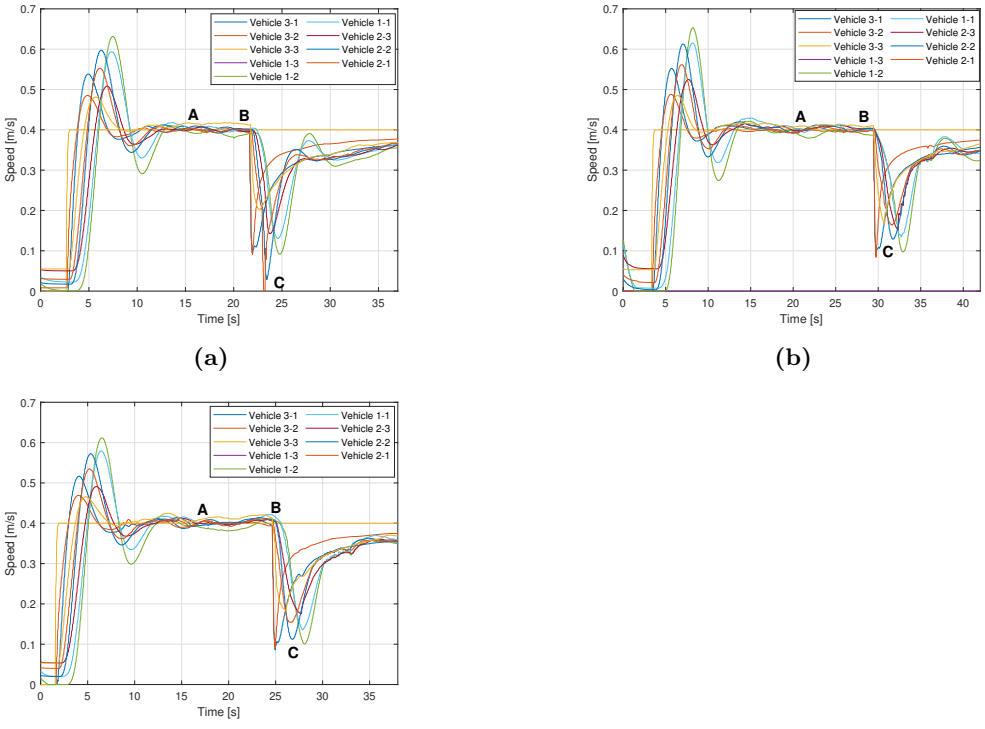


Fig. 2.5. Vehicles' velocities upon disbanding of platoon 3 for baseline control structure and proposed heuristic mitigation algorithm with $t_s = 0.5$s and 1s.

following platoons, causing some vehicles to come to a complete stop as indicated by point C.

While there are no collisions with the baseline control, sudden deceleration/ acceleration was observed. However, such abrupt changes in velocities are not observed when our proposed heuristic mitigation is in place (see Fig. 2.5b and 2.5c, where point C shows that none of the vehicles need to come to a halt). With the mitigation approach, vehicles comfortably decelerate and gradually accelerate to recover and maintain desired spacing and velocities, all without collisions. Furthermore, E_v was calculated for the three experiments and it was equal to 30.02%, 21.82% and 19.73% for Fig. 2.5a, 2.5b and 2.5c, respectively. These numbers indicate that with increasing t_s, the change in velocity is even smoother and more gradual, yet collisions do not occur. However, with $t_s = 1$ s, vehicles come to closer proximity, when compared with the results of $t_s = 0.5$ s. For reference, we have also

uploaded short videos of our experiments [36].

2.6 Conclusion

In this chapter, we presented and studied a disbanding attack which targets vehicular platoons and causes severe deviations in speed, including stop-and-go traffic and collisions between upstream vehicles. The attack exploits human-in-the-loop control, whereby a vehicle switches from automated control to human driving at the onset of an attack against a vehicle sensing system. Calculations of key attack factors, such as identifying both the platoon(s) to disband and time to disband, were carried out. Additionally, we proposed two mitigation algorithms that reduce sudden velocity changes and also decrease the number of accidents, hence ensuring resilient performance for platoons. Simulations and experimental results corroborate theory, displaying improved velocity deviations. Finally, the proposed heuristic mitigation approach was implemented and verified on a hardware testbed with a motion capture system and mobile robots representing platoons, and even at this stage, it showed better performance than baseline control structure.

REFERENCES

[1] E. Coelingh and S. Solyom, "All aboard the robotic road train," *IEEE Spectrum*, vol. 49, pp. 34–49, 2012.

[2] W. Ren and D. Green, "Continuous platooning: a new evolutionary operating concept for automated highway systems," in *American Control Conference (ACC), 1994*, June 1994.

[3] K.-Y. Liang, J. Mårtensson, and K. H. Johansson, "Fuel-saving potentials of platooning evaluated through sparse heavy-duty vehicle position data," *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 1061–1068, 2014.

[4] R. Rajamani, *Vehicle Dynamics and Control*, ser. Mechanical Engineering Series. Springer, 2011. [Online]. Available: https://books.google.com/books?id=eoy19aWAjBgC

[5] W. Vlakveld, S. W. O. Verkeersveiligheid, and V. e. L. Rijkswaterstaat. Water, *Transition of Control in Highly Automated Vehicles: A Literature Review*. SWOV Institute for Road Safety Research, 2015. [Online]. Available: https://books.google.com/books?id=D2mVjwEACAAJ

[6] M. Blanco, J. Atwood, H. M. Vasquez, T. Trimble, V. L. Fitchett, J. Radlbeck, G. Fitch, S. M. Russell, C. A. Green, B. Cullinane, and J. Morgan, "Human factors evaluation of level 2 and level 3 automated driving concepts," 08 2015.

[7] "Truck platooning vision 2025," 2016. [Online]. Available: www.eutruckplatooning.com

[8] N. Merat, A. H. Jamson, F. C. Lai, M. Daly, and O. M. Carsten, "Transition to manual: Driver behaviour when resuming control from a highly automated vehicle," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 27, pp.

274 – 282, 2014, vehicle Automation and Driver Behaviour. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1369847814001284

[9] R. Zheng, K. Nakano, S. Yamabe, M. Aki, H. Nakamura, and Y. Suda, "Study on emergency-avoidance braking for the automatic platooning of trucks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1748–1757, Aug 2014.

[10] J. Axelsson, "Safety in vehicle platooning: A systematic literature review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1033–1045, May 2017.

[11] N. Merat and A. Jamson, "How do drivers behave in a highly automated car?" 10 2017, pp. 514–521.

[12] C. Gold, D. Damböck, L. Lorenz, and K. Bengler, ""take over!" how long does it take to get the driver back into the loop?" *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 57, no. 1, pp. 1938–1942, 2013. [Online]. Available: https://doi.org/10.1177/1541931213571433

[13] A. Eriksson and N. A. Stanton, "Takeover time in highly automated vehicles: Noncritical transitions to and from manual control," *Human Factors*, vol. 59, no. 4, pp. 689–705, 2017, pMID: 28124573. [Online]. Available: https://doi.org/10.1177/0018720816685832

[14] S. Oncu, J. Ploeg, N. van de Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Network-aware analysis of string stability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1527–1537, Aug 2014.

[15] R. M. Gerdes, C. Winstead, and K. Heaslip, "Cps: an efficiency-motivated attack against autonomous vehicular transportation," in *Proceedings of the 29th Annual Computer Security Applications Conference*. ACM, 2013, pp. 99–108.

[16] S. Dadras, R. M. Gerdes, and R. Sharma, "Vehicular platooning in an adversarial environment," in *Proceedings of the 10th ACM Symposium on Information, Computer*

and Communications Security, ser. ASIA CCS '15. New York, NY, USA: ACM, 2015, pp. 167–178. [Online]. Available: http://doi.acm.org/10.1145/2714576.2714619

[17] B. DeBruhl, S. Weerakkody, B. Sinopoli, and P. Tague, "Is your commute driving you crazy?: a study of misbehavior in vehicular platoons," in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 2015, p. 22.

[18] R. W. van der Heijden, T. Lukaseder, and F. Kargl, "Analyzing attacks on cooperative adaptive cruise control (cacc)," *arXiv preprint arXiv:1710.05789*, 2017.

[19] D. D. Dunn, S. A. Mitchell, I. Sajjad, R. M. Gerdes, R. Sharma, and M. Li, "Regular: Attacker-induced traffic flow instability in a stream of semi-automated vehicles," in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, June 2017, pp. 499–510.

[20] M. Jagielski, N. Jones, C.-W. Lin, C. Nita-Rotaru, and S. Shiraishi, "Threat detection for collaborative adaptive cruise control in connected cars," in *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, ser. WiSec '18. New York, NY, USA: ACM, 2018, pp. 184–189. [Online]. Available: http://doi.acm.org/10.1145/3212480.3212492

[21] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," *Black Hat Europe*, vol. 11, p. 2015, 2015.

[22] C. Bergenhem, S. Shladover, E. Coelingh, C. Englund, and S. Tsugawa, "Overview of platooning systems," in *Proceedings of the 19th ITS World Congress, Oct 22-26, Vienna, Austria (2012)*, 2012.

[23] D. Yanakiev and I. Kanellakopoulos, "A simplified framework for string stability analysis in ahs," in *Proceedings of the 13th IFAC World Congress, 1996*, 1996, pp. 177–182.

[24] M. Amoozadeh, A. Raghuramu, C. Chuah, D. Ghosal, H. M. Zhang, J. Rowe, and K. Levitt, "Security vulnerabilities of connected vehicle streams and their impact on

cooperative driving," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 126–132, June 2015.

[25] C. Chou, C. Li, W. Chien, and K. Lan, "A feasibility study on vehicle-to-infrastructure communication: Wifi vs. wimax," in *2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, May 2009, pp. 397–398.

[26] A. Kesting, M. Treiber, and D. Helbing, "Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity," *Philosophical Trans. of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4585–4605, 2010.

[27] R. Chauhan, R. M. Gerdes, and K. Heaslip, "Attack against an fmcw radar," in *Proceedings of Embedded Security in Cars Conference*, 2014.

[28] C. Yan, W. Xu, and J. Liu, "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle," *DEF CON*, vol. 24, 2016.

[29] E. Yeh, J. Choi, N. Prelcic, C. Bhat, and R. Heath Jr, "Security in automotive radar and vehicular networks," *submitted to Microwave Journal*, 2016.

[30] T. Robinson, E. Chan, and E. Coelingh, "Operating platoons on public motorways: An introduction to the sartre platooning programme," in *17th world congress on intelligent transport systems*, vol. 1, 2010, p. 12.

[31] E. N. N. v. d. W. J. Ploeg, B. Scheepers and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *International IEEE Conference on Intelligent Transportation Systems*, 2011, pp. 260–265.

[32] J. B. Rawlings, "Tutorial overview of model predictive control," *IEEE Control Systems Magazine*, vol. 20, no. 3, pp. 38–52, June 2000.

[33] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*, A. Garulli and A. Tesi, Eds. London: Springer London, 1999, pp. 207–226.

[34] F. Borrelli, *Constrained optimal control of linear and hybrid systems*. Springer, 2003, vol. 290.

[35] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[36] Anonymous, "Mitigation and baseline algorithm experiments," June 2018. [Online]. Available: https://www.youtube.com/channel/UCI-UGJKT7C5E_8bs391LCpA

CHAPTER 3

REACHABILITY ANALYSIS FOR CONSTRAINED FALSE DATA INJECTION ATTACKS ON VEHICULAR PLATOONS

Vehicular platooning promises to bring faster, safer, and more efficient transportation. Automated platooned vehicles will rely on information obtained from inter-vehicle communication channels and on-board sensors to make driving decisions and achieve platooning. However, such reliance creates an opportunity for safety violating attacks intended to disrupt platoon formation and cause accidents. In this chapter, we investigate attacks mounted against the sensing functionality of platooned vehicles with the goal of manipulating the relative distance and speed measurements. More specifically, we are interested in approximating the set of final unsafe states, that can be reached by mounting realistically-constrained attacks capable of injecting false-data against an attacked sensor(s). For that purpose, we will use reachability analysis which enables us to realize whether it is possible to drive a platoon from initial to final states, given performance and physical bounds. Our results suggest that this type of attack is able to steer a platoon towards dangerous states and thereby generate impacts on passengers' safety by causing high-speed crashes.

3.1 Introduction

Vehicular platooning is a cyber-physical system (CPS) that employs automation, communication, sensing, and decision making capabilities with the objectives of combining a number of automated vehicles to follow each other while regulating their movements and maintaining predefined inter-vehicle distances and relative speeds. Vehicular platoons are gaining rapid interest and development, both academically and commercially, as they have shown numerous benefits, such as providing a safe and comfortable environment for the passengers while allowing them to focus on tasks other than driving [1]. They also have shown the ability to reduce traffic congestion on highways [2, 3], which leads to a better

and more efficient usage of roads and to fuel consumption [4].

In order to achieve the aforementioned objectives, each platooned vehicle implements a properly designed controller that determines the appropriate throttle and brakes commands [5] by using information collected from local sensors and from other vehicles through either inter-vehicle communication [6] or external networks [7]. As a result, vehicular platoons have a potential attack surface that can be exploited by malicious parties (attackers) and may result in disruptive platoon behavior. For this reason, security of platooning CPS has been widely researched in order to both define possible vulnerabilities that can be exploited by attackers, and to understand possible consequences, such as disarranging normal performance of platooned vehicles and causing fatal impacts, such as collisions at high speeds [8] or significantly-increased energy consumption in platoon vehicles due to modification of the behavior of one vehicle [9]. In addition, attacking vehicular platoons could also induce oscillations in vehicle movements, which lead to passenger discomfort, platoon instability, and reduced efficiency of platoon operation. These consequences can be triggered by an attacker capable of either controlling one vehicle in the platoon [10], multiple vehicles in a platoon [11], or of modifying messages transmitted between vehicles through communication channels [12, 13, 14]. False Data Injection (FDI) attacks are carried out by an adversary capable of manipulating the readings of CPS sensors, and thereby causing misbehavior in the decision-making process. Such attacks have been proven possible in previous works, such as [15], wherein authors propose conditions under which an FDI attack could destabilize an LTI control system without being detected. It has also been shown that an FDI attack could be staged against an electricity power grid, whereby the attacker had full or limited access to some of the installed meters or network topology and could thereby introduce errors in the states estimation, which could lead to unreliable operation of the power grid [16, 17]. Furthermore, the states estimation process was also the target of another FDI attack, as presented in [18], which was designed to compromise a subset of LTI discrete-system sensors without being detected.

As mentioned earlier, a platooning CPS controller requires measurements from on-board sensors for its operation. Existing work has demonstrated that RADAR, LIDAR, cameras, and ultra-sonic sensors, the most-used sensors in automated vehicles, can be victims of FDI attacks executed at a distance. For example, [19] presents external jamming and spoofing attacks that can be carried out against ultrasonic sensors and cameras, and experimental results show a possibility of malfunctioning a Tesla vehicle. Also, it was proven possible to falsify the readings of a vehicle's RADAR [20], LIDAR, or cameras [21] and, as a result, disrupt the behavior of the automated vehicle. In addition, FDI attacks on a vehicle's sensors have demonstrated impacts on the vehicle's platoon. Authors of [22] provide an analysis of different types of platooning controllers under FDI attacks mounted against RADAR and LIDAR sensors. Their results showed that such an attack could potentially lead to a crash between vehicles further behind the attacked vehicle. Also, the authors of [23] studied FDI attacks which compromised vehicular sensors measuring position, velocity, or both. The results showed multiple impacts, including passenger discomfort and collisions.

Reachability analysis defines the reachable set of a dynamic system, that is the set of all system states that can be attained within a finite time. Considering the physical bounds and performance constraints of the system states and inputs, reachability analysis helps us verify whether from a given initial point, a system can eventually reach another given final point. With that in mind, reachability can be applied to real-world applications where safety must be determined, such as collision avoidance problems in airplanes [24] and vehicles [25], or controller design for the platooning of unmanned aerial vehicles (UAV) [26, 27].

In this chapter, we are also concerned with the safety of a vehicular platoon operating in an adversarial setting wherein it is possible to target one or more of the platoon vehicle's sensors with an FDI attack. Particularly, we are interested in defining the set of final states that the platoon can reach as a result of experiencing a manipulation in the measurements obtained from one or more of the locally-equipped sensors. For that purpose, we will use the optimal control-based reachability approach to determine the reachable (final) set of states,

as it allows us to include attacker capabilities, physical limits on the vehicle's acceleration and speed, and resolution and physical limits of the attacked sensor(s) in the problem formulation as constraints, as will be explained Section 3.4. Furthermore, this approach requires a prior definition of the final states of interest. For that reason, and because our primary concern is the safety of the platoon, we will focus on unsafe states which can be translated as a collision between two or more vehicles in the platoon and at different speeds. Regardless of the type of equipped sensors, from this point on we will refer to the sensors measuring relative distance and speed as range and range-rate sensors, respectively. The contributions of this chapter are:

- We analyze the performance of a vehicular platoon undergoing an FDI attack mounted against one or multiple locally equipped range and range rate sensors. Specifically, we will define what conditions and capabilities are required by an adversary to make such an attack capable of violating the safety of the platoon.
- To generalize the problem, we will define threat models for targeting either the range sensors, range-rate sensors, or both. Also, we analyze the resulting impacts from those attacks
- After defining both the platoon and threat models, wherein the attack vector will be acting as the new control input to the system, we will use the optimal control-based reachability approach to determine the final reachable set by the platoon. This will show whether collision(s) are possible as a result of an FDI attack.

3.1.1 Related work

In the context of FDI attacks on vehicular sensors, the authors of [19] present external jamming and spoofing attacks that can be carried out against ultrasonic sensors and cameras, and experimental results even show a possibility of malfunctioning a Tesla vehicle. Also, it was proven possible to falsify the readings of a vehicle's RADAR [20], LIDAR, or cameras [21], and, as a result, disrupt the behavior of the automated vehicle. In addition, FDI attacks on a vehicle's sensors have demonstrated impacts on the vehicle's platoon.

Authors of [22] provide an analysis performed on various types of platooning controllers under FDI attacks mounted against RADAR and LIDAR sensors. Their results show that such attacks could potentially lead to a crash between vehicles further behind the attacked vehicle. Also, [23] studied FDI attacks which compromised platooned vehicles sensors measuring position, velocity, or both. These results show multiple impacts including passenger discomfort and collisions.

In the context of conducting reachability analyses, various methods have been proposed for obtaining the reachable sets. In [28, 29], ellipsoidal techniques are used to calculates outer elliptical bounds around the reachable set of a linear system. This method has been applied in problems such as finding algorithms for collision avoidance in UAVs [30] and determining new artificial physical bounds for a system's actuators [31]. Another method is generally known as Hamilton-Jacobi (HJ) reachability [24] and has been used to solve problems such as auto landing of an aircraft [32], and path planning for UAVs [33]. Finally, another method suggested for determining the reachable set is based upon using optimal control theory, wherein the final states points are included in the formulation of an optimization problem which, in turn, calculates the appropriate control sequence required to drive the system towards those final states and also considers state/input constraints [34]. This method has been applied in problems such as determining a safe landing area for a moon lander [35], and suggesting an alternate trajectory for vehicles to be tracked to avoid colliding with other vehicles [36].

In the context of security of vehicular platooning, reachability analysis has been used to quantify the impacts of attacks mounted against vehicular platoons. In [31], the authors defined reachable sets that a CACC-based platoon could reach while experiencing an attack on its V2V communication channels. The resulting sets included unsafe states for the platoon, wherein two or more vehicles could crash. In [37], the authors investigated the behavior of an ACC-based platoon during a motion modifying attack, wherein the attacker controls one of the platooned vehicles. The resulting reachable sets revealed that accidents are possible as a result of that attack as well. Similar to the aforementioned works, we are

also using reachability analysis to study the safety of a vehicular platoon. However, we will use that analysis with realistic scenarios of FDI attacks.

3.1.2 Organization

Section 3.2 explains the vehicle, vehicular platoon, and threat models used in this chapter. In Section 3.3, we present an algorithm to determine an FDI attack vector that induces instability in the attacked vehicle. Section 3.4 describes the approach used in this chapter to conduct the reachability analysis for constrained FDI attacks on platooned vehicles' sensors while Section 3.5 discusses the reachable sets resulting form those attacks. Finally, conclusions are given in Section 3.6.

3.2 System Model

The modeling of platooned vehicles as well as the control strategies, to achieve platooning, are discussed in this section.

3.2.1 Vehicle model

We consider a homogeneous platoon with n vehicles, which means that all vehicles share the same dynamics, controller design, and performance characteristics. In general, each platooned vehicle's dynamics are described as

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{3.1}$$

where $\boldsymbol{x}$ and $\boldsymbol{u}$ are the state and input (commands) vectors, respectively. The evolution of each vehicle's states over time is described as follows

$$\begin{aligned} \dot{x}_i(t) &= v_i(t) \\ \dot{v}_i(t) &= u_i(t), \text{ for } i = 1, \ldots, n \end{aligned} \tag{3.2}$$

where x_i, v_i, and u_i to the i^{th} vehicle's absolute position, absolute velocity, and commanded acceleration, respectively, and n is the number of vehicles in a platoon.

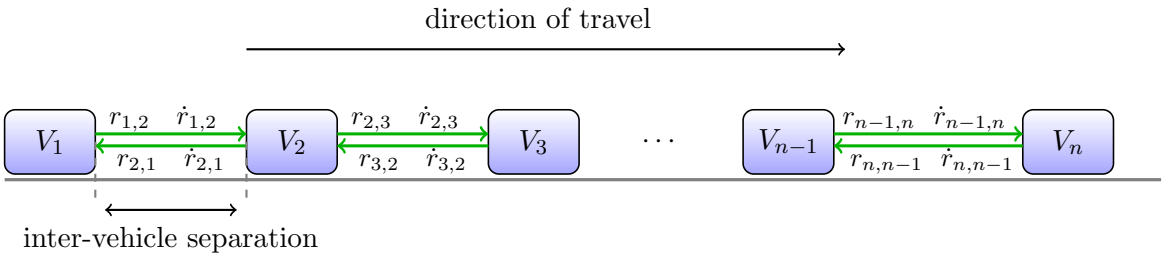


Fig. 3.1. A platoon with n vehicles. $r_{i,j}$ and $\dot{r}_{i,j}$ represent the relative distance and speed, respectively, measured by the i^{th} vehicle's range and range-rate sensors with respect to the j^{th} vehicle.

3.2.2 Platoon model

In this chapter, we consider a platoon with n vehicles, as shown in Fig. 3.1, where each vehicle is equipped with range and range-rate sensors. For platooned vehicles equipped with an ACC control structure, the latter utilizes information provided by the local vehicular sensors. For each platooned vehicle, the error coordinates are defined as follows

$$\begin{aligned} e_{xi}(t) &= x_{i+1}(t) - x_i(t) - x_d \\ e_{vi}(t) &= v_{i+1}(t) - v_i(t) \end{aligned} \tag{3.3}$$

where e_{xi} and e_{vi} refer to the i^{th} vehicle's position and velocity errors, respectively, and x_d is a constant denoting inter-vehicle desired separation. It should be noted that error states are fully measured using the locally equipped range and range-rate sensors. The evolution of error states over time can be described as follows

$$\begin{aligned} \dot{e}_{xi}(t) &= v_{i+1}(t) - v_i(t) \\ \dot{e}_{vi}(t) &= u_{i+1}(t) - u_i(t) \end{aligned} \tag{3.4}$$

The position and velocity errors can be described for all vehicles in a platoon using the following state-space representation

$$\dot{e}(t) = A_1 e(t) + B_1 u(t) \tag{3.5}$$

where

$$e(t) = \begin{bmatrix} e_{x1}(t) & \dots & e_{xn}(t) & e_{v1}(t) & \dots & e_{vn}(t) \end{bmatrix}^T$$
$$u(t) = \begin{bmatrix} u_1(t) & \dots & u_n(t) \end{bmatrix}^T$$

Matrices A_1 and B_1 are described in Appendix B. Each platooned vehicle uses a bidirectional control law to determine its commanded acceleration [38]. Bidirectional control is able to guarantee platoon string stability, which maintains proper traffic flow [5, 38], and it does not need any (V2V) transmitted information to generate driving decisions. Each vehicle's commanded acceleration is calculated according to its position in the platoon. For the last vehicle in a given platoon, we have

$$u_1(t) = k_p e_{x1}(t) + k_d e_{v1}(t), \tag{3.6}$$

where k_p and k_d are the controller's proportional and derivative gains, respectively. For the rest of the vehicles in the platoon, we have

$$\begin{aligned} u_i(t) &= k_p\big(e_{xi}(t) - e_{xi-1}(t)\big) + k_d\big(e_{vi}(t) - e_{vi-1}(t)\big), \\ &\text{for } i = 2, \dots, n \end{aligned} \tag{3.7}$$

Commanded acceleration of all vehicle can be combined in the following state-space representation

$$u(t) = A_2 e(t) \tag{3.8}$$

matrix A_2 is also defined in Appendix B.

3.2.3 Threat model

FDI attacks against vehicular sensors aim to generate harmful impacts in the platoon by injecting false data into the attacked sensor(s) in order to confuse their measurements.

Existing work has demonstrated that the most-used sensors in automated vehicles, such as LIDAR or cameras, can be jammed or spoofed, and that such attacks can be accomplished at a distance [19, 20, 21, 39]. For the purpose of demonstrating FDI attack impacts in our study, we assume the following: First, the attacker is informed of the platoon model, which includes the controller design and type of sensors used. Second, the attacker has the capability of compromising the reading of one or multiple sensors equipped on one or more platooned vehicles by using drones, units installed on the road for that purpose, or by an attacker-controlled vehicle driving alongside the platoon. Finally, the attack sequence (vector) can only assume discrete values such that once injected it does not violate the resolution of the attacked sensor(s). The last assumption helps create realistic attack scenarios. It also helps distinguish feasible attacks from non feasible ones.

1. Attacking Range Sensors: in this case, the commanded acceleration becomes as follows

$$\begin{aligned} u_1(t) &= k_p\big(e_{x1}(t) + \delta_{x1}(t)\big) + k_d e_{v1}(t) \\ &\vdots \\ u_n(t) &= k_p\big(\big(e_{xn}(t) + \delta_{xn}(t)\big) - e_{xn-1}(t)\big) \\ &\quad + k_d\big(e_{vn}(t) - e_{vn-1}(t)\big) \end{aligned} \tag{3.9}$$

where δ_{xi} is the amount of false-data injected against the i^{th} vehicle's range sensor. (3.9) can be rewritten as follows

$$\begin{aligned} u(t) &= A_2 e(t) + B_{2,x}\delta(t) \\ \delta(t) &= \begin{bmatrix} \delta_{x1}(t) & \dots & \delta_{xn}(t) \end{bmatrix}^T \end{aligned} \tag{3.10}$$

2. Attacking Range-rate Sensors: in this case, the commanded acceleration becomes as follows

$$\begin{aligned} u_1(t) &= k_p e_{x1}(t) + k_d\big(e_{v1}(t) + \delta_{v1}(t)\big) \\ &\vdots \\ u_n(t) &= k_p\big(e_{xn}(t) - e_{xn-1}(t)\big) \\ &\quad + k_d\big(\big(e_{vn}(t) + \delta_{vn}(t)\big) - e_{vn-1}(t)\big) \end{aligned} \tag{3.11}$$

where δ_{vi} is the amount of false-data injected against the i^{th} vehicle's range-rate sensor. (3.11) can be rewritten as follows

$$\begin{aligned} u(t) &= A_2 e(t) + B_{2,v}\delta(t) \\ \delta(t) &= \begin{bmatrix} \delta_{v1}(t) & \dots & \delta_{vn}(t) \end{bmatrix}^T \end{aligned} \tag{3.12}$$

3. Attacking Both Range and Range-rate Sensors: in this case, the commanded acceleration becomes as follows

$$\begin{aligned} u_1(t) &= k_p\big(e_{x1}(t) + \delta_{x1}(t)\big) + k_d\big(e_{v1}(t) + \delta_{v1}(t)\big) \\ &\vdots \\ u_n(t) &= k_p\big(\big(e_{xn}(t) + \delta_{xn}(t)\big) - e_{xn-1}(t)\big) \\ &\quad + k_d\big(\big(e_{vn}(t) + \delta_{vn}(t)\big) - e_{vn-1}(t)\big) \end{aligned} \tag{3.13}$$

which can be rewritten as follows

$$\begin{aligned} u(t) &= A_2 e(t) + B_{2,xv}\delta(t) \\ \delta(t) &= \begin{bmatrix} \delta_{x1}(t) & \dots & \delta_{xn}(t) & \delta_{v1}(t) & \dots & \delta_{vn}(t) \end{bmatrix}^T \end{aligned} \tag{3.14}$$

Matrices $B_{2,x}$, $B_{2,v}$, and $B_{2,xv}$ are given in Appendix B.

Considering the presence of attack vectors, acceleration commands, given in (3.8), become as follows

$$\begin{aligned} u(t) &= A_2 e(t) + B_a \delta(t) \\ B_a &\in \{B_{2,x}, B_{2,v}, B_{2,xv}\} \end{aligned} \tag{3.15}$$

by substituting (3.15) into (3.5), we get

$$\begin{aligned} \dot{e}(t) &= A_c e(t) + B_c \delta(t) \\ A_c &= A_1 + B_1 A_2 \\ B_c &= B_1 B_a \end{aligned} \tag{3.16}$$

3.3 Formulating an Instability-inducing FDI Attack Against Vehicular Platoons

In this section, we will formulate an FDI attack vector that aims to cause instability in the movement of a vehicular platoon. That means, by injecting the formulated attack vector into the targeted sensor(s), the attacked vehicle(s) will begin to behave erratically by accelerating/decelerating and, as a result, relative distance and speed will grow over time and accidents may occur.

3.3.1 Attacks on a single sensor

We will start by formulating an FDI attack vector that could be mounted against either the range or the range-rate sensor equipped on one of the platooned vehicles, whose flawed commanded acceleration is described as follows

$$\begin{aligned} \bar{u}_i(t) &= k_p\big(e_{xi}(t) - e_{xi-1}(t)\big) + k_d\big(e_{vi}(t) - e_{vi-1}(t)\big) + k_a \delta_i(t) \\ &= u_i(t) + k_a \delta_i(t) \end{aligned} \tag{3.17}$$

where k_a is equal to either k_p if the range sensor is targeted or k_d otherwise. On the other hand, for the attacked vehicle the evolution of distance and speed errors over time is

described as follows

$$\begin{aligned}\dot{\bar{e}}_{xi}(t) &= v_{i+1}(t) - v_i(t) \\ &= \bar{e}_{vi}(t) \\ \dot{\bar{e}}_{vi}(t) &= u_{i+1}(t) - \bar{u}_i(t)\end{aligned} \tag{3.18}$$

by substituting (3.17) into (3.18), we get

$$\dot{\bar{e}}_{vi}(t) = u_{i+1}(t) - u_i(t) - k_a\delta_i(t) \tag{3.19}$$

In order to show platoon instability as a result of an FDI attack, we will use the following Lyapunov candidate function

$$V(\bar{e}) = \bar{e}^T(t)P\bar{e}(t) \tag{3.20}$$

where

$$\bar{e}(t) = \begin{bmatrix}\bar{e}_{xi}(t) & \bar{e}_{vi}(t)\end{bmatrix}^T \tag{3.21}$$

and ($P = P^T$) is a symmetric positive definite function described as follows

$$P = \begin{bmatrix}P_1 & P_2 \\ P_2 & P_3\end{bmatrix} \tag{3.22}$$

by differentiating (3.20) we get (time notation is omitted)

$$\dot{V}(\bar{e}) = 2\bar{e}_{xi}P_1\dot{\bar{e}}_{xi} + 2\bar{e}_{vi}P_2\bar{e}_{vi} + 2\big(\bar{e}_{xi}P_2 + \bar{e}_{vi}P_3\big)\dot{\bar{e}}_{vi} \tag{3.23}$$

by substituting (3.18) and (3.19) into (3.23) we get the following

$$\dot{V}(\bar{e}) = 2\bar{e}_{xi}P_1\dot{\bar{e}}_{xi} + 2\bar{e}_{vi}P_2\bar{e}_{vi} + 2\big(\bar{e}_{xi}P_2 + \bar{e}_{vi}P_3\big)\big(u_{i+1} - u_i - k_a\delta_i\big) \tag{3.24}$$

we will define the following attack vector

$$\delta_i = \frac{-1}{k_a}\big(- u_{i+1} + u_i + h_1\bar{e}_{xi} + h_2\bar{e}_{vi}\big) \tag{3.25}$$

by substituting (3.25) into (3.24) we get the following

$$\dot{V}(\bar{e}) = 2\bar{e}_{xi}P_1\dot{\bar{e}}_{xi} + 2\bar{e}_{vi}P_2\bar{e}_{vi} + 2\big(\bar{e}_{xi}P_2 + \bar{e}_{vi}P_3\big)\big(h_1\bar{e}_{xi} + h_2\bar{e}_{vi}\big) \tag{3.26}$$

which can be rewritten as follows

$$\dot{V}(\bar{e}) = \bar{e}^T Q\bar{e} \tag{3.27}$$

where

$$Q = \begin{bmatrix} 2h_1P_2 & P_1 + h_2P_2 + h_1P_3 \\ P_1 + h_2P_2 + h_1P_3 & 2\big(P_2 + h_2P_3\big) \end{bmatrix} \tag{3.28}$$

In order to destabilize the platoon, the attacker must select gains h_1 and h_2 such that $\dot{V}$ is not negative. Therefore, we will select the attack gains such that the matrix Q is positive semi-definite as follows

$$\begin{aligned} 2h_1P_2 &\geq 0 \\ 2\big(P_2 + h_2P_3\big) &\geq 0 \\ 4h_1P_2\big(P_2 + h_2P_3\big) - \big(P_1 + h_2P_2 + h_1P_3\big)^2 &\geq 0 \end{aligned} \tag{3.29}$$

and the attack vector is given by (3.25).

3.3.2 Attacks on two sensors

In this section, we will formulate an FDI attack vector that could be launched against both the range and range-rate sensors equipped on one of the platooned vehicles, whose

commanded acceleration is described as follows

$$\begin{aligned}\bar{u}_i(t) &= k_p\big(e_{xi}(t) - e_{xi-1}(t)\big) + k_d\big(e_{vi}(t) - e_{vi-1}(t)\big) + k_p\delta_{i,1}(t) + k_d\delta_{i,2}(t) \\ &= u_i(t) + k_p a_{i,1}(t) + k_d a_{i,2}(t)\end{aligned} \tag{3.30}$$

by substituting (3.30) into (3.18) we get

$$\dot{\bar{e}}_{vi}(t) = u_{i+1}(t) - u_i(t) - k_p\delta_{i,1}(t) - k_d\delta_{i,2}(t) \tag{3.31}$$

We will use the same candidate Lyapunov function given in (3.20) to derive conditions for platoon instability. Therefore, (3.31) is substituted into (3.23) and we get

$$\dot{V}(\bar{e}) = 2\bar{e}_{xi}P_1\bar{e}_{vi} + 2\bar{e}_{vi}P_2\bar{e}_{vi} + 2\big(\bar{e}_{xi}P_2 + \bar{e}_{vi}P_3\big)\big(u_{i+1} - u_i - k_p\delta_{i,1} - k_d\delta_{i,2}\big) \tag{3.32}$$

we will define the following two attack vectors

$$\begin{aligned}\delta_{i,1} &= \frac{-1}{2k_p}\big(- u_{i+1} + u_i + h_1\bar{e}_{xi} + h_2\bar{e}_{vi}\big) \\ \delta_{i,2} &= \frac{-1}{2k_d}\big(- u_{i+1} + u_i + h_1\bar{e}_{xi} + h_2\bar{e}_{vi}\big)\end{aligned} \tag{3.33}$$

by substituting (3.33) into (3.32) we get the following

$$\dot{V}(\bar{e}) = 2\bar{e}_{xi}P_1\bar{e}_{vi} + 2\bar{e}_{vi}P_2\bar{e}_{vi} + 2\big(\bar{e}_{xi}P_2 + \bar{e}_{vi}P_3\big)\big(h_1\bar{e}_{xi} + h_2\bar{e}_{vi}\big) \tag{3.34}$$

which is similar to the expression given in (3.26). As a result, we can reach the same conclusions for selecting the gains h_1 and h_2 as shown in (3.29).

3.3.3 Finding the attack vector sequence

In this section, we will explain how to find the FDI attack vector sequence such that the resulting vector is realistic in constraints and once injected in the attacked sensor instability is induced in the platoon. We will begin by defining the matrix P as the identity matrix. As a result, (3.29) can be rewritten as

$$\begin{aligned} h_2 &\geq 0 \\ h_1 &\leq -1 \end{aligned} \tag{3.35}$$

Therefore, the attack vectors given in (3.25) and (3.33) will destabilize the platoon if the gains h_1 and h_2 are selected according to (3.35). However, to make the FDI attack more realistic, we still need to consider the following constraints

- the instantaneous values of the FDI attack vector cannot assume any continuous values. That is, the attack sequence will result in spoofed measurements agree with the resolution of the attacked sensor(s). As a result, the instantaneous values are discrete and selected from a predefined range of feasible values.
- once injected, the instantaneous values of the FDI attack vector cannot result in a spoofed measurement that violates the physical bounds of the attacked sensor.
- the instantaneous values of the FDI attack vector will meet the instability condition given in (3.35), as will be explained later.

Since the instantaneous values are selected from a predefined range, it is possible that at some time steps more than one value meet the aforementioned constraints. To handle that, the attack vector sequence will be selected based on minimizing the following cost function

$$J_c = \sum_{k=t}^{t+N_h} \delta_i^T(k) Q \delta_i(k) \tag{3.36}$$

where i is the index of the attacked vehicle, N_h is a time horizon, and $(Q = Q^T > 0)$ is a weighing matrix. Equation (3.36) can be rewritten as

$$\begin{aligned} J_c &= \delta_i^T(t) Q \delta_i(t) + \cdots + a\delta_i^T(t+N_h) Q \delta_i(t+N_h) \\ &= \Delta^T \bar{Q} \Delta \end{aligned} \tag{3.37}$$

where

$$\Delta = \begin{bmatrix} \delta_i(t) & \dots & \delta_i(t+N_h) \end{bmatrix}$$

$$\bar{Q} = \begin{bmatrix} Q & 0 & \dots & 0 \\ 0 & Q & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & Q \end{bmatrix}$$

In order to make the calculations more tractable, and using (3.35), one of the attacker gains, h_1 or h_2, will be assigned a constant value. For example, we can define the following

$$h_1 = -1 \tag{3.38}$$

by substituting (3.38) into (3.25) and rewriting the latter we get the following

$$h_2 = \frac{-1}{\bar{e}_{vi}(t)}\big(k_a\delta_i(t) - u_{i+1}(t) + u_i(t) + \bar{e}_{xi}(t)\big) \geq 0 \tag{3.39}$$

which means that the attack vector sequence must be determined such that the condition given in (3.39) is true. Using the same approach, we can get the following conditions for the attack vectors given in (3.33)

$$h_2 = \frac{-1}{\bar{e}_{vi}(t)}\big(2k_p\delta_{i,1}(t) - u_{i+1}(t) + u_i(t) + \bar{e}_{xi}(t)\big) \geq 0 \tag{3.40}$$

and

$$h_2 = \frac{-1}{\bar{e}_{vi}(t)}\big(2k_d\delta_{i,2}(t) - u_{i+1}(t) + u_i(t) + \bar{e}_{xi}(t)\big) \geq 0 \tag{3.41}$$

In summary, (3.39) defines the instability condition for the case of attacking one sensor while (3.40) and (3.41) define the instability conditions for the case of attacking two sensors. Next, we need to determine the attack vector sequence Δ. For that purpose, we will use

the following Branch and Bound based algorithm

1. Inputs to the algorithm are: the current measurements of the relative distance and speed of the attacked vehicle, the current commanded acceleration of the attacked and preceding vehicles, and a predefined range of feasible values for the attack vector. Output of the algorithm is the attack vector sequence Δ.

2. At the current time step $k = t$, set the following

$$\begin{aligned} J &>> 0 \\ \Delta &= zeros(N_h, 1) \\ J_c &= 0 \text{ (initially)} \\ \Delta_c &= zeros(N_h, 1) \end{aligned} \tag{3.42}$$

3. Get the current measurements of relative distance and speed for the attacked vehicle and use them to calculate $\bar{e}_{xi}(k)$ and $\bar{e}_{vi}(k)$. Also, determine $u_i(k)$ and $u_{i+1}(k)$.

4. Select a candidate value for $\delta_i(k)$ from the predefined range.

5. Depending on the number of attacked sensors, calculate h_2 using either (3.39) or (3.40) and (3.41). Does h_2 satisfy the corresponding instability condition(s)? If yes, then continue. If no, then go to (4).

6. Once injected into the attacked sensor(s), is the spoofed measurement(s) within the bounds of the sensor(s)? If yes, then continue. If no, then go to (4).

7. Store $\delta_i(k)$ in Δ_c and calculate the following

$$J_c = \Delta_c^T \bar{Q} \Delta_c \tag{3.43}$$

8. Is $J_c \leq J$? If yes, then continue. If no, then go to (4), which is the Bound part.

9. Is $k = t + N_h$? If yes, then continue. If no, then go to (12), which is the Branch part.

10. Set the following

$$\begin{aligned} J &= J_c \\ \Delta &= \Delta_c \end{aligned} \tag{3.44}$$

11. Have all the values in the range been tested? If yes, then go to (13). If no, then go to (4).

12. For the next time step $k = t + 1$, determine $\bar{e}_{xi}(k)$ and $\bar{e}_{vi}(k)$ using the model given in (3.18) and use them to calculate $u_i(k)$ and $u_{i+1}(k)$. Then go to (4).

13. We have determined Δ.

The steps for determining the FDI attack vector Δ are given in Algorithm 2. Fig. 3.2 shows simulation results for two attack cases and using the formulated attack vectors given in (3.25) and (3.33). In this simulation, we have a platoon of ten vehicles and an attacker is targeting the range and range-rate sensors of the lead (tenth) vehicle. Before the attack, the platoon is travelling at steady-state. That is, the desired inter-vehicle separation of 5 m and a desired speed of 30 m/s are achieved for all vehicles in the platoon. Also, the attack vector sequence was determined using Algorithm 2. For the results shown in Fig. 3.2a and 3.2c, the range sensor of the lead (tenth) vehicle is targeted with an FDI attack vector given in (3.25) and determined using Algorithm 2. We can see in Fig. 3.2a how the inter-vehicle separation of the attacked vehicle, with respect to the following (ninth) vehicle, is growing larger than the desired separation and then the targeted vehicle collides with the following vehicle at time almost equals to 28 s. This collision happens is because the injected FDI attack vector is misleading the platooning controller of the attacked vehicle and, as a result, the generated control commands are manipulated. We can also see in Fig. 3.2c that the collision happens at a high relative speed of almost -3.5 m/s.

Algorithm 2: Determining an optimal FDI attack vector

Input: for the current time step ($k = t$) the measurements of the relative distance and speed of the attacked vehicle, the current commanded acceleration of the attacked and preceding vehicles, and a predefined range of feasible values $\boldsymbol{a}_r$ for the attack vector.

Output: attack vector sequence Δ.

set $J >> 0$
$\Delta \leftarrow zeros(N_h, 1)$
$J_c \leftarrow 0$
$\Delta_c \leftarrow zeros(N_h, 1)$;
for $i = 1 : |\boldsymbol{a}_r|$ **do**
    $\delta_i(k) \leftarrow \boldsymbol{a}_r(i)$;
    $h_2 \leftarrow$ either (3.39) or (3.40) and (3.41) (depending on the number of attacked sensors) ;
    if *($h_2 \leq 0$)* **then**
        if *($\delta_i(k)$ does not violate the attacked sensor bounds)* **then**
            store $\delta_i(k)$ into Δ_c ;
            $J_c \leftarrow \Delta_c^T \bar{Q} \Delta_c$;
            if *($J_c < J$)* **then**
                if *($k < t + N_h$)* **then**
                    start the algorithm for the next time step ($k = t + 1$) where the inputs $\bar{e}_{xi}(k)$ and $\bar{e}_{vi}(k)$ are calculated using using (3.18) and use them to calculate $u_i(k)$ and $u_{i+1}(k)$;
                else
                    $J \leftarrow J_c$;
                    $\Delta \leftarrow \Delta_c$;

For the results shown in Fig. 3.2b and 3.2d, both the range and range rate sensors of the lead (tenth) vehicle are targeted with FDI attack vectors given in (3.33) and determined using Algorithm 2. Similar to attack case above, we can also see in Fig. 3.2b how the inter-vehicle separation of the attacked vehicle, with respect to the following (ninth) vehicle, is growing larger than the desired separation and then the targeted vehicle collides with the following vehicle at time almost equals to 11.8 s. This collision also happens due to the influence of the two FDI vectors on the platooning controller of the attacked vehicle. We can see, however, in Fig. 3.2d that in this attack case the collision happens at a higher relative speed of almost -6 m/s. In summary, these simulation results show that it is possible to craft an FDI attack vector to target one or two locally equipped sensors of a platooned

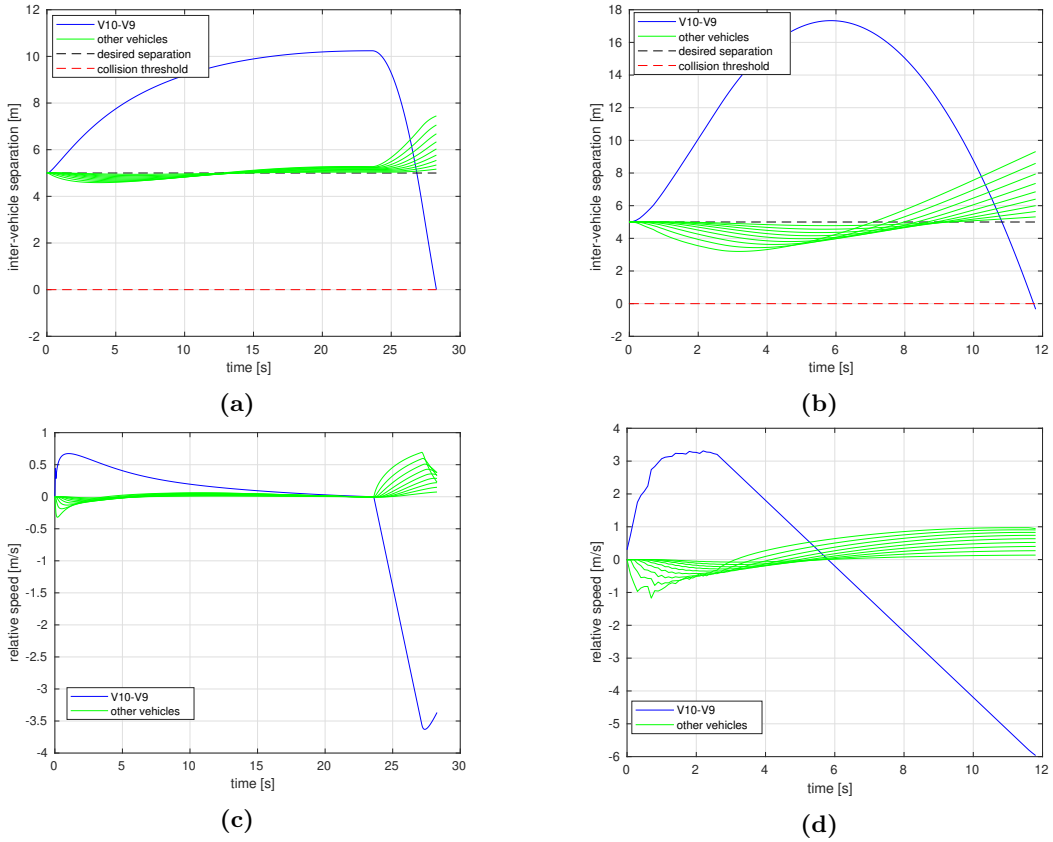


Fig. 3.2. (a) and (c) show inter-vehicle separation and relative speed profiles, respectively, of a ten-vehicle platoon where the lead (tenth) vehicle's range sensors is targeted with an FDI attack vector, calculated using Algorithm 2. (b) and (d) show inter-vehicle separation and relative speed profiles, respectively, of a ten-vehicle platoon where the lead (tenth) vehicle's range and range rate sensors are targeted with FDI attack vectors, both calculated using Algorithm 2.

vehicles and disrupt the formation. Furthermore, these results also show that such FDI attacks are able to induce harmful impacts on the attacked vehicle , such as collisions and at high relative speeds.

3.4 Reachability Analysis For Constrained FDI Attacks

Generally, reachability analysis is a mathematical tool which provides information about the evolution of dynamic system states over time considering that the system may have physical constraints on the control inputs and the states. In this work, we will use this analysis to answer the following question: "Given the attacker capability to manipulate

one or more functionalities of the vehicle's automation system, is it possible to drive the vehicular platoon to an unsafe state (i.e., collisions between two or more vehicles within the platoon)? If so, what is the speed of impact (collision)?"

We use the optimal control based reachability method [34, 35] in order to compute the reachable set of a platoon undergoing an FDI attack. Using this method, the error state space is divided into a number of equidistant target points e_s and for each one of them an optimal control problem is solved to determine whether a feasible trajectory exists between initial states e_0 and the target states e_s. Mathematically, we seek a solution to the following optimization problem

$$\underset{\delta(.)}{\text{minimize}}\ J = \frac{1}{2}||Ce(m) - e_s||_2^2 \tag{3.45}$$

subject to

- initial error states.
- dynamics of the platoon, which are the error states, acceleration (control commands), and the FDI vector.
- constraints on the state, input, targeted sensors, and FDI vector.
- the FDI attack vector is determined such that the increment/decrement of the spoofed measurements is according to the attacked sensor(s) resolution.

The matrix C defines the target vehicle, by selecting its position and velocity errors from the state vector e. The attacker intends to cause a collision with the target vehicle, while the attacked vehicle is where the attacker injects the FDI attack vector. If a solution can be found for (3.45), then there is an attack sequence $\delta(.)$ which can minimize the distance between the final state of the platoon $e(m)$ and e_s, meaning the attacker can cause the platoon to steer towards e_s. If, on the other hand, a solution does not exist, then the attacker cannot drive the platoon to the candidate states e_s.

Since our primary concern is determining the safety of the vehicular platoon while experiencing an FDI attack, we will merely define e_s as the unsafe points in the error state

space, that is, the points wherein the position error is equal to $-x_d$ (for collisions), and for various velocity errors (speed of impact). In order to solve the problem in (3.45) numerically, we need the following formulations

3.4.1 Evolution of errors state vector

For an initial state vector $e(0)$, the error coordinates of the platoon , given in (3.16), will develop over time for $k = 0, 1, \ldots, m$ as follows

$$\begin{aligned} e(1) &= Ae(0) + B\delta(0) \\ e(2) &= Ae(1) + B\delta(1) = A^2e(0) + AB\delta(0) + B\delta(1) \\ &\vdots \\ e(m) &= A^me(0) + A^{m-1}B\delta(0) + \cdots + B\delta(m-1) \end{aligned} \tag{3.46}$$

final error state vector can be rewritten as

$$e(m) = \bar{A}e(0) + \bar{B}\boldsymbol{\delta} \tag{3.47}$$

where

$$\begin{aligned} \bar{A} &= A^m \\ \bar{B} &= \begin{bmatrix} A^{m-1}B & A^{m-2}B & \ldots & B \end{bmatrix} \\ \boldsymbol{\delta} &= \begin{bmatrix} \delta(0) & \delta(1) & \ldots & \delta(m-1) \end{bmatrix}^T \end{aligned}$$

3.4.2 Initial conditions and constraints

We assume that the FDI attack begins once the platoon is at the steady-state, which means both desired separation and relative speed are achieved for all vehicles. Mathematically, the steady-state of the platoon is equivalent to zero position and velocity errors for all vehicles. Besides, in order to create realistic scenarios for the FDI attacks, we define the following constraints

- At any time sample, the attack vector must take a value between a predefined minimum $\delta_{\min}$ and maximum $\delta_{\max}$ values, as shown below

$$\delta(k) \leq \delta_{\max}$$
$$\delta(k) \geq \delta_{\min}$$

which can be rewritten as follows

$$\begin{aligned} \boldsymbol{\delta} &\leq \boldsymbol{\delta}_{\max} \\ \boldsymbol{\delta} &\geq \boldsymbol{\delta}_{\min} \end{aligned} \tag{3.48}$$

where

$$\boldsymbol{\delta}_{\max} = \begin{bmatrix} \delta_{\max} & \dots & \delta_{\max} \end{bmatrix}^T$$
$$\boldsymbol{\delta}_{\min} = \begin{bmatrix} \delta_{\min} & \dots & \delta_{\min} \end{bmatrix}^T$$

- As shown in section 3.2.3, the attack vector has an effect on the calculation of commanded acceleration. Furthermore, each vehicle has physical acceleration limits. For those two reasons, the attack vector must not result in acceleration commands violates a predefined minimum $u_{\min}$ and maximum $u_{\max}$ limits once injected into the attack sensors, as shown below

$$A_2 e(k) + B_2 \delta(k) \leq u_{\max}$$
$$A_2 e(k) + B_2 \delta(k) \geq u_{\min}$$

using (3.46), this constraint can be rewritten as follows

$$\begin{aligned} K_1 e(0) + K_2 \boldsymbol{\delta} &\leq \boldsymbol{u}_{\max} \\ K_1 e(0) + K_2 \boldsymbol{\delta} &\geq \boldsymbol{u}_{\min} \end{aligned} \tag{3.49}$$

where

$$\boldsymbol{u}_{\max} = \begin{bmatrix} u_{\max} & \dots & u_{\max} \end{bmatrix}^T$$
$$\boldsymbol{u}_{\min} = \begin{bmatrix} u_{\min} & \dots & u_{\min} \end{bmatrix}^T$$

- Each sensor has physical limits, that is the reading is always between a minimum $s_{\min}$ and a maximum $s_{\max}$ values. That means, once injected, the attack vector will not result in a spoofed measurement outside the attacked sensor limits, as shown below

$$k_3 e(k) + \delta(k) \leq s_{\max}$$
$$k_3 e(k) + \delta(k) \geq s_{\min}$$

where k_3 is a row vector specifies error states corresponding to the attacked sensors. Using (3.46), this constraint can be rewritten as follows

$$\begin{aligned} K_3 e(0) + K_4 \boldsymbol{\delta} &\leq \boldsymbol{s}_{\max} \\ K_3 e(0) + K_4 \boldsymbol{\delta} &\geq \boldsymbol{s}_{\min} \end{aligned} \tag{3.50}$$

where

$$\boldsymbol{s}_{\max} = \begin{bmatrix} s_{\max} & \dots & s_{\max} \end{bmatrix}^T$$
$$\boldsymbol{s}_{\min} = \begin{bmatrix} s_{\min} & \dots & s_{\min} \end{bmatrix}^T$$

- No collision should be induced in the platoon before reaching the end of attack window (time), as shown below

$$k_5 e(k) + \delta(k) \leq \psi$$

where k_5 is a row vector specifies the position errors in the state vector and ψ is the

collision threshold, which is equal to $-x_d$ in our case. Using (3.46), this constraint is rewritten as follows

$$K_5 e(0) + K_6 \boldsymbol{\delta} \leq \Psi \tag{3.51}$$

where

$$\Psi = \begin{bmatrix} \psi & \dots & \psi \end{bmatrix}^T$$

- Increment/decrement of the FDI attack vector is predefined using a certain resolution. For that reason, the range of possible values for $\delta(k)$ is also predefined and the solution of the problem in (3.45) is set to integer.

All constraints given in (3.48)-(3.51) can be combined in the following compact form

$$A_{ineq} \boldsymbol{\delta} \leq b_{ineq} \tag{3.52}$$

Definitions of K_1, K_2, K_3, K_4, K_5, K_6, A_{ineq}, and b_{ineq} are given in Appendix B.

3.4.3 Computation of FDI reachable sets

The cost function of the problem in (3.45) can be rewritten as

$$\begin{aligned} J &= \frac{1}{2}\left[(Ce(m) - e_s)^T (Ce(m) - e_s)\right] \\ &= \frac{1}{2}\left[e^T(m) C^T C e(m) - 2 e_s^T C e(m) + e_s^T e_s\right] \end{aligned} \tag{3.53}$$

by substituting (3.47) into (3.53), we get

$$J = M_1 \boldsymbol{\delta} + \boldsymbol{\delta}^T M_2 \boldsymbol{\delta} + \text{other terms} \tag{3.54}$$

where

$$M_1 = e^T(0)\bar{A}^T C^T C\bar{B} - e_s^T C\bar{B}$$
$$M_2 = \bar{B}^T C^T C\bar{B}$$

It should be noted that the "other terms" in (3.54) do not include any attack vector sequence and, hence, will be omitted since they do not affect the minimization of J. In summary, for each one of the target states e_s of interest, the reachable set is determined by solving the following

$$\begin{aligned} \min_{\boldsymbol{\delta}} \quad & M_1\boldsymbol{\delta} + \boldsymbol{\delta}^T M_2\boldsymbol{\delta} \\ \text{s.t.} \quad & A_{ineq}\boldsymbol{\delta} \leq b_{ineq} \end{aligned} \tag{3.55}$$

3.5 Results and Discussion

For FDI attacks, the approach explained in Section 3.4 is used to determine the reachable set of the attacked platoon. CPLEX solver was used to determine the integer solution of (3.55). Tables 3.1 and 3.2 show the reachable sets resulting from mounting FDI attacks against one range sensor and range-rate sensor, respectively, equipped on one vehicle in a platoon with ($n = 4$). In each table, $\delta_{x,i}$ and $\delta_{v,i}$ refer, respectively, to the attacked range and range-rate sensors equipped on the i^{th} attacked vehicle, V_i refers to the i^{th} target vehicle in the platoon, specified by C in (3.55), $c_{i,j}$ refers to a collision between the i^{th} and j^{th} vehicles, and the numbers shown in parenthesis are the maximum reachable speed of impact with respect to the two collided vehicles. For these results, resolution of the attacked sensor is selected as 0.5 m and 0.25 m/s^2 for the range and range-rate sensors, respectively. We can see in the aforementioned tables that different impacts can be generated for different scenarios of FDI attacks. For example, attacking the range sensor of the 1^{st} vehicle in the platoon can cause the target vehicle V_1 and the preceding 2^{nd} vehicle to collide at relative speed that could reach -1.955 m/s however attacking the same sensor does not cause any accidents when the target is any vehicle other than the first one V_1 in the platoon, as shown

in the first row of Table 3.1. On the other hand, we can see in the second row of Table 3.2 that attacking the range-rate sensor of the 2^{nd} vehicle in the platoon can cause a crash at the target vehicle V_3 and, in addition, the 2^{nd} and 4^{th} vehicles, even though these last two were not part of the attacker's intention in the first place.

Tables 3.3 and 3.4 show the reachable set resulting from attacking two range sensors and range-rate sensors, respectively, equipped on two vehicles in the same platoon ($n = 4$). In each table, $\delta_{x,ij}$ and $\delta_{v,ij}$ refer, respectively, to the attacked range and range-rate sensors equipped on the i^{th} and j^{th} attacked vehicles. We see clearly that targeting two sensors,

Table 3.1. Reachable set for FDI attacks on a single range sensor

	V_1	V_2	V_3	V_4
$\delta_{x,1}$	$c_{1,2}$ (-1.995)	-	-	-
$\delta_{x,2}$	-	$c_{2,3}$ (-3.996)	-	-
$\delta_{x,3}$	-	-	$c_{3,4}$ (-4.005)	$c_{3,4}$ (-1.848) $c_{4,5}$ (-2.003)
$\delta_{x,4}$	-	-	-	$c_{4,5}$ (-5.987)

Table 3.2. Reachable set for FDI attacks on a single range-rate sensor

	V_1	V_2	V_3	V_4
$\delta_{v,1}$	$c_{1,2}$ (-1.977)	-	-	-
$\delta_{v,2}$	-	$c_{2,3}$ (-2.017)	$c_{2,3}$ (-1.959) $c_{3,4}$ (-1.992) $c_{4,5}$ (-2.210)	-
$\delta_{v,3}$	-	-	$c_{3,4}$ (-2.432)	$c_{3,4}$ (-3.715) $c_{4,5}$ (-2.003)
$\delta_{v,4}$	-	-	-	$c_{4,5}$ (-7.962)

Table 3.3. Reachable set for FDI attacks on two range sensors

	V_1	V_2	V_3	V_4
$\delta_{x,12}$	$c_{1,2}$ (-2.021)	$c_{2,3}$ (-5.997)	$c_{2,3}$ (-1.873) $c_{3,4}$ (-2.004) $c_{4,5}$ (-2.130)	$c_{2,3}$ (-3.666) $c_{3,4}$ (-3.853) $c_{4,5}$ (-3.997)
$\delta_{x,13}$	$c_{1,2}$ (-4.003) $c_{3,4}$ (-3.860)	$c_{1,2}$ (-1.807) $c_{2,3}$ (-2.013)	$c_{3,4}$ (-7.984)	$c_{3,4}$ (-5.743) $c_{4,5}$ (-6.002)
$\delta_{x,14}$	$c_{1,2}$ (-3.988)	$c_{1,2}$ (-1.831) $c_{2,3}$ (-1.987) $c_{3,4}$ (-1.771)	$c_{1,2}$ (-1.468) $c_{2,3}$ (-1.508) $c_{3,4}$ (-1.982)	$c_{4,5}$ (-8.006)
$\delta_{x,23}$	-	$c_{2,3}$ (-4.003) $c_{3,4}$ (-4.008)	$c_{3,4}$ (-7.993)	$c_{3,4}$ (-5.723) $c_{4,5}$ (-6.005)
$\delta_{x,24}$	-	$c_{2,3}$ (-3.996) $c_{4,5}$ (-3.998)	$c_{2,3}$ (-1.833) $c_{3,4}$ (-1.997)	$c_{4,5}$ (-8.997)
$\delta_{x,34}$	-	-	$c_{3,4}$ (-3.566)	$c_{4,5}$ (-8.270)

regardless of type, on two different vehicles, generates a bigger reachable set for the FDI attack and it is possible in some scenarios to cause collisions at greater speeds of impacts when compared to the results of attacking one sensor only.

Tables 3.5 and 3.6 show the reachable set resulting from attacking two range and range-rate sensors equipped on one or two vehicles, respectively, in the same platoon ($n = 4$). Finally, we increased the size of the platoon to ($n = 5$) and determined the reachable set resulting from attacking two range-rate sensors on two different vehicles. Table 3.7 shows the results for these attack cases.

Similarly, we have conducted the same analyses to determine the reachable sets of FDI attacks on a single/double range or range-rate sensors however we neglected the constraint regarding the sensor resolution, which means that $\delta(.)$ can take any continuous values

Table 3.4. Reachable set for FDI attacks on two range-rate sensors

	V_1	V_2	V_3	V_4
$\delta_{v,12}$	$c_{1,2}$ (-8.004)	$c_{2,3}$ (-8.102)	$c_{3,4}$ (-4.193)	-
$\delta_{v,13}$	$c_{1,2}$ (-7.989) $c_{2,3}$ (-7.990)	$c_{2,3}$ (-7.984)	$c_{3,4}$ (-3.970)	-
$\delta_{v,14}$	$c_{1,2}$ (-6.003) $c_{3,4}$ (-5.921)	$c_{1,2}$ (-5.585) $c_{2,3}$ (-5.975) $c_{3,4}$ (-6.138)	$c_{3,4}$ (-7.981)	$c_{4,5}$ (-5.988)
$\delta_{v,23}$	-	$c_{2,3}$ (-5.967)	$c_{3,4}$ (-1.974)	$c_{3,4}$ (-1.982) $c_{4,5}$ (-2.015)
$\delta_{v,24}$	-	$c_{2,3}$ (-5.997) $c_{3,4}$ (-6.224)	$c_{3,4}$ (-7.984)	$c_{4,5}$ (-3.991)
$\delta_{v,34}$	-	-	$c_{3,4}$ (-7.977)	$c_{4,5}$ (-6.011)

Table 3.5. Reachable set for FDI attacks on range & range-rate sensors

	V_1	V_2	V_3	V_4
$\delta_{x,1}$ $\delta_{v,1}$	$c_{1,2}$ (-2.010)	$c_{1,2}$ (-1.866) $c_{2,3}$ (-1.987) $c_{3,4}$ (-2.166)	-	-
$\delta_{x,2}$ $\delta_{v,2}$	-	$c_{2,3}$ (-1.991) $c_{3,4}$ (-2.593)	-	-
$\delta_{x,3}$ $\delta_{v,3}$	-	-	$c_{3,4}$ (-1.799)	$c_{4,5}$ (-1.791)
$\delta_{x,4}$ $\delta_{v,4}$	-	-	-	$c_{4,5}$ (-3.992)

between $\delta_{\min}$ and $\delta_{\max}$. The reason for that was to compare the results with those shown in Tables 3.1 to 3.6. In the case of continuous $\delta(.)$, resulting reachable set was bigger in terms of the number of induced collisions and the magnitude of the speed of impact. However,

Table 3.6. Reachable set for FDI attacks on two range & range-rate sensors

	V_1	V_2	V_3	V_4
$\delta_{x,12}$ $\delta_{v,12}$	$c_{1,2}$ (-7.984)	-	-	-
$\delta_{x,13}$ $\delta_{v,13}$	$c_{1,2}$ (-1.970) $c_{2,3}$ (-2.532)	-	-	-
$\delta_{x,14}$ $\delta_{v,14}$	$c_{1,2}$ (-4.027) $c_{2,3}$ (-4.460) $c_{3,4}$ (-4.701)	$c_{1,2}$ (-1.883) $c_{2,3}$ (-2.004) $c_{3,4}$ (-2.254)	$c_{3,4}$ (-6.011)	$c_{4,5}$ (-3.974)
$\delta_{x,23}$ $\delta_{v,23}$	-	$c_{2,3}$ (-2.015)	-	-
$\delta_{x,24}$ $\delta_{v,24}$	-	$c_{2,3}$ (-7.952) $c_{3,4}$ (-8.118)	$c_{3,4}$ (-3.963)	$c_{4,5}$ (-3.985)
$\delta_{x,34}$ $\delta_{v,34}$	-	-	$c_{3,4}$ (-1.983)	-

Table 3.7. Reachable set for FDI attacks on two range-rate sensors ($n = 5$)

	V_1	V_2	V_3	V_4	V_5
$\delta_{v,12}$	$c_{1,2}$ (-5.221)	$c_{2,3}$ (-8.343)	-	-	-
$\delta_{v,13}$	$c_{1,2}$ (-7.559)	$c_{2,3}$ (-7.401)	$c_{3,4}$ (-3.720)	-	-
$\delta_{v,14}$	$c_{1,2}$ (-7.022) $c_{3,4}$ (-6.631)	$c_{2,3}$ (-6.775) $c_{3,4}$ (-6.138)	$c_{3,4}$ (-6.981)	$c_{4,5}$ (-4.218)	-
$\delta_{v,15}$	$c_{1,2}$ (-5.277)	-	-	-	$c_{5,6}$ (-6.011)
$\delta_{v,23}$	-	$c_{2,3}$ (-5.967)	$c_{3,4}$ (-1.974)	-	-
$\delta_{v,24}$	-	$c_{2,3}$ (-6.127)	$c_{3,4}$ (-5.226)	$c_{4,5}$ (-4.991)	-
$\delta_{v,25}$	-	$c_{2,3}$ (-7.034)	-	-	$c_{5,6}$ (-6.501)
$\delta_{v,34}$	-	-	$c_{3,4}$ (-8.244)	$c_{4,5}$ (-7.935)	-
$\delta_{v,35}$	-	-	$c_{3,4}$ (-7.900)	-	$c_{5,6}$ (-6.113)
$\delta_{v,45}$	-	-	-	-	$c_{5,6}$ (-5.731)

that scenario represents an unrealistic case, as it is not feasible to inject false-data that could take any arbitrary value. In summary, whether it is an attack on a single or multiple sensors, the reachability analysis results shown in the aforementioned tables indicate that the impacts of FDI attacks on the sensors of platooned vehicles are serious and must be considered during the design of platooning controllers which rely on such sensors.

3.6 Conclusion

In this chapter, we focused on FDI attacks that can be mounted against vehicular range and range-rate sensors. Such attacks have been shown to be possible by previous studies and can induce accidents. As an example, we formulated an FDI attack against one or two vehicular sensors that aims to induce instability, i.e., disrupts the platoon formation and eventually could lead to collisions. Furthermore, we considered realistic constraints in formulating our attack, such as a discrete attack vector sequence and non-violation of attacked sensor(s) measurement bounds, in order to produce the most realistic attack scenarios possible. We also employed reachability analysis to further study FDI attacks against vehicular sensors. Our reasoning was that such an analysis would enable us to validate whether it were possible for such attacks to cause collisions by targeting sensors on a larger scale, such as sensors of two vehicles and/or of different types, and at which speed of impact such collisions would be made possible. Our results indicate that FDI attacks are serious and must be considered during the design of platooning controllers, which rely on the measurements of potentially attackable sensors.

REFERENCES

[1] E. Coelingh and S. Solyom, "All aboard the robotic road train," *IEEE Spectrum*, vol. 49, pp. 34–49, 2012.

[2] W. Ren and D. Green, "Continuous platooning: a new evolutionary operating concept for automated highway systems," in *American Control Conference (ACC), 1994*, June 1994.

[3] E. van Nunen, M. R. J. A. E. Kwakkernaat, J. Ploeg, and B. D. Netten, "Cooperative competition for future mobility," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1018–1025, Sep. 2012.

[4] K.-Y. Liang, J. Mårtensson, and K. H. Johansson, "Fuel-saving potentials of platooning evaluated through sparse heavy-duty vehicle position data," *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 1061–1068, 2014.

[5] R. Rajamani, *Vehicle Dynamics and Control*, ser. Mechanical Engineering Series. Springer, 2011. [Online]. Available: https://books.google.com/books?id=eoy19aWAjBgC

[6] B. van Arem, C. J. G. van Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 429–436, Dec 2006.

[7] S. Oncu, J. Ploeg, N. van de Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Network-aware analysis of string stability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1527–1537, Aug 2014.

[8] B. DeBruhl, S. Weerakkody, B. Sinopoli, and P. Tague, "Is your commute driving you crazy?: a study of misbehavior in vehicular platoons," in *Proceedings of the 8th ACM*

Conference on Security & Privacy in Wireless and Mobile Networks. ACM, 2015, p. 22.

[9] R. M. Gerdes, C. Winstead, and K. Heaslip, "Cps: an efficiency-motivated attack against autonomous vehicular transportation," in *Proceedings of the 29th Annual Computer Security Applications Conference*. ACM, 2013, pp. 99–108.

[10] S. Dadras, R. M. Gerdes, and R. Sharma, "Vehicular platooning in an adversarial environment," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, ser. ASIA CCS '15. New York, NY, USA: ACM, 2015, pp. 167–178. [Online]. Available: http://doi.acm.org/10.1145/2714576.2714619

[11] D. D. Dunn, S. A. Mitchell, I. Sajjad, R. M. Gerdes, R. Sharma, and M. Li, "Regular: Attacker-induced traffic flow instability in a stream of semi-automated vehicles," in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, June 2017, pp. 499–510.

[12] A. Petrillo, A. Pescapè, and S. Santini, "A collaborative approach for improving the security of vehicular scenarios: The case of platooning," *Computer Communications*, vol. 122, pp. 59–75, 2018.

[13] M. Amoozadeh, A. Raghuramu, C. Chuah, D. Ghosal, H. M. Zhang, J. Rowe, and K. Levitt, "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 126–132, June 2015.

[14] A. Alipour-Fanid, M. Dabaghchian, H. Zhang, and K. Zeng, "String stability analysis of cooperative adaptive cruise control under jamming attacks," in *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, Jan 2017, pp. 157–162.

[15] Y. Mo and B. Sinopoli, "False data injection attacks in control systems," 2010.

[16] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 21–32. [Online]. Available: http://doi.acm.org/10.1145/1653662.1653666

[17] M. A. Rahman and H. Mohsenian-Rad, "False data injection attacks with incomplete information against smart power grids," in *2012 IEEE Global Communications Conference (GLOBECOM)*, Dec 2012, pp. 3153–3158.

[18] Y. Mo and B. Sinopoli, "False data injection attacks in control systems," *Preprints of the 1st Workshop on Secure Control Systems*, 01 2010.

[19] C. Yan, W. Xu, and J. Liu, "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle," *DEF CON*, vol. 24, 2016.

[20] R. Chauhan, R. M. Gerdes, and K. Heaslip, "Attack against an fmcw radar," in *Proceedings of Embedded Security in Cars Conference*, 2014.

[21] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," *Black Hat Europe*, vol. 11, p. 2015, 2015.

[22] R. W. van der Heijden, T. Lukaseder, and F. Kargl, "Analyzing attacks on cooperative adaptive cruise control (cacc)," *arXiv preprint arXiv:1710.05789*, 2017.

[23] M. Jagielski, N. Jones, C.-W. Lin, C. Nita-Rotaru, and S. Shiraishi, "Threat detection for collaborative adaptive cruise control in connected cars," in *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, ser. WiSec '18. New York, NY, USA: ACM, 2018, pp. 184–189. [Online]. Available: http://doi.acm.org/10.1145/3212480.3212492

[24] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, July 2005.

[25] I. Xausa, R. Baier, M. Gerdts, M. Gonter, and C. Wegwerth, "Avoidance trajectories for driver assistance systems via solvers for optimal control problems," in *20th International Symposium on Mathematical Theory of Networks and Systems*, Melbourne, Australia, 2012, cD-ROM, Paper No. 294, 8 pages,. [Online]. Available: https://hal.inria.fr/hal-00712878

[26] M. Chen, Q. Hu, C. Mackin, J. F. Fisac, and C. J. Tomlin, "Safe platooning of unmanned aerial vehicles via reachability," *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 4695–4701, 2015.

[27] M. Chen, Q. Hu, J. F. Fisac, K. Akametalu, C. Mackin, and C. J. Tomlin, "Reachability-based safety and goal satisfaction of unmanned aerial platoons on air highways," 2016.

[28] A. Kurzhanski and P. Varaiya, "On ellipsoidal techniques for reachability analysis. part i: External approximations," *Optimization Methods and Software*, vol. 17, no. 2, pp. 177–206, 2002.

[29] A. A. Kurzhanskiy and P. Varaiya, "Ellipsoidal techniques for reachability analysis of discrete-time linear systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 1, pp. 26–38, Jan 2007.

[30] Y. Zhou and J. S. Baras, "Reachable set approach to collision avoidance for uavs," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 5947–5952.

[31] S. H. Kafash, J. Giraldo, C. Murguia, A. A. Cardenas, and J. Ruths, "Constraining attacker capabilities through actuator saturation," in *2018 Annual American Control Conference (ACC)*, June 2018, pp. 986–991.

[32] A. M. Bayen, M. Mitchell, M. M. K. Oishi, and C. J. Tomlin, "Aircraft autolander safety analysis through optimal control-based reach set computation," 2006.

[33] M. Chen, S. Bansal, J. F. Fisac, and C. J. Tomlin, "Robust sequential trajectory planning under disturbances and adversarial intruder," *IEEE Transactions on Control Systems Technology*, pp. 1–17, 2018.

[34] R. Baier and M. Gerdts, "A computational method for non-convex reachable sets using optimal control," in *2009 European Control Conference (ECC)*, Aug 2009, pp. 97–102.

[35] Y. E. Arslantaş, T. Oehlschlägel, and M. Sagliano, "Safe landing area determination for a moon lander by reachability analysis," *Acta Astronautica*, vol. 128, pp. 607 – 615, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094576516307846

[36] M. Gerdts and I. Xausa, "Avoidance trajectories using reachable sets and parametric sensitivity analysis," in *System Modeling and Optimization*, D. Hömberg and F. Tröltzsch, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 491–500.

[37] S. Dadras and R. Gerdes, "Reachable set analysis of vehicular platooning in adversarial environment," in *ACC*, June 2018, p. 5568.

[38] D. Yanakiev and I. Kanellakopoulos, "A simplified framework for string stability analysis in ahs," in *Proceedings of the 13th IFAC World Congress, 1996*, 1996, pp. 177–182.

[39] E. Yeh, J. Choi, N. Prelcic, C. Bhat, and R. Heath Jr, "Security in automotive radar and vehicular networks," *submitted to Microwave Journal*, 2016.

CHAPTER 4

MITIGATION OF ATTACKS AGAINST HVAC SYSTEM TEMPERATURE SENSORS USING MOVING TARGET DEFENSE

Heating, Ventilating, and Air Conditioning (HVAC) systems are considered an integral part of smart automated buildings and are primarily employed to provide an acceptable indoor environment in terms of thermal comfort and occupants' air quality. The application of appropriate control strategies in HVAC systems is important to improving the energy efficiency of smart buildings. In this chapter, we utilize Model Predictive Control (MPC) technique to formulate an optimal controller that aims to achieve an acceptable temperature tracking of a desired set point in each zone of the building. To develop such a controller, a model of the process under consideration (the smart building) is needed. For that purpose, we employ a thermal model which captures heat storage and transfer between connected spaces of the building, as well as the influence of outside temperature. Several previous studies have defined potential vulnerabilities in HVAC systems that could be exploited by parties with malicious intentions in order to induce harmful impacts. One possible vulnerability is manipulating the measurements of temperature sensors, which are installed in various sections of buildings employing HVAC systems. An important factor that facilitates such attacks against temperature sensors is the fact that the MPC controller, which uses those sensors, is static in nature, and thus, attackers can easily induce predictable impacts. Therefore, in this chapter, we consider attacks that modify the readings of temperature sensors and show how such tampering could mislead the MPC controller and, as a result, cause occupants' discomfort. Furthermore, in order to counter such attacks, we suggest Moving Target Defense (MTD) technique-based algorithms, which aim to add unpredictability to the system by constantly changing the sensors set used by the MPC controllers and, as a consequence, reduce the impacts of potential attacks.

4.1 Introduction

A smart building is a term used to describe a structure that utilizes automation technology to supervise and control important functionalities/subsystems such as fire and flood safety, lighting, heating/cooling, and building ventilation [1, 2]. Designing a smart building, or even upgrading an existing building to a smart one, requires installing sensors and actuators to manage the relevant subsystems. Additionally, dedicated controllers are implemented to collect and analyze data from sensors and thereby generate the appropriate commands to the actuators of each subsystem. Employing advanced control algorithms is highly critical to reduce energy consumption and operating costs in smart buildings [3]. Smart buildings can provide multiple benefits, such as improved comfort level for the occupants, efficient operation of the building's subsystems, improved life cycle of utilities, customizable spaces, and 24/7 monitoring, all of which result in a boost of productivity.

An HVAC system is an example of a CPS that is commonly employed in smart residential and commercial buildings, as it can provide thermal comfort and consume less energy [4]. For the purpose of designing an appropriate controller that achieves certain goals for HVAC systems, detailed information about the heat dynamics of the building under consideration is needed. Acquiring and employing an accurate building thermal model is also helpful with regard to the decision-making process of the controller, especially if the control strategy is highly dependent on a model of the process under control, such as MPC technique [5, 6]. The thermal model is derived from the physical properties of buildings, e.g. the material used in the building structure, heat storage and loss in each section of the buildings, and interaction between physically-connected spaces. These parameters are tailored to the form of a thermal building model. To achieve a better performance, disturbances such as outside weather, heat provided by presence of occupants, and machines and devices inside each building section should be considered as well.

A number of existing works have demonstrated that it is possible for parties with malicious intentions (attackers) to pose threats against the operation of some CPSs, including threats that could lead to destabilizing and inducing collisions among platooned vehicles

[7, 8] or an unreliable state estimation process in electrical grids [9, 10]. Also, some previous works have suggested defenses to prevent or lessen the effects of such threats [11, 12, 13]. Similarly, in this chapter, we consider the possibility of the presence of attacks intended to influence HVAC sensors and disrupt their performance. Specifically, we will show that manipulating the measurements of temperature sensors by injecting false data and creating incorrect readings could lead to deceiving HVAC systems which rely on those measurements. Such manipulations could lead to raising the temperature in the building, thereby causing occupant discomfort. In addition to potential attacks, incorrect measurements resulting from defective sensors may also mislead the HVAC system, and thereby generate the same harmful impacts mentioned above. Thus, if not detected and repaired, such sensors can pose a threat to the operation of the corresponding HVAC system. With the goal of deterring such threats, in this chapter, we will suggest defenses that aim to constantly change the behaviour of the HVAC system and preemptively reduce the chances of mounting successful attacks.

MTD has been suggested as a countermeasure intended to decrease the ability of an attacker to influence the targeted CPS [14]. From an attacker perspective, such an ability stems from the fact that most, if not all, CPSs are static in their operation, i.e., the components/functionalities required for the CPS's performance, such as sensors, actuators, or communication channels are already assigned and rarely changed. That static nature gives an attacker the necessary time to analyze the targeted CPS and define its weaknesses. MTD attempts to tackle this problem by constantly and unpredictably changing the behavior of the CPS, and thereby adding a dynamic nature. Therefore, MTD is considered a proactive strategy [14]. In the context of control systems, the MTD mechanism utilizes switching among available actuators and/or sensors, such that the attacker's knowledge of the control system becomes uncertain.

Assuming the presence of potential attacks against HVAC-equipped sensors, the main contribution of this chapter is to suggest two MTD-based proactive algorithms, each of which determines a random set of installed sensors to be used for the following two tasks:

First, a partial measurement of the temperatures in some of the building's zones. Second, a prediction (estimation) of the temperatures in the remaining zones. To implement the second task, we will formulate an optimal state observer that primarily relies on the collected data of inputs and outputs and the randomly-selected sensor set. Achieving the two aforementioned tasks guarantees the availability of the data (measurements) required for the operation of the HVAC system. By continuously selecting a random set at each time step, the attacker's ability to induce predictable effects by targeting one or multiple sensors is minimized.

4.1.1 Related work

In this section, we discuss some of the previous studies related to HVAC systems security and MTD mechanism.

1. Security of HVAC systems

   The authors of [15] defined possible vulnerabilities in the automation systems that employ HVAC technology. Such vulnerabilities include the attacker's capability to gain physical access either to the controllers, by guessing the correct password and shutting down the whole system, or to the interconnection between the HVAC's critical components, such as actuators, and, as a result, generating negative impacts. To counter such vulnerabilities, the authors also suggested a neural network-based intrusion detection mechanism. The authors of [16] have noted another vulnerability in HVAC systems, the inaccurate measurements in the temperature or air flow rate sensors. By exploiting such vulnerability, targeted sensors could produce measurements that are either below (negative bias) or above (positive bias) the real value of the measured quantities. Similarly, a wavelet neural work was also suggested and trained in order to diagnose faulty sensor(s).

   In [17], several threats against HVAC systems were defined, including manipulating of the set points, sensors feeding either a constant false measurement or varying measurements within the bounds of the sensor measurements, or sending harmful commands

to the actuators. In the same work, a system model-based detection method was suggested. Similar to the above mentioned works, in this chapter we will focus on potential attacks against the temperature sensors of HVAC systems that result in incorrect measurements. Furthermore, we also suggest countermeasures to reduce the impacts generated by such attacks.

2. MTD-based countermeasures

Initially, MTD was suggested for applications in the area of computer and network security [18]. For instance, the authors of [19] proposed an MTD based algorithm to protect the privacy of IPV6 users by repeatedly changing the addresses belonging to the sender and receiver, and thus, leaving the attacker unable to identify the two communication hosts. Similarly, the authors of [20] developed an MTD-based strategy for mutating the IP address to create high chances of unpredictability while maintaining the original configuration of the address. Also, MTD-based techniques have been applied in the area of smart grids protection [13, 21, 22], specifically for the process of state estimation which is critically important to ensure a reliable operation of the electric grid.

MTD has also been applied in the context of control theory. In [23], the authors implemented an MTD-based mechanism that introduced additional states to the control system, with time-varying dynamics that can be measured using additional sensors. These new states are difficult to identify by the attacker, and thus, the latter fails to design stealthy attacks. Also, the authors of [24] considered Denial-of-Service (DoS) attacks on control systems that can cease control commands. To prevent such attacks, the authors proposed an MTD-based mechanism that randomly switches among multiple controllers, such that alternate control commands are available to replace the ceased ones, and the operation of the control system is not disrupted.

Close to our work, the authors of [25] proposed an MTD-based mechanism that randomly switches among multiple LQR controllers or sensors, such that unpredictability of the control system is increased. The switching is based on maximizing the entropy

produced and minimizing the control cost. The authors also suggested a detection mechanism in order to identify the attacked actuating/sensing mode and take it offline. On the other hand, the authors of [26] used an MTD-based mechanism that randomly switches between different controllers or sensors. However, the switching is based upon solving a formulated optimization problem that minimizes the attack impacts. Similar to the aforementioned works, in this chapter, we also propose MTD-based countermeasures that aim to reduce the impacts of attacking the temperature sensors of HVAC systems. Our algorithms randomly switch among the installed sensors, such that a set of them is selected, and an estimation is obtained for the non-measured temperatures. The switching in both of our algorithms relies mainly upon minimizing the deviation in state (temperature) estimation.

4.1.2 Organization

Section 4.2 explains the thermal model derived for the smart building. Section 4.3 explains the formulation of an MPC based controller for temperature tracking. Section 4.4 describes the threat model against temperature sensors of HVAC systems. In Section 4.5, we discuss the formulation of an optimal state observer, the two suggested MTD based algorithms, and the simulation results. Conclusions are given in Section 4.6.

4.2 Thermal Model

For the purpose of designing an appropriate controller that achieves certain goals for HVAC systems, detailed information about the heat dynamics of the building under consideration is needed. Acquiring and employing an accurate building thermal model is helpful with the decision-making process of the controller, especially if the control strategy is highly dependent upon a model of the process under control, such as the MPC technique. A thermal model is derived from the physical properties of buildings, e.g., the material used in the building structure, heat storage and loss in each section of the building, and interaction between physically connected spaces. Furthermore, disturbances such as outside weather and the ground temperature are considered as well.

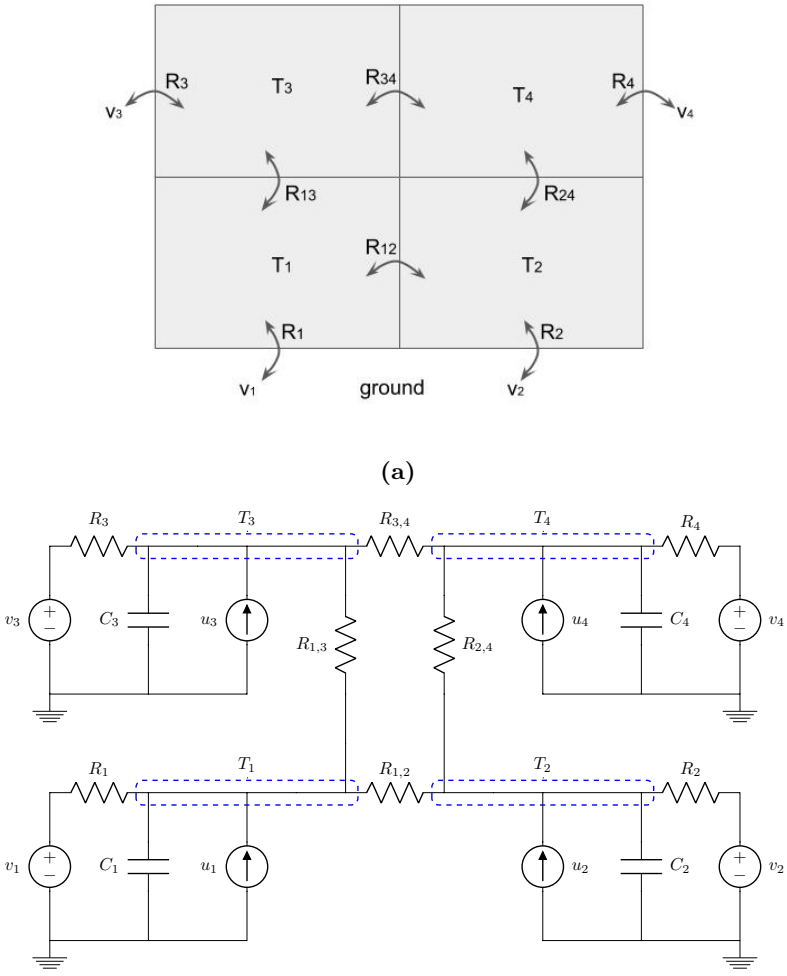


Fig. 4.1. a) A side view of a two-floor building with two zones on each floor. T_i, for $i = 1, \ldots, 4$, is the temperature of each zone. b) RC network representation of the building shown in Fig. 4.1a.

In this chapter, we use the thermal model developed in [27, 28], using the grey-box approach. In this model, the building is divided into a number of zones, wherein each zone is represented as an *RC* electrical circuit. For the simulation results presented in this chapter, we consider a building with two floors and two zones per floor, as shown in Fig. 4.1a and its *RC* equivalent circuit in Fig. 4.1b. In the *RC* representation of the building, R represents the thermal conduction between any two connected (neighboring) zones, C represents the thermal capacity in each zone, T_i is the i^{th} zone's measured temperature, and u_i is the

input, calculated by the MPC controller and supplied to the i^{th} zone. Furthermore, this model also considers the sun temperatures v_3 and v_4, conducted through the windows to the 3^{rd} and 4^{th} zones, and the ground temperatures v_1 and v_2, also conducted to the 1^{st} and 2^{nd} zones.

In this study, we assume that each zone in the building is equipped with a Variable Air-flow Volume (VAV), which is a terminal box that provides conditioned air [29]. The conditioned air is provided to each zone with a specific air flow rate that can be selected from multiple predefined discrete levels [30]. Each VAV unit consists of a damper, which regulates the air flow rate, and a heating coil, to raise the supplied air temperature if needed [31]. Therefore, the control input u will be calculated using our MPC based controller from a predefined discrete range, as will be explained in Section 4.3.

By using Nodal analysis, the heat dynamics of each zone in the building can be described in first-order differential equations. For example, the heat dynamics of the 1^{st} zone in Fig. 4.1b can be described as follows

$$C_1\dot{T}_1(t) = \frac{1}{R_1}\left(v_1(t) - T_1(t)\right) + \frac{1}{R_{12}}\left(T_2(t) - T_1(t)\right) + \frac{1}{R_{13}}\left(T_3(t) - T_1(t)\right) + u_1(t) \tag{4.1}$$

Similarly, the same analysis can be used to write the heat dynamics for the other zones. As a result, the thermal model of the whole building can be described using the following state space representation

$$\begin{aligned} \dot{x}(t) &= A_c x(t) + B_c u(t) + G_c w(t) \\ y(t) &= C x(t) \end{aligned} \tag{4.2}$$

where $x \in \mathbb{R}^{n\times 1}$, $u \in \mathbb{R}^{p\times 1}$, $w \in \mathbb{R}^{n\times 1}$, and $y \in \mathbb{R}^{n\times 1}$ (with $n = p = 4$) refer to the state, input, disturbance, and output vectors, respectively, defined as follows

$$\begin{aligned} x(t) &= \begin{bmatrix} T_1(t) & T_2(t) & T_3(t) & T_4(t) \end{bmatrix}^T \\ u(t) &= \begin{bmatrix} i_1(t) & i_2(t) & i_3(t) & i_4(t) \end{bmatrix}^T \\ w(t) &= \begin{bmatrix} v_1(t) & v_2(t) & v_3(t) & v_4(t) \end{bmatrix}^T \\ y(t) &= \begin{bmatrix} T_1(t) & T_2(t) & T_3(t) & T_4(t) \end{bmatrix}^T \end{aligned} \tag{4.3}$$

also $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times p}$, $G \in \mathbb{R}^{n\times n}$, and $G \in \mathbb{C}^{n\times n}$ are system matrices defined as follows

$$A_c = \begin{bmatrix} \frac{-1}{C_1}(\frac{1}{R_1} + \frac{1}{R_{12}} + \frac{1}{R_{13}}) & \frac{1}{C_1 R_{12}} & \frac{1}{C_1 R_{13}} & 0 \\ \frac{1}{C_2 R_{12}} & \frac{-1}{C_2}(\frac{1}{R_2} + \frac{1}{R_{12}} + \frac{1}{R_{24}}) & 0 & \frac{1}{C_2 R_{24}} \\ \frac{1}{C_3 R_{13}} & 0 & \frac{-1}{C_3}(\frac{1}{R_3} + \frac{1}{R_{13}} + \frac{1}{R_{34}}) & \frac{1}{C_3 R_{34}} \\ 0 & \frac{1}{C_4 R_{24}} & \frac{1}{C_4 R_{34}} & \frac{-1}{C_4}(\frac{1}{R_4} + \frac{1}{R_{24}} + \frac{1}{R_{34}}) \end{bmatrix}$$

$$B_c = \begin{bmatrix} \frac{1}{C_1} & 0 & 0 & 0 \\ 0 & \frac{1}{C_2} & 0 & 0 \\ 0 & 0 & \frac{2}{C_3} & 0 \\ 0 & 0 & 0 & \frac{1}{C_4} \end{bmatrix}$$

$$G_c = \begin{bmatrix} \frac{1}{C_1 R_1} & 0 & 0 & 0 \\ 0 & \frac{1}{C_2 R_2} & 0 & 0 \\ 0 & 0 & \frac{2}{C_3 R_3} & 0 \\ 0 & 0 & 0 & \frac{1}{C_4 R_4} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(4.4)

Finally, the model given in (4.2) can be discretized, using the forward difference approximation [32], into the following discrete-time system (building) model

$$x(k+1) = Ax(k) + Bu(k) + Gw(k) \tag{4.5}$$

4.3 Formulating an MPC based Controller for Temperature Tracking

In this section, we will formulate an MPC based controller with the goal to regulate the temperature at each zone in the building a reference (desired) temperature. For this purpose, we will consider the model given in (4.4). To ensure an offset-free tracking in the presence of model uncertainty and unmeasured disturbances, we will add integral action to the controller. One method for incorporating an integrator in the MPC controller framework is to modify the plant model such that the input is the control difference $\Delta u(k)$, instead of $u(k)$ [33, 34]. This is achieved by taking the difference of both sides of (4.5) as follows

$$\Delta x(k+1) = A\Delta x(k) + B\Delta u(k) + G\Delta w(k) \tag{4.6}$$

where

$$\begin{aligned} \Delta x(k) &= x(k) - x(k-1) \\ \Delta u(k) &= u(k) - u(k-1) \\ \Delta w(k) &= w(k) - w(k-1) \end{aligned} \tag{4.7}$$

The suggested MPC controller determines $\Delta u(k)$ which minimizes the following cost function

$$J = \sum_{l=k}^{k+N-1} \left\{ \left[x_r(l+1) - x(l+1) \right]^T Q \left[x_r(l+1) - x(l+1) \right] + \Delta u^T(l) R \Delta u(l) \right\} \tag{4.8}$$

where $x_r \in \mathbb{R}^{n\times 1}$ refers to the reference state vector, and $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{p\times p}$ are weighing matrices. We can rewrite (4.8) as follows

$$J = (X_r - X)^T \bar{Q}(X_r - X) + \Delta U^T \bar{R} \Delta U \tag{4.9}$$

where

$$\begin{aligned} X_r &= \begin{bmatrix} x_r(k+1) & x_r(k+2) & \dots & x_r(k+N) \end{bmatrix}^T \\ X &= \begin{bmatrix} x(k+1) & x(k+2) & \dots & x(k+N) \end{bmatrix}^T \\ \Delta U &= \begin{bmatrix} \Delta u(k) & \Delta u(k+1) & \dots & \Delta u(k+N-1) \end{bmatrix}^T \end{aligned} \tag{4.10}$$

from (4.7), we can rewrite $\Delta x(k)$ as follows

$$x(k+1) = \Delta x(k+1) + x(k) \tag{4.11}$$

by substituting (4.5) into (4.11), we can write the following

$$\begin{aligned} x(k+1) &= \Delta x(k+1) + x(k) \\ &= A\Delta x(k) + B\Delta u(k) + G\Delta w(k) + x(k) \\ &\vdots \\ x(k+N) &= x(k) + (A^{k+N} + A^{k+N-1} + \dots + A)x(k) \\ &\quad + (A^{k+N-1}B + \dots + B)\Delta u(k) + \dots \\ &\quad + (A^{k+N-2}B + \dots + B)\Delta u(k+1) + \dots \\ &\quad + B\Delta u(K+N-1) \\ &\quad + (A^{k+N-1}G + \dots + G)\Delta w(k) + \dots \\ &\quad + (A^{k+N-2}G + \dots + G)\Delta w(k+1) + \dots \\ &\quad + G\Delta w(K+N-1) \end{aligned} \tag{4.12}$$

which can be rewritten in the following compact form

$$X = M_1 x(k) + M_2 \Delta x(k) + M_3 \Delta U + M_4 \Delta W \tag{4.13}$$

where

$$\Delta W = \begin{bmatrix} \Delta w(k) & \Delta w(k+1) & \dots & \Delta w(k+N-1) \end{bmatrix}^T \tag{4.14}$$

Matrices $\bar{Q}$, $\bar{R}$, M_1, M_2, M_3, and M_4 are defined in Appendix C. By substituting (4.13) into (4.9), we get the following

$$J = \Delta U^T H_c \Delta U + f_c \Delta U + \text{other terms} \tag{4.15}$$

where

$$\begin{aligned} H_c &= M_3^T \bar{Q} M_3 \\ f_c &= 2\big(\big[x^T(k) M_1^T + \Delta x^T(k) M_2^T \\ &\quad + \Delta W^T M_4^T - X_r^T \big] \bar{Q} M_3 \big) \end{aligned} \tag{4.16}$$

it should be noted that the "other terms" in (4.15) do not include the term ΔU and, hence, shall be omitted. Therefore, the MPC controller determines the control sequence ΔU by solving the following optimization problem

$$\underset{\Delta U}{\text{minimize}}\ J(k) = \Delta U^T H_c \Delta U + f_c \Delta U \tag{4.17}$$

the current control input $u(k)$ is determined by using $\Delta u(k)$, which is obtained from ΔU, and the previous input as follows

$$u(k) = \Delta u(k) - u(k-1) \tag{4.18}$$

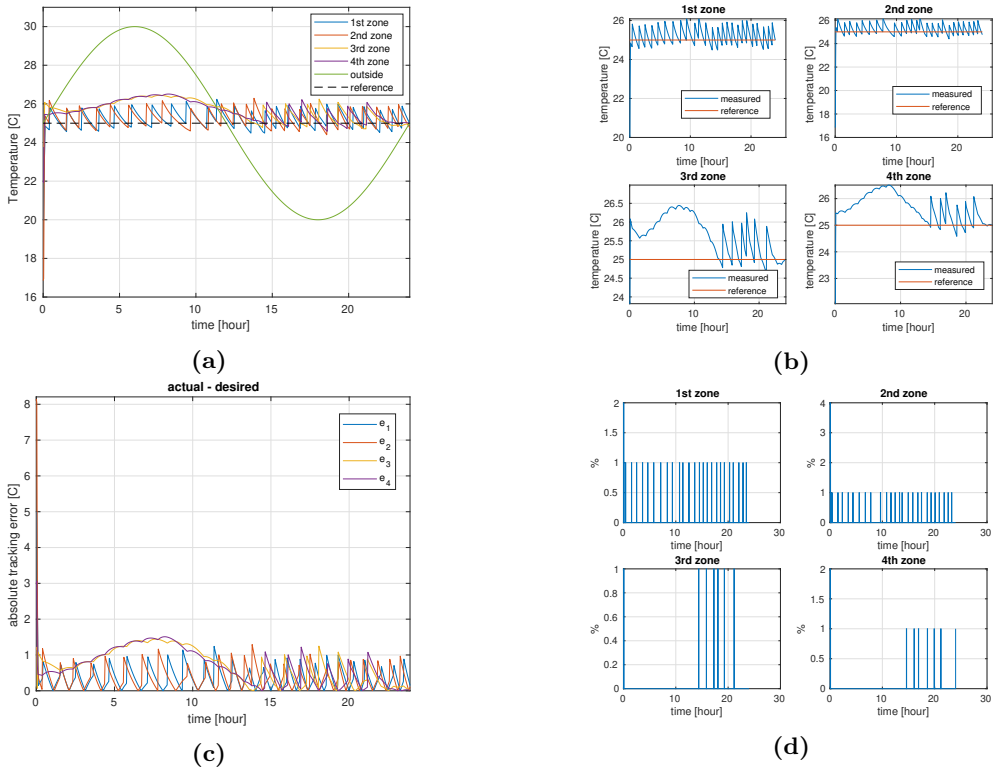
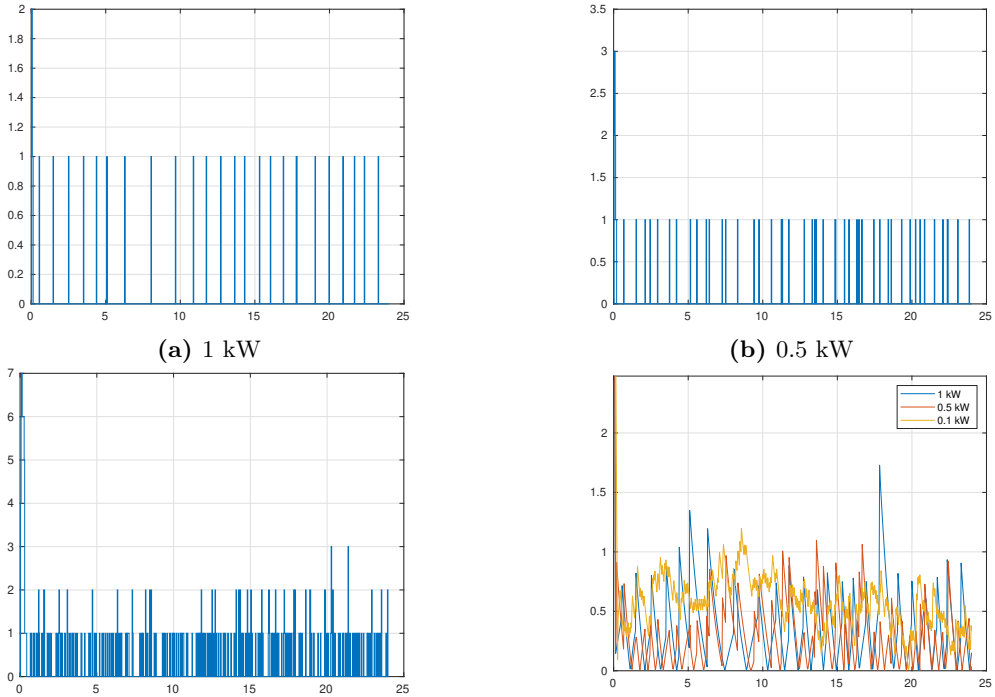


Fig. 4.2. a) Temperature profiles of the outside and four zones in a smart building where our MPC controller is tracking a desired temperature of 25C. b) Measured and reference temperature profiles of each zone. c) The absolute temperature tracking error of each zone. d) Control input of each zone.

4.3.1 Results

For the results presented in this section, we used MATLAB to simulate a building wherein $n = 4$ and (number of zones) and $p = 4$ (number of control inputs), using the system thermal model developed in Section 4.2. The MPC controller regulates the temperature in each zone to a desired value of 25C and the simulation time is equal to twenty four hours.

Fig. 4.2 shows simulation results using the MPC-based controller to regulate the temperature of four zones inside a building wherein the controller is aiming to keep the temperatures close to or around 25C (desired setting) by providing heating only. For these results, the simulation time is a one-day period. Fig. 4.2a shows the temperature profiles of the four zones, the outside of the building, and the reference (desired setting) during the simulation time. We can see that the controller is able to keep the indoor temperatures



(a) 1 kW **(b)** 0.5 kW **(c)** 0.1 kW **(d)** 1st zone tracking error

Fig. 4.3. The control input discrete sequences resulting from using an input energy of a) 1kW, b) 0.5kW, and c) 0.1 kW. d) The absolute temperature tracking error resulting from using the shown input sequences.

close to the reference. Fig. 4.2c shows the absolute tracking error in the four zones. It is clear that the controller manages to keep the tracking errors around 1C. However, we can also see that e_3 and e_4, the tracking errors of the third and fourth zones, respectively, exceed 1C, particularly within the time before 12 hours. This is because of the effect of the outside temperature on the above-mentioned zones. As we can see in Fig. 4.2a, the outside temperature fluctuates throughout the day, and heat propagates from the outside to the third and fourth zones. As a result, propagated heat manages to raise the temperature above the reference. For that reason, the controller does not generate control commands for those zones during the first 12 hours, as is shown in Fig. 4.2d. Fig. 4.2b shows the measured and reference temperature profiles in the four zones. We can see in Fig. 4.2d that the controller generates specific non-zero discrete commands to provide heating specifically for the time instances when the measured temperature is below the reference.

In Fig. 4.3, we focus on the temperature response of the 1^{st} zone in the building. As

was explained in Section 4.2, the control input u is selected from predefined discrete levels, which are obtained as a percentage of a certain energy. Therefore, three values for the input energy are defined and tested with the proposed MPC controller. Fig. 4.3a, 4.3b, and 4.3c show the percentage of air mass flow supplied to the 1^{st} zone as a result of using an input energy of 1kW, 0.5kW, and 0.1KW, respectively. We can see that the input always settles at a level of 1%, indicating less energy consumed by the controller. However, we can also see that the time needed to keep the controller turned on becomes less as the predefined input energy increases. Furthermore, regardless of the discrete levels the controller uses, we can see in Fig. 4.3d that it is possible to achieve the desired temperature for the 1^{st} zone with an acceptable absolute tracking error which is less than 2C at the worst.

4.4 Threat Model Against HVAC Systems

Generally, an HVAC system, as a CPS, employs optimal controllers which require the integrating and managing of various subsystems/functionalities. For instance, communication protocols are utilized to connect multiple components of the HVAC system and relay data to the controllers. Also, a diverse network of sensors, such as temperature or air pressure sensors, is installed at different parts of the smart building in order to perceive and monitor the condition either inside or outside the building. Obtaining accurate measurements from the installed sensors is particularly important in order to ensure both a reliable operation by the optimal controllers and the generation of correct commands. At the same time, employing the aforementioned functionalities creates potential vulnerabilities in the HVAC systems that can be exploited by attackers intending to disrupt the normal (proper) operation [35]. For example, these attacks could be the exchange of false or modified information [36] or the manipulating of sensor-measured values to modify the generated control commands [37], also referred to as false-data injection attacks.

In the context of CPS security, the operation of HVAC systems in the presence of possible threats against the interconnection among the system's components has been studied [15]. Furthermore, the inaccurate measurements of sensors deployed for HVAC systems has also been identified as a vulnerability. For instance, the installed sensors may constantly

feed positive or negative biases, i.e., temperature measurements that are larger or smaller, respectively, than the real values [16] or the sensors may feed falsified, yet still seemingly realistic, readings [17]. Therefore, in this chapter, we also consider possible scenarios of attacks intending to manipulate the temperature sensor(s) measurements of HVAC systems, such that the controllers using those measurements are misled.

To demonstrate the impacts of attacks against HVAC-sensing functionality in our study, we make the following two assumptions: First, the attacker is capable of targeting one or multiple sensors installed in a smart building. Second, the mounted attack(s) succeeds in manipulating the readings of the targeted sensor(s). We consider three possible types of attacks against HVAC sensors, namely, a negative bias attack, wherein the manipulated measurement is constant and below the real sensed temperature, a sine-wave attack, wherein the attacker is injecting within-the-bounds sinusoidal-like data into the targeted sensor(s), and a random-value attack, wherein the attacker is also injecting within-the-bounds data into the targeted sensor(s), but such data is randomly selected from the range within the sensor bounds. To assess the impacts induced by the aforementioned attacks, we will define and use the following metric

$$A_{impact} = \frac{1}{T_a} \sum_{k=t_0}^{t_0+T_a} \big[x(k) - x_r(k)\big]^T \big[x(k) - x_r(k)\big] \tag{4.19}$$

where t_0 refers to the initial time instance when the attack begins, T_a is the length of attack time, and x_r is the vector of desired temperature. In short, A_{impact} represents the mean squared error of each zone's temperature with respect to x_r.

Fig. 4.4a shows A_{impact} calculated both for an attack-free case, and when targeting one or multiple temperature sensors with the attacks defined in the legend. Regardless of the type of attack, it is clear that more severe impacts are induced as the number of targeted sensors increases. Fig. 4.4b shows the simulated temperature profiles of a specific case of negative bias attack, wherein the first zone sensor is feeding a constant false reading of 22C. We can see how the temperature of the first (targeted) zone rises, due to misleading the controller by the falsified measurement. Furthermore, we can also see the attack impact

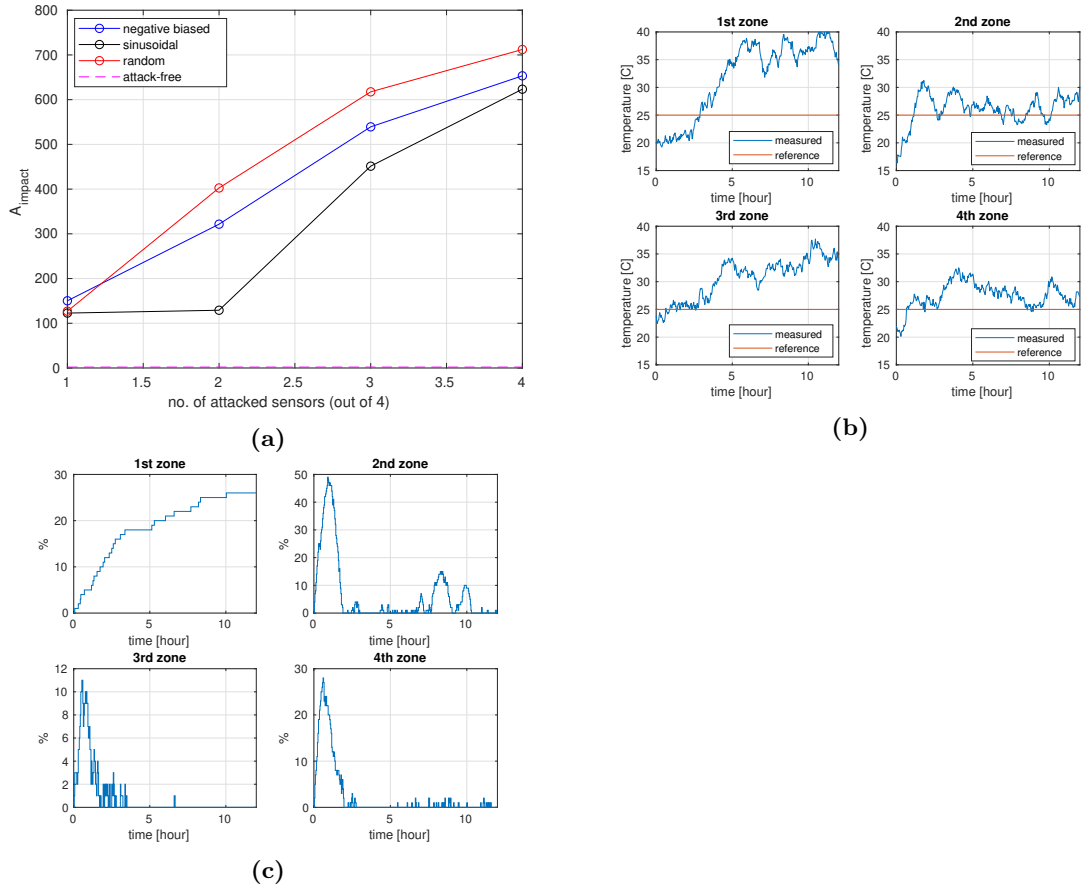


Fig. 4.4. (a) Attack impact (A_{impact}) calculated for different attack cases. b) Measured and reference temperature profiles of four zone in a building where the sensor of the first zone is targeted with a negative bias attack. c) Control input profiles of the four zones generated by the MPC controller in response to the attack case given in (b).

on the fourth zone temperature, as it also rises, despite the fact that zone sensor was not attacked. Fig. 4.4c shows the controller actions for the same attack case. The attack impact is clearly seen in the control input of the targeted zone, as it is increasing over time, which indicates a greater energy consumption. For this attack case, A_{impact} is found equal to 149.253.

4.5 MTD-based Algorithms

In order to increase the uncertainty of the attacker's knowledge about the HVAC system structure, particularly the operation of the MPC controller, we suggest two MTD algorithms each of which constantly changes the set of sensors selected to produce an estimation of the

system states, thereby adding unpredictability to the system, and, as a result, reducing the impacts of any potential attacks against one or multiple sensors installed in the building.

4.5.1 State estimation using an optimal observer

The goal of employing an MTD-based strategy with the designed MPC controller is to increase the uncertainty of the attacker's knowledge about the HVAC system structure, and, as a consequence, reduce the impacts of potential attacks against the system sensors. We suggest a way to achieve that goal by constantly changing the set of sensors with measurements fed to the MPC controller. In the baseline system structure, we assume that each zone of the smart building is equipped with at least one temperature sensor. Furthermore, we also assume, in Section 4.4, that one or more of those sensors could be potentially compromised. Therefore, instead of using measurements from all of them, a subset of those sensors is selected randomly, and its readings used both to perform state estimation in order to provide data for the remaining unmeasured temperatures and to use that data to generate control commands.

State estimation is a process that provides knowledge about system states which cannot be measured or determined directly on the basis of available sensor measurements, a model of that system, and previous inputs/outputs. Furthermore, an observer is a circuit block or a computer software capable of performing the state estimation process. An important factor to consider when it comes to state estimation is that the available sensor measurements, along with the system model, can render the system fully observable, which means it is possible to reconstruct all unmeasured states [38]. As a part of our suggested MTD-based algorithms, we will utilize the state estimation process to determine the temperatures for which sensors have not been selected during the implementation of the algorithms. For that purpose, we will formulate an optimal state observer as follows

1. Formulation of the optimal observer

Our formulated observer uses previously collected data to estimate the current temperatures. The estimated system dynamics can be described as follows

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + Gw(k) \\ \hat{y}(k) &= C\hat{x}(k) \end{aligned} \tag{4.20}$$

where $\hat{x}$ and $\hat{y}$ refer to the estimated state and output vectors, respectively. If A is nonsingular, as is the case with our system model, then we can write

$$\hat{x}(k) = A^{-1}\hat{x}(k+1) - A^{-1}Bu(k) - A^{-1}Gw(k) \tag{4.21}$$

using (4.21), we can describe the past state vector over a time horizon N recursively as follows

$$\begin{aligned} \hat{x}(k-1) &= A^{-1}\hat{x}(k) - A^{-1}Bu(k-1) - A^{-1}Gw(k-1) \\ &\vdots \\ \hat{x}(k-N) &= A^{-N}\hat{x}(k) - \cdots - A^{-1}Bu(k-N) \\ &\quad + A^{-N}Gw(k-1) - \cdots - A^{-1}Gw(k-N) \end{aligned} \tag{4.22}$$

using (4.22), we can describe the past output vector over N recursively as follows

$$\begin{aligned} \hat{y}(k-1) &= C\hat{x}(k-1) \\ &= CA^{-1}\hat{x}(k) - CA^{-1}Bu(k-1) \\ &\quad - CA^{-1}Gw(k-1) \\ &\vdots \\ \hat{y}(k-N) &= C\hat{x}(k-N) \\ &= CA^{-N}\hat{x}(k) - \cdots - CA^{-1}Bu(k-N) \\ &\quad + CA^{-N}Gw(k-1) - \cdots - CA^{-1}Gw(k-N) \end{aligned} \tag{4.23}$$

which can written as follows

$$\hat{Y} = \bar{A}\hat{x}(k) + \bar{B}U + \bar{G}W \tag{4.24}$$

where

$$\begin{aligned} \hat{Y} &= \begin{bmatrix} \hat{y}(k-N) & \hat{y}(k-N+1) & \dots & \hat{y}(k-1) \end{bmatrix}^T \\ U &= \begin{bmatrix} u(k-N) & u(k-N+1) & \dots & u(k-1) \end{bmatrix}^T \\ W &= \begin{bmatrix} w(k-N) & w(k-N+1) & \dots & w(k-1) \end{bmatrix}^T \end{aligned} \tag{4.25}$$

Matrices $\bar{A}$, $\bar{B}$, and $\bar{G}$ are defined in Appendix C. The squared error between measured and estimated outputs can be defined as follows

$$J(k) = \big(Y - \hat{Y}\big)^T \big(Y - \hat{Y}\big) \tag{4.26}$$

where Y is a vector of N collected previous outputs. By substituting (4.24) into (4.29) we get

$$J(k) = \hat{x}(k) H_e \hat{x}(k) + f_e \hat{x}(k) \tag{4.27}$$

where

$$\begin{aligned} H_e &= \bar{A}^T \bar{A} \\ f_e &= 2\big([U^T \bar{B}^T + W^T \bar{G}^T - Y^T]\bar{A}\big) \end{aligned} \tag{4.28}$$

In summary, the objective of our formulated optimal state observer is to estimate the current states (temperatures) vector $\hat{x}(k)$ by solving the following optimization problem

$$\underset{\hat{x}(k)}{\text{minimize}}\ J(k) = \hat{x}(k) H_e \hat{x}(k) + f_e \hat{x}(k) \tag{4.29}$$

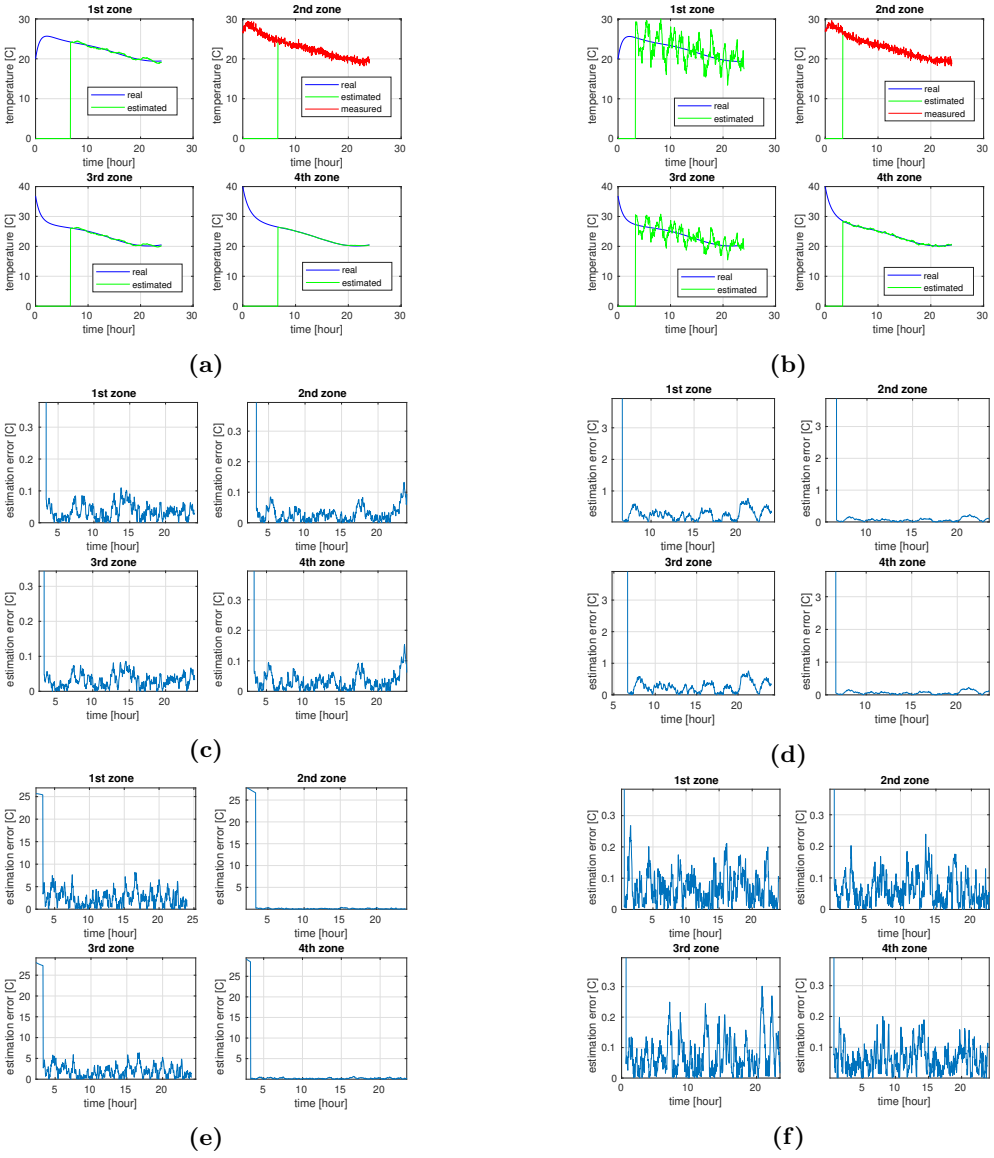


Fig. 4.5. a) Real and estimated temperature profiles using the optimal observer with 200 previous readings from the second zone's sensor. b) Real and estimated temperature profiles using the optimal observer with 100 previous readings from the second zone's sensor. c) Absolute estimation error for an observer using 100 previous from the second and third zones' sensors. d) Absolute estimation error for the results shown in (a). e) Absolute estimation error for the results shown in (b). f) Absolute estimation error for an observer using 30 previous from the second, third, and fourth zones' sensors.

2. Estimation results

Fig. 4.5 shows various simulation results using the optimal observer formulated in Section 1 and for different cases. Fig. 4.5a shows the results for estimating the

temperatures of the four zones in the building, whereby the observer is using the measurements from a single sensor installed in the second zone. For this case, the number of collected data samples (N_e) is equal to 200. Given that the sampling frequency is equal to 1/120 Hz, collecting 200 data samples requires about six and a half hours, which is when the estimation begins (green line) as shown in the same figure. For the same case, the estimation error, or the absolute difference between real and estimated temperatures, is shown in Fig. 4.5d. We can see clearly in both Figures mentioned above that the observer is capable of predicting the temperatures of the first, third, and fourth zones reliably, where the maximum estimation errors are equal to 0.421 C, 0.398 C, and 0.135 C, respectively.

Fig. 4.5b and 4.5e show the results for another case, wherein the observer is still using a single sensor but N_e is equal to 100. For this case, the estimation process starts earlier, at time equals to 3 hours, when compared to the previous case, as N_e is smaller. It is clear that the observer does not perform as reliably as it did in the previous case, particularly with regard to predicting the temperatures of the first and third zones, wherein the maximum prediction errors are equal to 7.022 C and 5.384 C, respectively. From both cases explained above, we conclude that increasing the value of N_e helps to improve the performance of the state observer although a bigger N_e means a longer time is needed for the data collecting process.

Alternatively, we will run the observer in new cases, where in addition to N_e, the number of installed sensors is varied. Fig. 4.5c shows the estimation error for a case wherein the observer is using two sensors, installed in the second and third zones, and N_e is equal to 100. We can see that the observer's performance is improved when compared to the case shown in Fig. 4.5b and 4.5e, wherein N_e was also equal to 100. In fact, it is clear that the prediction of the first and fourth zones' temperatures, where no sensors are installed, is nearly equivalent to that of the second and third zones, where we assumed the presence of sensors, a clear indication of a reliable estimation. Finally, Fig. 4.5f shows the estimation error for the case wherein the observer is using

the measurements of three sensors installed in the second, third, and fourth zones, and N_e is equal to 30. Similar to the previous case, we can see nearly equivalent results for all predictions, however, with more sensors and fewer collected data samples. In summary, we conclude from all the cases above that the performance of the formulated optimal observer improves as the value of N_e increases. Furthermore, we also conclude that in order to produce a reliable estimation, the observer needs fewer data samples as the number of installed sensors increases, which, in turn, indicates that a shorter time is needed to collect those samples.

In order to analyze the robustness of both our MPC-based controller and optimal observer, we run a number of simulations wherein we assumed the presence of measurement noise and system uncertainty. In these simulations, an MPC-based controller regulates the temperature in four indoor zones where control commands are generated using either measurements from each zone's temperature sensor or estimations from the optimal observer formulated in Section 4.5.1. Also, for each simulation run, we calculated the mean absolute temperature tracking error e_i (for i = 1,2,3,4) for the four zones. Depending on the number of measurements used in state estimation, we color-coded the results for using one, two, three, and four sensors with brown, green, yellow, and blue, respectively.

Table 4.1 shows the values of e_i calculated for the four zones wherein measurements

Table 4.1. Mean absolute tracking error e_i (for i = 1,2,3,4) calculated with measurements noise of zero mean and the variance shown in the leftmost column. Brown, green, yellow, and blue colored cells indicate state estimation calculated using measurements from one, two, three, and four sensors, respectively.

	using sensor measurements				using estimated state vector $\hat{x}$							
	e_1	e_2	e_3	e_4	e_1		e_2		e_3		e_4	
20%	0.524	0.391	0.565	0.343	0.226	0.213	0.274	0.194	0.344	0.316	0.322	0.301
					0.207	0.162	0.199	0.181	0.236	0.211	0.329	0.196
40%	1.519	1.311	1.942	1.761	0.238	0.216	0.294	0.237	0.351	0.317	0.320	0.325
					0.214	0.201	0.215	0.209	0.321	0.214	0.324	0.214
60%	2.411	2.623	3.052	3.098	0.343	0.311	0.305	0.285	0.348	0.328	0.347	0.361
					0.306	0.293	0.240	0.224	0.325	0.291	0.336	0.237
80%	3.746	3.730	3.849	3.992	0.402	0.358	0.363	0.348	0.364	0.342	0.421	0.405
					0.369	0.362	0.341	0.346	0.349	0.327	0.344	0.347

are susceptible to zero mean noise and the variance shown in the leftmost column. We can see that the controller performs well, with noise variance equal to or below 20%, and wherein temperature tracking is achieved with e_i values close to or below 0.5C. However, we can also see that the performance of the controller worsens as the variance increases above 20%, as the values of e_i are also increasing. On the other hand, it is clear that the use of the estimated state vector $\hat{x}$ with the MPC-based controller improves results, regardless of the number of sensors used for estimation, since all values of e_i are close to or below 0.4C. Furthermore, it is also clear that employing more sensor reading for estimations improves the temperature tracking by reducing the value of e_i.

Table 4.2 shows the values of e_i calculated for the four zones wherein we assumed that the system matrix A, which is used to simulate the thermal dynamics of the building, has uncertainty. That is, the matrix A is perturbed by random values as follows

$$A_{new} = A + \beta .* A \tag{4.30}$$

where $\beta \in \mathbb{R}^{n \times n}$ is a matrix containing random uniformly distributed numbers generated within the ranges shown in the leftmost column of Table 4.2. Whether using direct measurements or an estimation, we can see that the controller is performing well with system uncertainty for the cases corresponding to the range of ($\pm$10%) or below. Similarly, the same pattern is exhibited when using $\hat{x}$, as the values of e_i are

Table 4.2. Mean absolute tracking error e_i calculated with system uncertainty where matrix A is perturbed with random values selected from within the ranges shown in the leftmost column. Cell colors are defined similarly as in Table 4.1.

	using sensor measurements				using estimated state vector $\hat{x}$							
	e_1	e_2	e_3	e_4	e_1		e_2		e_3		e_4	
$\pm$2%	0.297	0.209	0.375	0.355	0.368	0.225	0.223	0.251	0.385	0.346	0.374	0.326
					0.220	0.213	0.239	0.230	0.357	0.317	0.325	0.341
$\pm$6%	0.233	0.241	0.476	0.543	0.676	0.439	0.328	0.361	0.520	0.386	0.382	0.375
					0.364	0.356	0.313	0.352	0.391	0.404	0.382	0.389
$\pm$10%	0.491	0.482	0.507	0.556	0.791	0.510	0.496	0.426	0.612	0.474	0.545	0.527
					0.463	0.421	0.424	0.411	0.459	0.461	0.536	0.425
$\pm$30%	2.305	1.249	2.203	1.791	2.114	1.629	1.531	1.197	2.252	2.159	1.765	1.173
					0.948	0.957	0.871	0.855	1.221	1.127	1.046	1.135

reduced, and employing more sensors for the state estimation process improves the temperature tracking.

4.5.2 First MTD based algorithm (MTD1)

In order to understand the main idea behind this algorithm, we will begin with a case study. We will consider a building with four zones where a single temperature sensor is installed in each zone. Therefore, the following combinations of sensor sets can render the system fully observable and, hence, will be used in the state estimation process

$$\begin{aligned} \text{sensor} =& \{s_1, s_2, s_3, s_4\} \\ \text{sensor set} =& \{s_4, s_3, s_3s_4, s_2, s_2s_4, s_2s_3, \\ & s_2s_3s_4, s_1, s_1s_4, s_1s_3, s_1s_3s_4, \\ & s_1s_2, s_1s_2s_4, s_1s_2s_3\} \end{aligned} \tag{4.31}$$

where s_i is the sensor installed in the i^{th} zone. Each one of those sets can be obtained using the corresponding row(s) of the output matrix C. For this particular case, we will assume that the second sensor s_2 was targeted with a negative bias attack. Therefore, the sensor set given in (4.31) will contain intact sets, which include s_2, and flawed sets, which do not include s_2. As a first step of this algorithm, the following cost function J is calculated for each set given in (4.31)

$$\begin{aligned} J_i(k) = & \left[y_i(k) - C_i\hat{x}_i(k)\right]^T Q_{y_i} \left[y_i(k) - C_i\hat{x}_i(k)\right] \\ & \text{for } i = 1, \ldots, |\text{sensor set}| \end{aligned} \tag{4.32}$$

where C_i is the row(s) of output matrix C that result in the output vector y_i, and Q_{y_i} is a weighing matrix. Figures 4.6a and 4.6b show the calculated values of J for both the intact and flawed sets, given in (4.31), respectively. For both those figure, the MTD based algorithm is engaged at the same time with the optimal observer, which is almost at time

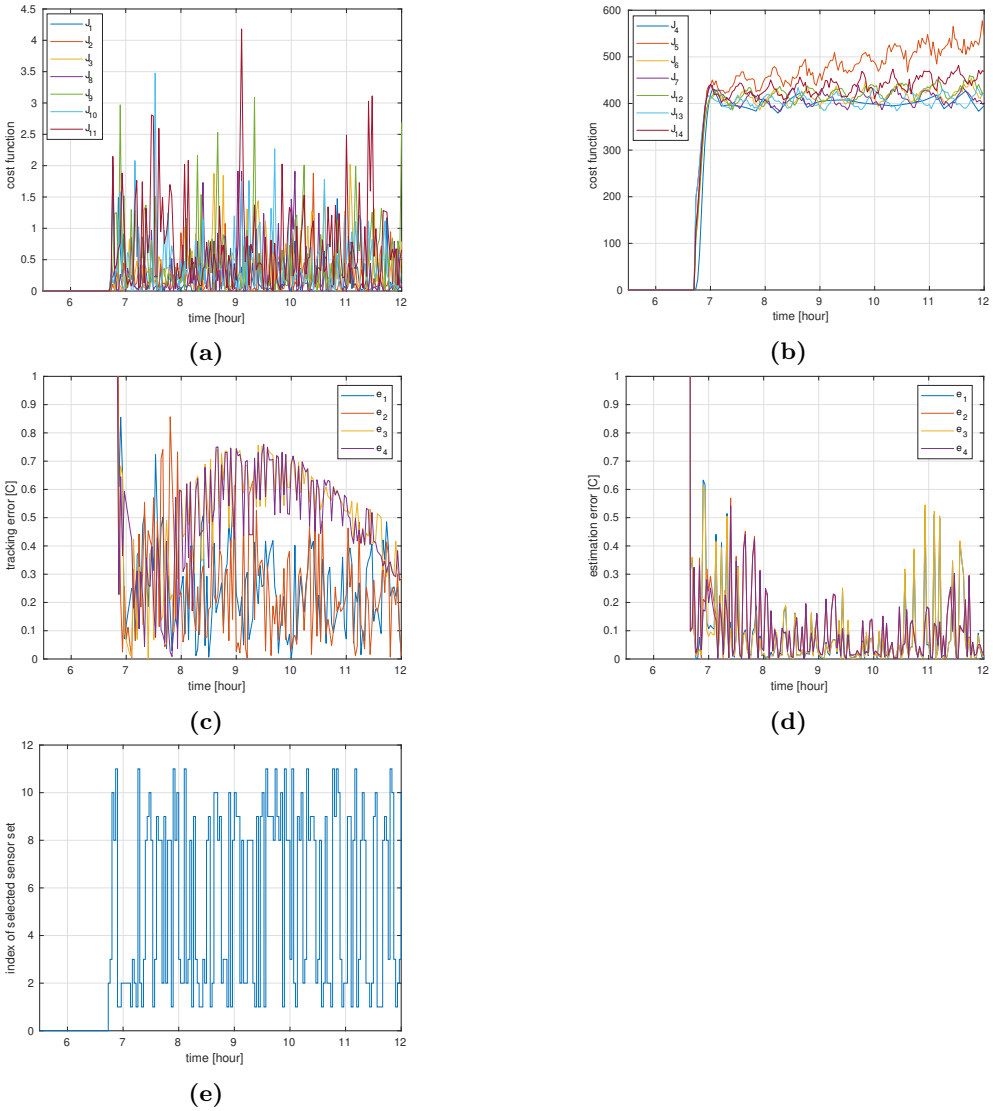


Fig. 4.6. a) Cost function (J) calculated, using (4.32), for the intact sensor sets. b) Cost function (J) calculated for the flawed sensor sets. c) Absolute temperature tracking error when the MPC controller is using measurements selected by the MTD based algorithm. d) Absolute temperature estimation error when the observer is using measurements selected by the MTD based algorithm. e) Indices of sensor sets selected by the MTD based algorithm.

equal to 6.6 hours. We can see clearly that intact sets produce smaller values for J compared to those produced by the flawed sets. This is an expected behaviour due to the presence of a manipulated measurement from s_2 which, in turn, will induce an estimated vector $\hat{x}$ deviating from the actual measurements vector y. As a result, we have an indication of an attacked (manipulated) measurement although we cannot define exactly which sensor is

Algorithm 3: First MTD based defense (MTD1)

Input: a combination of sensor sets along with their measurement y_i vectors, for $i = 1, \ldots, |\text{sensor sets}|$.

Output: a randomly selected estimated state vector $\hat{x}_r$, where $1 \leq r \leq |\text{sensor set}|$.

for $i = 1 : |sensor\ set|$ **do**
    determine $\hat{x}_i$ using the optimal observer and y_i ;
    $J_i \leftarrow \left[y_i - C_i\hat{x}_i\right]^T Q_{y_i} \left[y_i - C_i\hat{x}_i\right]$;
arrange J_i in an ascending order;
randomly select a J_r from the first half of the ascending arrangement;
$\hat{x}_r \leftarrow$ the estimated state vector corresponding to J_r;

giving that measurement. Therefore, we will benefit from this attack indication in suggesting our first MTD based algorithm whose steps are as follows

1. At the current time step k, use the optimal observer to determine $\hat{x}_i(k)$ for each sensor set.

2. Calculate $J_i(k)$ using (4.32) for each sensor set. Then, arrange $J_i(k)$ in an ascending order.

3. Out of the first half of the ascending arrangement, randomly select a $J_r(k)$, where $1 \leq r \leq |\text{sensor set}|$.

4. Use the estimated state vector $\hat{x}_r(k)$, obtained using the measurements provided by the r^{th} set, in the MPC controller to determine $u(k)$.

5. For the next time step $k + 1$, collect measurements from the sensors and go to step (1).

The steps for our first MTD based defense approach are shown in Algorithm 3. For the same attack scenario against s_2, Figure 4.6c shows the absolute temperature tracking error when the MPC controller is employing measurements selected by the above explained algorithm. Similarly, Figure 4.6d shows the absolute temperature estimation error when the optimal observer is employing measurements selected by the above explained algorithm. In

addition, Figure 4.6e shows the indices of selected sets which belong to the following range $(1, 2, 3, 8, 9, 10, 11)$. We can see that the algorithm selects only one of the intact set, whose indices are shown in the legend of Figure 4.6a, which indicates the success of our MTD based approach to deter the attack mounted against s_2.

Since increasing the number of targeted sensors will reduce the number of intact sets, this algorithm can fail to reduce the impacts of multi sensor attacks. One possible remedy for this drawback is to increase the number of installed temperature sensors in the building. Another possible remedy is to modify the algorithm, which leads us to suggesting another MTD based algorithm in the next section.

4.5.3 Second MTD based algorithm (MTD2)

In this section, we suggest another MTD based algorithm with the goal to overcome the shortcomings of the first algorithm. For this algorithm, we also will use the same sensor sets given in (4.31) which make the system fully observable. In order to understand the idea behind this algorithm, we will also consider the same case study from the previous section. By examining Fig. 4.6a, we can see that the cost function J of the intact sets does not exactly reach zero, which is expected since we have inherent measurement noise. We will benefit from that Figure in selecting a threshold value, which we will refer to as ϵ henceforth, and use it in our second MTD based algorithm whose steps are as follows

1. Initially, define $\epsilon > 0$ by making use of Fig. 4.6a. ϵ should be a value where at each time step, one or multiple J, of intact sets, is equal to or smaller than.

2. At the current time step k, randomly select two sensor sets.

3. use the optimal observer to determine $\hat{x}_1(k)$ and $\hat{x}_2(k)$ for the selected sets.

4. Calculate the modified cost function for each selected set as follows

$$J_i(k) = \frac{1}{|k - k_0|} \sum_{l=k_0}^{k} \left\{ \left[y_i(l) - C_i \hat{x}_i(l) \right]^T Q_{y_i} \left[y_i(l) - C_i \hat{x}_i(l) \right] \right\} \tag{4.33}$$

where k_0 is the index of the initial time step, and $i = 1, 2$.

5. If the following is true

$$J_1(k) \leq \epsilon \text{ and } J_2(k) \leq \epsilon \tag{4.34}$$

then select either one of the sets J_r, where $r \in \{1, 2\}$, and go to step (9).

6. Else if the following is true

$$J_1(k) > \epsilon \text{ and } J_2(k) \leq \epsilon \tag{4.35}$$

then, ignore $J_1(k)$ along with its sensor set. Select another set randomly and go to step (3).

7. Else if the following is true

$$J_1(k) \leq \epsilon \text{ and } J_2(k) > \epsilon \tag{4.36}$$

then, ignore $J_2(k)$ along with its sensor set. Select another set randomly and go to step (3).

8. Else, ignore both sets. Select another two sets randomly and go to step (3).

9. Use the estimated state vector $\hat{x}_r(k)$, obtained in step (3), in the MPC controller to determine $u(k)$.

10. For the next time step $k + 1$, go to step (2).

The steps for our second MTD based defense approach are shown in Algorithm 4. Results for using this algorithm and a performance comparison, with respect to the previous algorithm, are given in the next section.

4.5.4 Performance evaluation for the MTD-based algorithms

In order to show the efficacy of the two suggested defenses, we will evaluate the performance of the two MTD-based algorithms with the designed MPC controller when one or

Algorithm 4: Second MTD based defense (MTD2)

Input: a combination of sensor sets along with their measurement y_i vectors, for $i = 1, \ldots, |\text{sensor sets}|$, a threshold value $\epsilon > 0$, initial and current time steps k_0 and k.

Output: a randomly selected estimated state vector $\hat{x}_r$, where $1 \leq r \leq |\text{sensor set}|$.

```
randomly select two sensor sets y_1 and y_2 ;
BreakFlag ← 0 ;
while BreakFlag = 0 do
    for i = 1 : 2 do
        determine x̂_i using the optimal observer and y_i ;
        J_i ← equation (4.33) ;
    if (J_1 ≤ ε and J_2 ≤ ε) then
        J_r ← either J_1 or J_2 ;
        BreakFlag ← 1 ;
    else if (J_1 > ε and J_2 ≤ ε) then
        ignore J_1 and y_1 ;
        J_1 ← randomly select another set ;
    else if (J_1 ≤ ε and J_2 > ε) then
        ignore J_2 and y_2 ;
        J_2 ← randomly select another set ;
    else
        ignore J_1, J_2, y_1, and y_2 ;
        J_1 ← randomly select another set ;
        J_2 ← randomly select another set ;
x̂_r ← the estimated state vector corresponding to J_r ;
```

multiple sensors are compromised. In our evaluation, we consider a building with four zones that utilizes an MPC controller. The MPC controller inputs are the measurements provided from four temperature sensors, one installed per each zone. In this case, the four sensors form a total of 14 sensor sets wherein each one can be used for the state estimation process. We also assume that all the installed sensors are susceptible to the three types of attacks defined in the threat model. Furthermore, in our evaluation, we include the results for a baseline scenario in which the MPC controller operates without any of the MTD defenses, and also attack-free scenarios in which the MPC controller operates with either of the two defenses, though the sensors are not compromised.

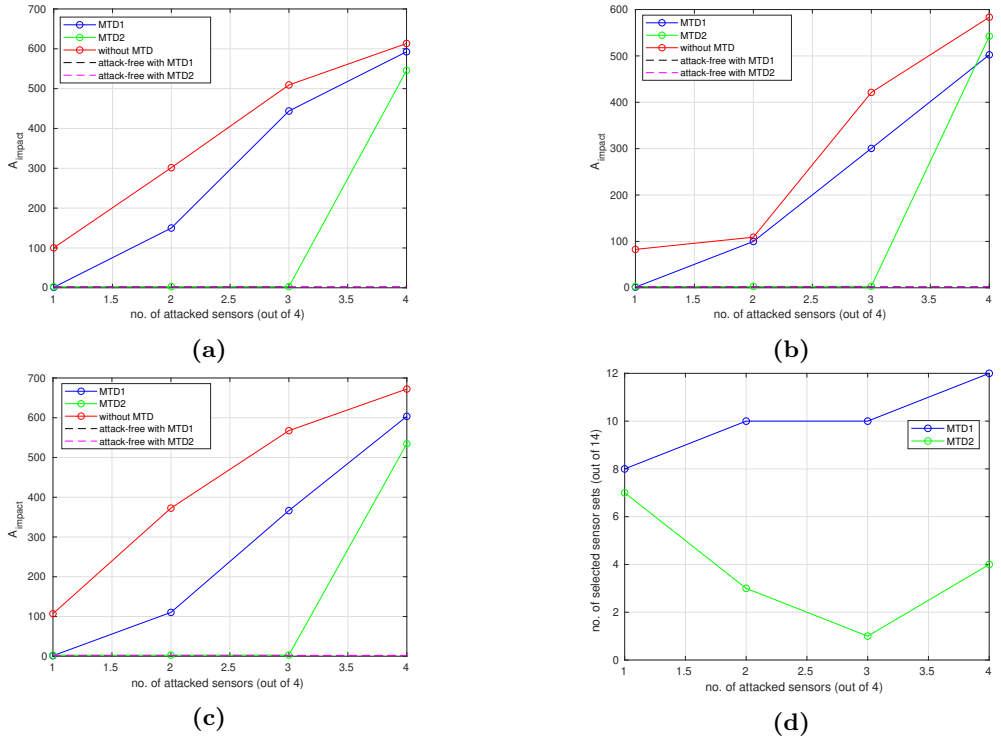


Fig. 4.7. A comparison for using the suggested MTD based algorithms to reduce A_{impact} generated by the following attacks: a) negative bias, b) sinusoidal, and c) random. d) Number of selected sensor sets by the two MTD based algorithms and for different cases of random attack. For all these results, we assume that four temperature sensors are installed in the building.

Fig. 4.7a shows the attack impact A_{impact} for varying cases with respect to the application of the MTD-based algorithms and the number of sensors targeted by a constant-value attack. In this Figure, we can see that both MTD-based algorithms manage to deter the above-mentioned attack, targeting only a single sensor in the building, for which the resulting A_{impact} is nearly equivalent with respect to each attack and to that resulting from the attack-free cases. However, for cases involving multiple-sensor attacks, we can see clearly that MTD1 fails when it comes to attack cases targeting 2 (50%) or more of the installed sensors. On the other hand, MTD2 produces values for A_{impact} for attack cases involving up to 3 (75%) of the sensors equivalent to those produced by the attack-free cases, and this algorithm fails only with attacks targeting all of the sensors, which, in turn, increases the cost of the mounted attacks.

Fig. 4.7b shows A_{impact} under different cases involving a sinusoidal attack. Similar to the previous attack case, we see the same performance by the two MTD-based algorithms. We can also draw the same conclusion from Fig. 4.7c, which shows the results for cases of random attack. In general, we can see from the results depicted in Fig. 4.7 that both algorithms achieve the main goal of MTD, which is to create a proactive defense by adding uncertainty to the control structure employed by the smart building, using the potentially attacked sensors, and, hence, reducing the attack space, even though one of the algorithms outperforms the other.

For MTD1, the algorithm relies on the values of performance indices, calculated using (4.32) without considering how significantly big or small those values are. For that reason and also because of the limited number of sensor sets, we see that that algorithm fails to deter attacks against multiple sensors. On the other hand, the MTD2 algorithm relies on the performance indices, calculated using (4.33), as well as the predefined threshold ϵ. This reliance allows the algorithm to search for the intact sensor sets and randomly select one of them. As a result, it is clear from Fig. 4.7a, 4.7b, and 4.7c that MTD2 fails only when all (100%) of the sensors are attacked, resulting in the unavailability of intact sets. Fig. 4.7d shows the number of selected sets by both MTD-based algorithms for the different cases of random attack depicted in Fig. 4.7c. We can see clearly how MTD tends to select fewer numbers of sets as the number of attacked sensors increases, meaning that the number of intact sets decreases.

Alternatively, we will evaluate the performance of the suggested algorithms with a redundancy of sensors. More specifically, we will consider a four-zoned smart building with an MPC controller whose inputs are the measurements from eight temperature sensors, two installed per each zone. In this case, the eight sensors provide a total of 64 sensor sets, each of which can be used for the state estimation process. Fig. 4.8a, 4.8b, and 4.8c show A_{impact} with the application of the MTD-based algorithms with sensors redundancy, for different cases of constant-value, sinusoidal, and random attacks, respectively, mounted against one or multiple sensors. In all the aforementioned figures, we can see that MTD2 exhibits

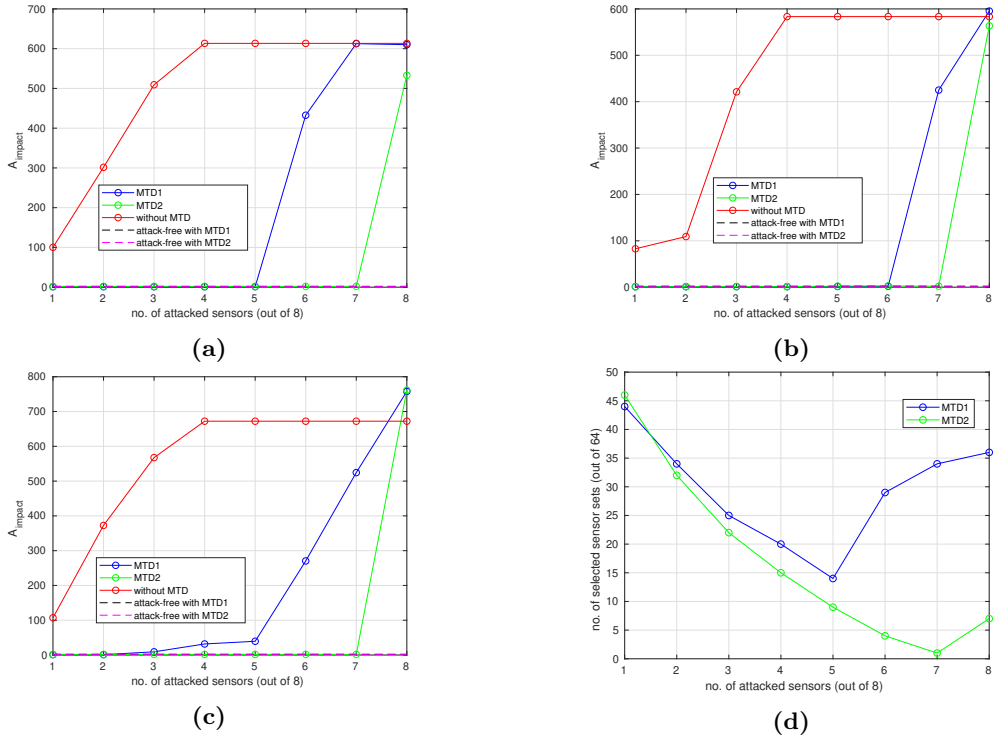


Fig. 4.8. A comparison for using the suggested MTD based algorithms to reduce A_{impact} generated by the following attacks: a) negative bias, b) sinusoidal, and c) random. d) Number of selected sensor sets by the two MTD based algorithms and for different cases of random attack. For all these results, we assume that eight temperature sensors are installed in the building.

the same performance, in that it fails only when all the equipped sensors are attacked. Furthermore, in Fig. 4.8d, which shows the number of selected sets for different cases of random attack, MTD2 selects sets in a descending order as the number of attacked sensors grows, which, in turns, reduces the attack window.

Fig. 4.8a and 4.8b show that the performance of MTD1 is improved with sensor redundancy, as the algorithm fails when 75% and 87.5% of the installed sensors are attacked, respectively. However, we can see in Fig. 4.8c that MTD1 fails after 50% of the sensors are attacked. This drawback could be solved by adding more sensors to the system structure, and, as a result, creating more sensor sets. In summary, the suggested MTD-based algorithms are able to reduce the impacts of attacks mounted against the temperature sensors. On one hand, MTD1 requires sensor redundancy in order to be able to deter multiple-sensor

attack cases. On the other hand, MTD2 does not need more than one sensor per zone as long as the resulting sensor sets make the system observable. Furthermore, MTD2 only fails when all sensors are attacked. However, it should be noted that implementing MTD2 requires a number of offline system simulations to calculate the cost functions, shown in Fig. 4.6a and 4.6b, in order to select an appropriate value for ϵ.

4.6 Conclusion

In this chapter, we considered the HVAC systems which are employed in residential and commercial smart buildings to provide thermal comfort for the occupants. First, we formulated an MPC based controller with the goal of tracking a desired temperature for each zone in the building. Then, we analyzed the impacts that could result from targeting (attacking) the temperature sensors, with readings used by the MPC-based controller, by manipulating their measurements. Such deliberate actions could lead to deceiving the controller, and, hence, cause discomfort for the occupants. For that reason, we suggested two MTD-based algorithms, with the goal of reducing the impacts of such attacks. Our suggested algorithms constantly change the sensor set with readings fed to the controller and also estimate the other non-measured temperatures. The results showed an improvement in the performance of the MPC-based controller if combined with either one of the algorithms, in terms of providing an acceptable temperature tracking despite the presence of sensor attacks.

REFERENCES

[1] D. Snoonian, "Smart buildings," *IEEE Spectrum*, vol. 40, no. 8, pp. 18–23, Aug 2003.

[2] J. Kleissl and Y. Agarwal, "Cyber-physical energy systems: Focus on smart buildings," in *Design Automation Conference*, June 2010, pp. 749–754.

[3] T. Weng and Y. Agarwal, "From buildings to smart buildings-sensing and actuation to improve energy efficiency," *IEEE Design Test of Computers*, vol. 29, no. 4, pp. 36–44, Aug 2012.

[4] D. Minoli, K. Sohraby, and B. Occhiogrosso, "Iot considerations, requirements, and architectures for smart buildings-energy optimization and next-generation building management systems," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 269–283, Feb 2017.

[5] J. B. Rawlings, "Tutorial overview of model predictive control," *IEEE Control Systems Magazine*, vol. 20, no. 3, pp. 38–52, June 2000.

[6] K. R. Muske and J. B. Rawlings, "Model predictive control with linear models," *AIChE Journal*, vol. 39, no. 2, pp. 262–287, 1993. [Online]. Available: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.690390208

[7] M. Jagielski, N. Jones, C.-W. Lin, C. Nita-Rotaru, and S. Shiraishi, "Threat detection for collaborative adaptive cruise control in connected cars," in *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, ser. WiSec '18. New York, NY, USA: ACM, 2018, pp. 184–189. [Online]. Available: http://doi.acm.org/10.1145/3212480.3212492

[8] R. van der Heijden, T. Lukaseder, and F. Kargl, "Analyzing attacks on cooperative adaptive cruise control (cacc)," in *2017 IEEE Vehicular Networking Conference (VNC)*, Nov 2017, pp. 45–52.

[9] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 21–32. [Online]. Available: http://doi.acm.org/10.1145/1653662.1653666

[10] M. A. Rahman and H. Mohsenian-Rad, "False data injection attacks with incomplete information against smart power grids," in *2012 IEEE Global Communications Conference (GLOBECOM)*, Dec 2012, pp. 3153–3158.

[11] B. DeBruhl, S. Weerakkody, B. Sinopoli, and P. Tague, "Is your commute driving you crazy?: a study of misbehavior in vehicular platoons," in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 2015, p. 22.

[12] S. H. Kafash, J. Giraldo, C. Murguia, A. A. Cardenas, and J. Ruths, "Constraining attacker capabilities through actuator saturation," in *2018 Annual American Control Conference (ACC)*, June 2018, pp. 986–991.

[13] M. A. Rahman, E. Al-Shaer, and R. B. Bobba, "Moving target defense for hardening the security of the power system state estimation," in *Proceedings of the First ACM Workshop on Moving Target Defense*, ser. MTD '14. New York, NY, USA: ACM, 2014, pp. 59–68. [Online]. Available: http://doi.acm.org/10.1145/2663474.2663482

[14] R. Zhuang, S. A. DeLoach, and X. Ou, "Towards a theory of moving target defense," in *Proceedings of the First ACM Workshop on Moving Target Defense*, ser. MTD '14. New York, NY, USA: ACM, 2014, pp. 31–40. [Online]. Available: http://doi.acm.org/10.1145/2663474.2663479

[15] C. B. Jones and C. Carter, "Trusted interconnections between a centralized controller and commercial building hvac systems for reliable demand response," *IEEE Access*, vol. 5, pp. 11 063–11 073, 2017.

[16] Z. Du, X. Jin, and Y. Yang, "Fault diagnosis for temperature, flow rate and pressure sensors in vav systems using wavelet neural network," *Applied Energy*, vol. 86, no. 9, pp. 1624 – 1631, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261909000233

[17] D. C. Wardell, R. F. Mills, G. L. Peterson, and M. E. Oxley, "A method for revealing and addressing security vulnerabilities in cyber-physical systems by modeling malicious agent interactions with formal verification," *Procedia Computer Science*, vol. 95, pp. 24 – 31, 2016, complex Adaptive Systems Los Angeles, CA November 2-4, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050916324619

[18] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*, 1st ed. Springer Publishing Company, Incorporated, 2011.

[19] M. Dunlop, S. Groat, W. Urbanski, R. Marchany, and J. Tront, "Mt6d: A moving target ipv6 defense," in *2011 - MILCOM 2011 Military Communications Conference*, Nov 2011, pp. 1321–1326.

[20] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "Openflow random host mutation: Transparent moving target defense using software defined networking," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN '12. New York, NY, USA: ACM, 2012, pp. 127–132. [Online]. Available: http://doi.acm.org/10.1145/2342441.2342467

[21] J. Tian, R. Tan, X. Guan, and T. Liu, "Hidden moving target defense in smart grids," in *Proceedings of the 2Nd Workshop on Cyber-Physical Security and Resilience in Smart Grids*, ser. CPSR-SG'17. New York, NY, USA: ACM, 2017, pp. 21–26. [Online]. Available: http://doi.acm.org/10.1145/3055386.3055388

[22] J. Tian, R. Tan, X. Guan, and T. Liu, "Enhanced hidden moving target defense in smart grids," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 2208–2223, March 2019.

[23] S. Weerakkody and B. Sinopoli, "Detecting integrity attacks on control systems using a moving target approach," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 5820–5826.

[24] Z. Pang, G. Liu, and Z. Dong, "Secure networked control systems under denial of service attacks*," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 8908 – 8913, 2011, 18th IFAC World Congress. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S147466701645041X

[25] A. Kanellopoulos and K. G. Vamvoudakis, "A moving target defense control framework for cyber-physical systems," *IEEE Transactions on Automatic Control*, pp. 1–1, 2019.

[26] J. Giraldo and A. A. Cardenas, *Moving Target Defense for Attack Mitigation in Multi-Vehicle Systems*. Cham: Springer International Publishing, 2019, pp. 163–190. [Online]. Available: https://doi.org/10.1007/978-3-030-10597-6_7

[27] M. Maasoumy, A. Pinto, and A. Sangiovanni-Vincentelli, "Model-based hierarchical optimal control design for hvac systems," *ASME 2011 Dynamic Systems and Control Conference and Bath/ASME Symposium on Fluid Power and Motion Control, DSCC 2011*, vol. 1, 01 2011.

[28] P. Bacher and H. Madsen, "Identifying suitable models for the heat dynamics of buildings," *Energy and Buildings*, vol. 43, no. 7, pp. 1511 – 1522, 2011.

[29] M. Maasoumy, M. Razmara, M. Shahbakhti, and A. S. Vincentelli, "Handling model uncertainty in model predictive control for energy efficient buildings," *Energy and Buildings*, vol. 77, pp. 377 – 392, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378778814002771

[30] T. Wei, Yanzhi Wang, and Q. Zhu, "Deep reinforcement learning for building hvac control," in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2017, pp. 1–6.

[31] A. Kelman and F. Borrelli, "Bilinear model predictive control of a hvac system using sequential quadratic programming," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 9869 – 9874, 2011, 18th IFAC World Congress. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016451979

[32] F. Borrelli, *Constrained optimal control of linear and hybrid systems*. Springer, 2003, vol. 290.

[33] D. Ruscio, "Model predictive control with integral action: A simple mpc algorithm," *Modeling, Identification and Control: A Norwegian Research Bulletin*, vol. 34, pp. 119–129, 01 2013.

[34] M. A. Stephens, C. Manzie, and M. C. Good, "Model predictive control for reference tracking on an industrial machine tool servo drive," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 2, pp. 808–816, May 2013.

[35] S. Clements and H. Kirkham, "Cyber-security considerations for the smart grid," in *IEEE PES General Meeting*, July 2010, pp. 1–5.

[36] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on cyber security for smart grid communications," *IEEE Communications Surveys Tutorials*, vol. 14, no. 4, pp. 998–1010, Fourth 2012.

[37] W. Wang and Z. Lu, "Cyber security in the smart grid: Survey and challenges," *Computer Networks*, vol. 57, no. 5, pp. 1344 – 1371, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128613000042

[38] J. P. Hespanha, *Linear Systems Theory: Second Edition*, ned - new edition, 2 ed. Princeton University Press, 2018. [Online]. Available: http://www.jstor.org/stable/j.ctvc772kp

CHAPTER 5

ATTACK MITIGATION IN ADVERSARIAL PLATOONING USING DETECTION-BASED SLIDING MODE CONTROL AND SWITCHING OF CONTROL FROM AUTO TO HUMAN

In this chapter, we study the behavior of a single vehicular platoon where one of the platooned vehicles is controlled by an attacker. The latter is able to modify the platooning controller of the seized vehicle and, hence, produce sudden accelerating/decelerating movements that can lead to collisions inside the platoon. A previous study suggested a sliding mode controller which uses only local vehicular sensor information without the need for inter-vehicle communications, to mitigate the impacts of the aforementioned attack. Also, the suggested controller is assisted with a decentralized attack detection. Simulation results from that study demonstrate that collisions are eliminated, or significantly reduced at certain cases. However, the same results also indicate that the intact vehicles concede platooning and start following the attacker. For instance, the lead vehicle, assuming not being attacked, will no longer follow a reference trajectory, which is part of its platooning goals, once it detects an attack behind it. Therefore, we will modify the suggested mitigation controller such that the collisions are also reduced and the control of intact vehicles will eventually switch from auto to human driving, i.e., disband the platoon, such that the attacker cannot have an influence on them anymore.

5.1 Introduction

The formation and maintaining of automated vehicular platoons is an area of extensive and ongoing research. Platooning has been shown to have environmental, safety, and passenger comfort benefits [1, 2]. They also help to alleviate traffic congestion on highways [3] and have proven more fuel efficient than manually-operated vehicles [4]. The safety and security of a platooning CPS is important and essential. For a platooning CPS, it has been

shown that a single attacker can disrupt normal operations simply and easily, and that such disruptions can cause catastrophic collisions [5]. For that reason, a previous study [6] proposed a sliding mode controller aimed at ensuring that the impacts induced as a result of the attack presented in [5] would be mitigated. More specifically, a sliding mode controller was formulated with the goal of ensuring that deviations from expected platoon behavior due to the aforementioned attack would not cause collisions.

In this chapter, our work builds upon the results presented in [6] and also attempts to solve the safety problem of an adversarial environment. The results of the aforementioned work indicated that the proposed sliding mode controller was able to considerably eliminate, or at least significantly educe, accidents. However, though the engagement of the proposed controller resulted in a collision-free platoon, said platoon was still influenced by the attacker-controlled vehicle. For that reason, we modify the proposed sliding mode controller such that it is still able to avoid collisions and, at the same time, lead the way to disbanding, which is switching control from auto to humans.

5.1.1 Related work

Sliding Mode Control has been used previously in many scenarios. Platooning strategies exist where sliding mode control has been used in a homogeneous platoon [7] under normal operation. Graph theoretic approaches similar to ours have been used before in platooning [8, 9], and in general problems of multiple vehicle target tracking in the presence of uncertainties [10].

Apart from platooning, much work has been completed in interconnected dynamic cyber-physical systems. The security and robustness of these systems in the face of an attack or failure is crucial and an active area of research [11]. Graph theory and information flow analysis have been used to analyze such systems as well [8, 9]. Much of this work focused on ensuring suitable operating conditions for dynamic systems, with an emphasis on stability, controllability, and observability.

5.1.2 Organization

Section 5.2 explains the platooning, vehicle, and threat models used in simulations for this chapter. In Section 5.3, we present the sliding mode controller and its modification for attacks mitigation. Section 5.4 discusses the simulation results. Conclusions are given in Section 5.5.

5.2 System Model

In this section, we discuss the vehicle, platooning, and threat models used for simulations of this chapter.

5.2.1 Platooning Model

In keeping with the current literature [12, 13], each vehicle is analyzed as a double integrator system, where the control input is a desired acceleration. For an n-vehicle platoon, the state vector $x \in \mathbb{R}^{2n}$ is made up of positions and velocities and the input vector $u \in \mathbb{R}^n$ consists of control inputs. The state and input vectors can be expressed as

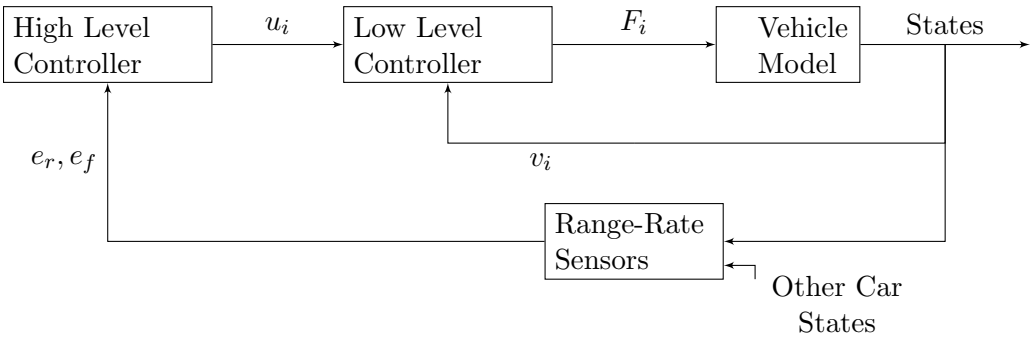
$$\begin{aligned} x &= \begin{bmatrix} x_1 & x_2 & \dots & x_n & v_1 & v_2 & \dots & v_n \end{bmatrix}^T \\ u &= \begin{bmatrix} u_1 & u_2 & \dots & u_n \end{bmatrix}^T \end{aligned} \tag{5.1}$$

and the state space system becomes

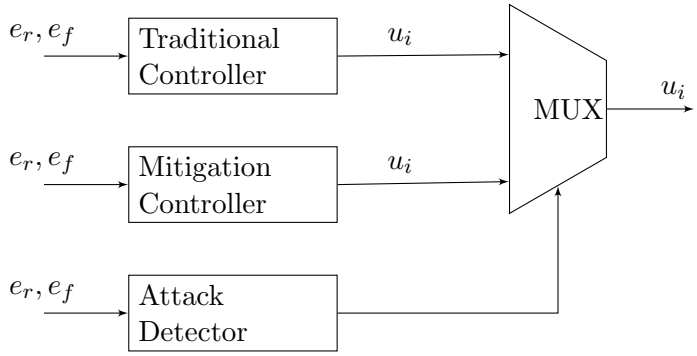
$$\dot{x} = \begin{bmatrix} 0_{n\times n} & I_{n\times n} \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix} x + \begin{bmatrix} 0_{n\times n} \\ I_{n\times n} \end{bmatrix} u \tag{5.2}$$

where car i has position and velocity x_i, v_i, respectively, and control input u_i. These positions are measured from the center of mass of all the cars. In the bidirectional control scheme, we have

$$u_i = f_i(x_{i-1} - x_i, v_{i-1} - v_i, x_{i+1} - x_i, v_{i+1} - v_i) \tag{5.3}$$



(a) Controllers of a single vehicle.



(b) Inside high level controller: attack detection and controller selection based on attack state.

Fig. 5.1. Overview of Platoon. Each vehicle knows its own velocity and measures a relative distance and velocity from rear and front (e_r, e_f). These same measurements are used in the high level controller to switch between rear or front tracking if an attack is detected [6].

which means that each vehicle's control input can only use relative distance and speed measurements, obtained using the local range and range rate sensors, respectively, with respect to its immediate neighbors. This function f_i constitutes a high level controller that is meant to be independent of a vehicle's dynamics (Fig. 5.1b); as such the control input u_i serves as the vehicles desired acceleration.

As the rearmost and leader vehicle lack a follower and predecessor, respectively, they follow a slightly modified version of (5.3) wherein the rearmost car uses a unidirectional

law, and the leader follows a reference trajectory while maintaining a follower separation

$$\begin{aligned} u_1 &= f_1(x_{i+1} - x_i, v_{i+1} - v_i) \\ u_n &= f_n(x_{i-1} - x_i, v_{i-1} - v_i, x_{\text{ref}} - x_i, v_{\text{ref}} - v_i) \end{aligned} \tag{5.4}$$

5.2.2 Vehicle Model

A realistic model of a vehicle has a throttle input or some other type of actuator. The purpose of this section is to find how a desired acceleration can be achieved based on our knowledge of the vehicle. A model of the vehicle's dynamics is required in this case. This can be specific to different vehicles, but the general idea is to find an expression for the control input required for a desired acceleration. This constitutes the low level controller shown in Fig. 5.1a.

The vehicle model we use is a 2^{nd} order plant with a linear friction/drag coefficient. Such models are easy to analyze while capturing the major dynamics of the system. Similar models have been used in other control systems literature to analyze fundamental properties of single vehicles and platoons [14, 15]

$$\begin{aligned} \dot{x}_i &= v_i \\ \dot{v}_i &= \alpha F_i - \beta v_i \end{aligned} \tag{5.5}$$

where $F_i \in [F^-, F^+]$ is a variable to set the actuator (throttle) and α, β are the model's parameters which can be chosen based on the vehicle's internal design values or through system modeling [12].

For the high level controller described in (5.3) to work, we need to compensate for the internal dynamics of the vehicle. We use feedback linearization [12, 15] to compensate for terms in the model described by (5.5) gives us

$$F_i = \frac{1}{\alpha}\left(u_i + \beta v_i\right) \tag{5.6}$$

Note that this controller does require a velocity measurement of vehicle i. A sensor which

provides this reading will be required, but this is just car sensing its internal data and does not violate the decentralized condition.

The reason for including this model is to emphasize that there are bounds F^-, F^+ on F_i which lead to saturation. We simulate with these saturation limits in order to demonstrate our controller on a realistic system where the desired acceleration cannot always be achieved. A favorable consequence of this is that we cannot achieve infinite acceleration.

Substituting (5.6) into (5.5) gives us the required double integrator type system for each vehicle

$$\begin{aligned} \dot{x}_i &= v_i \\ \dot{v}_i &= u_i \end{aligned} \tag{5.7}$$

as long as the condition $\frac{1}{\alpha}(u_i + \beta v_i) \in [F^-, F^+]$ holds true. These saturation constraints apply to the attacker as well and ensure that we do not have a car with unrealistic capabilities.

We also add another constraint on this controller which prohibits reverse motion. This is to maintain relevance with the real application of AHS. It can be expressed as $F_i > F_i^-$ if $(v_i > 0)$ and $F_i > 0$ otherwise, which means that if a vehicle's speed is zero or below, it cannot apply negative actuator input.

5.2.3 Threat Model

For this chapter, we consider a platoon of n members, each equipped with front- and rear-facing sensors that measure relative distance and velocity. Aside from the attacker-controlled vehicle, each of the vehicles adheres to the same control law and has the same capabilities, as described in the previous section (5.2.2). The last member is indexed as 1 and the leader is at index n. We focus on the bidirectional platoon scheme [16], wherein every car gathers information (e.g. relative distance and speed) about, and reacts to the movements of, both the preceding and following vehicle. The leader attempts to maintain a separation with its follower and has access to a reference trajectory. The last car only

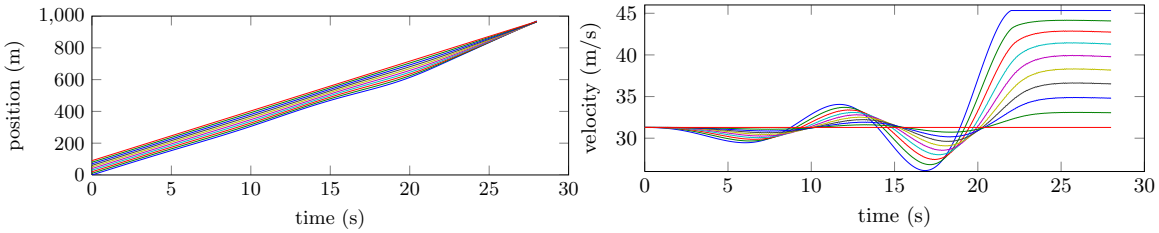


Fig. 5.2. Oscillatory behavior brought on by an attacker, resulting in a high speed crash [5]. Each line represents the trajectory of a vehicle in a ten vehicle platoon with an attacker at the rear.

tracks the car immediately in front of it.

We assume just a single attacker in control of a car at an arbitrary position in the platoon. The attack car is possibly more powerful than the regular cars, in that it may have greater acceleration capabilities. The goal of the attacker is to cause multiple collisions. To accomplish this, the attacker follows a modified control law that induces oscillations in the platoon (Fig. 5.2). It has been shown that an attacker can leverage oscillatory behavior to cause more accumulated damage, and more collisions over time, than if he simply accelerates in one direction. This can be achieved simply by changing the controller gains [5]. The attack always begins in a steady state configuration, when the cars are traveling at their desired separations, which was chosen to be one car length of separation in our tests.

If we limit our discussion to a single attacker, we propose to use the consensus condition that is required in both cases anyway. We recommend a secondary controller that attempts to keep a constant distance from the more dangerous and uncooperative car (in front or behind), relying instead on the other car to move and make room. Under normal circumstances, a traditional bidirectional law is followed; however, upon detection of anomalous behavior indicating the onset of an attack, this secondary controller is engaged to mitigate the attack (Fig. 5.1b). This approach allows a straightforward, ultimate boundedness analysis, and simulation results show that it greatly reduces total damage when compared with a bidirectional scheme.

Since total or instantaneous damage is not formally defined, we propose to use a metric that depends upon two things: whether an impact takes place and the relative velocity of

the colliding vehicles. The choice of measuring damage is motivated by previous work on automated vehicle and platooning safety [17, 18]. To measure the accumulation of damage, we assign the following as a rate of change to a state D:

$$\dot{D} = c^T v_{\text{rel}} \tag{5.8}$$

where c is an $n-1$ length vector whose entries are 0 normally, but 1 if there is a collision. v_{rel} is a vector containing the absolute values of $n-1$ relative velocities at time of collision.

5.3 Attack Mitigation

In [6], two sliding mode controllers were designed for the front and rear of each platooned vehicle. Then, the two controllers were combined using the graph theory [10]. Furthermore, a detection scheme was also suggested to detect possible attacks with respect to the front and rear of each platooned vehicle. The final form of one of the controllers, front or rear, is as follows

$$u_i = \text{sat}\left(\frac{s}{\epsilon}\right)[2k_1 v_{\max} + a_{\max} + \epsilon] \tag{5.9}$$

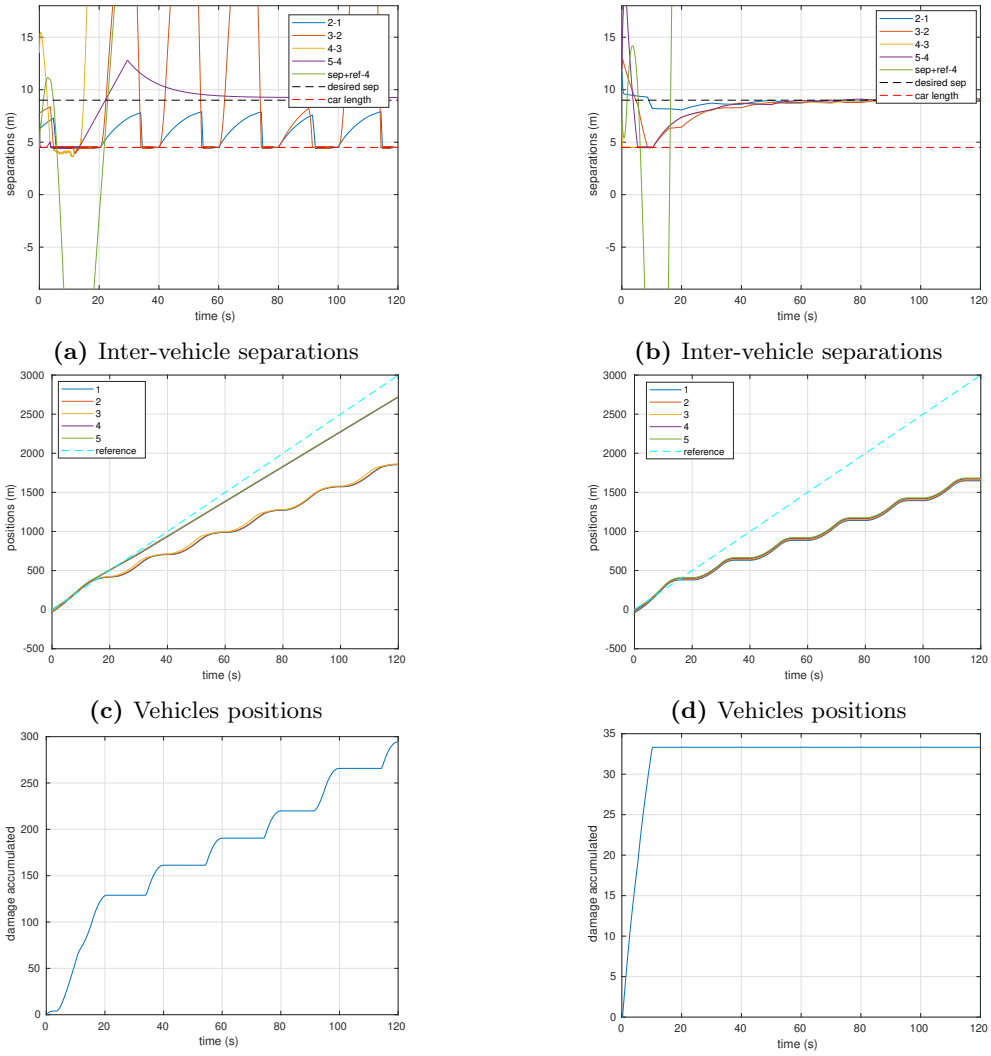
with

$$\begin{aligned} s &= k_1 e_1 + e_2 \\ e_1 &= x_{i+1} - x_i - \sigma_{ref} \\ e_2 &= v_{i+1} - v_i \end{aligned} \tag{5.10}$$

and

$$\text{sat}\left(\frac{s}{\epsilon}\right) = \begin{cases} \frac{s}{\epsilon}, & \text{if } \left\|\frac{s}{\epsilon}\right\| < 1 \\ \text{sgn}\left(\frac{s}{\epsilon}\right), & \text{otherwise} \end{cases} \tag{5.11}$$

where e_1 and e_2 are the position and speed errors, σ_{ref} is the inter-vehicle desired separation, and $k_1, \epsilon > 1$.



(a) Inter-vehicle separations

(b) Inter-vehicle separations

(c) Vehicles positions

(d) Vehicles positions

(e) Damage

(f) Damage

Fig. 5.3. Inter-vehicle separations, vehicular positions, and damage data with an attacker-controlled vehicle at position 3 in the platoon. Results shown in (a), (c), and (e) are obtained using the suggested mitigation controller without detection. Results shown in (b), (d), and (f) are obtained using the suggested mitigation controller with detection.

Table 5.1. Simulation Parameters

Vehicle Dynamics		Controller	Detection Filter
normal	attacker		
$F_i^+ = 1$ $\beta = 0.1$ $\alpha = 5$	$F_i^+ = 1$ $\beta = 0.1$ $\alpha_{\text{att}} \geq 5$	$k_1 = 0.1$ $\epsilon = 0.025$	$l_1 = 200$ $l_2 = 600$ $f_{\text{cutoff}} = 0.01$ Hz

To demonstrate the effectiveness of our approach, we consider a five-vehicle platoon with the attacker at position three. The attacker vehicle follows a square-wave acceleration pattern, wherein the attacker applies maximum control effort followed by minimum control effort, with a given frequency f_{att}. Our platooning goals stipulate $\sigma_{ref} = 9$ m and $v_{\text{ref}} = 25$ ms^{-1}, for which each car length $l = 4.5$ m (one car length of separation between cars). The parameters in the dynamic model of the cars, controller, and detection filter are given in Table 5.1. In order to increase attacker power, α_{att} was chosen to be greater than α. This is equivalent to having a more powerful engine. Consequently the maximum acceleration and velocity of the attacker will be equal or higher than those of normal vehicles. Also, we assume that the third vehicle in the platoon is controlled by the attacker.

Figure 5.3 shows the results for an attack case wherein the attacker is of equal power to other vehicles in the platoon ($\alpha_{\text{att}} = \alpha$). We can see clearly that despite the fact that the mitigation controller was used, the damage increased with time, as shown in Fig. 5.3a, 5.3c, and 5.3e, while, on the other hand, engaging the same mitigation controller with detection significantly reduced the accumulated damage, as shown in Fig. 5.3b, 5.3d, and 5.3f.

Despite the ability of our first mitigation controller to significantly reduce collisions, as shown in Fig. 5.3, it should be noted that the leader gives up the reference trajectory once it detects the attacker behind it. That is to say, the platooning is abandoned and the leader, in addition to other followers, are influenced by the attacker-controlled vehicle which, in turn, act as the new reference for the platoon. For that reason, we will modify our suggested mitigation controller in order to avoid that undesirable effect. By engaging our second suggested mitigation, the goal is to gradually increase the inter-vehicle separations until it is possible to switch control of intact vehicles from auto to human drivers and effectively disband the platoon. The first step to this mitigation approach is adjusting the value of $a_{\max}$ used to calculate the commanded acceleration in (5.9) which is modified as follows

$$u_i = \text{sat}\left(\frac{s}{\epsilon}\right)[2k_1 v_{\max} + a_m + \epsilon] \tag{5.12}$$

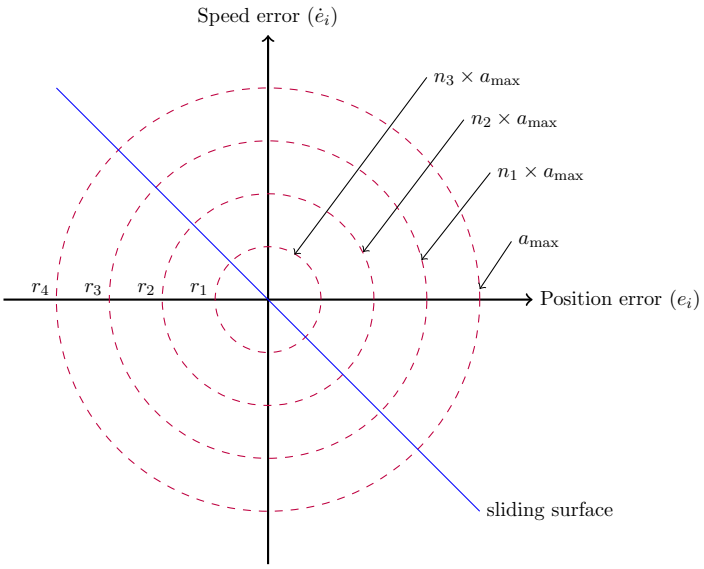


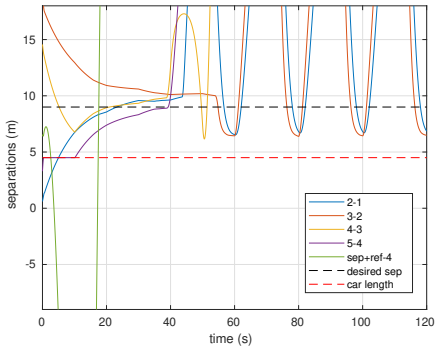
Fig. 5.4. The position - speed errors state space diagram which is employed for our second mitigation controller. Depending on the current values of both errors, the value of a_m, used in (5.12), is modified as $n_j \times a_{\max}$ $(j = 1, 2, 3)$, where $(0 < n_3 < n_2 < n_1 < 1)$. The radii of concentric circles are determined as given in (5.13).

For that purpose, we will use the state space diagram shown in Fig. 5.4, which shows the position-speed errors for the i^{th} platooned vehicle, to determine the value of a_m. The diagram also includes the sliding surface used by the fist mitigation controller. Depending upon the value of σ_{ref}, a number of concentric circles are added to the diagram, their radii are defined as follows

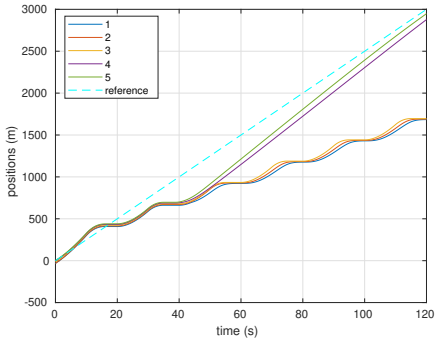
$$\begin{aligned} r_1 &= n_3 \times \sigma_{ref} \\ r_2 &= n_2 \times \sigma_{ref} \\ r_3 &= n_1 \times \sigma_{ref} \\ r_4 &= \sigma_{ref} \end{aligned} \tag{5.13}$$

where

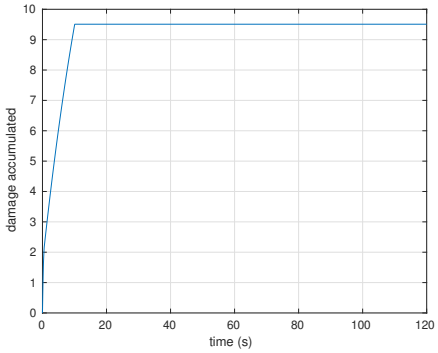
$$0 < n_3 < n_2 < n_1 < 1 \tag{5.14}$$



(a) Inter-vehicle separations



(b) Vehicles positions



(c) Damage

Fig. 5.5**.** Inter-vehicle separations, vehicular positions, and damage data with an attacker-controlled vehicle at position 3 in the platoon. These results are obtained using the second suggested mitigation controller with detection.

At each time step, the values of $e_i(t)$ and $\dot{e}_i(t)$ are calculated. Then, a_m is equal to $n_3 \times a_{\max}$, $n_2 \times a_{\max}$, $n_1 \times a_{\max}$, or $a_{\max}$ if $(e_i(t), \dot{e}_i(t))$ lie inside the circle with radius r_1, r_2, r_3, or r_4, respectively. The second step of this mitigation approach is to gradually increase the inter-vehicle separations until it is safe for the non-attacked vehicles to switch control from auto to human driving and thus disband the platoon, which means the latter vehicles are no longer controlled by the attacker. Also, at each time step, the position error $e_i(t)$ is determined, and if it lies inside the inner circle, which circle's radius is equal to r_1, then the desired inter-vehicle separation increase by 0.5 m. The last action is repeated until the inter-vehicle separation reaches a certain predefined threshold, at which point the disbanding process begins. Fig. 5.5 shows the results for using the second mitigation controller, with which the disbanding begins once the inter-vehicle separation is increased by two meter. We can see that the damage is further reduced, compared with the results

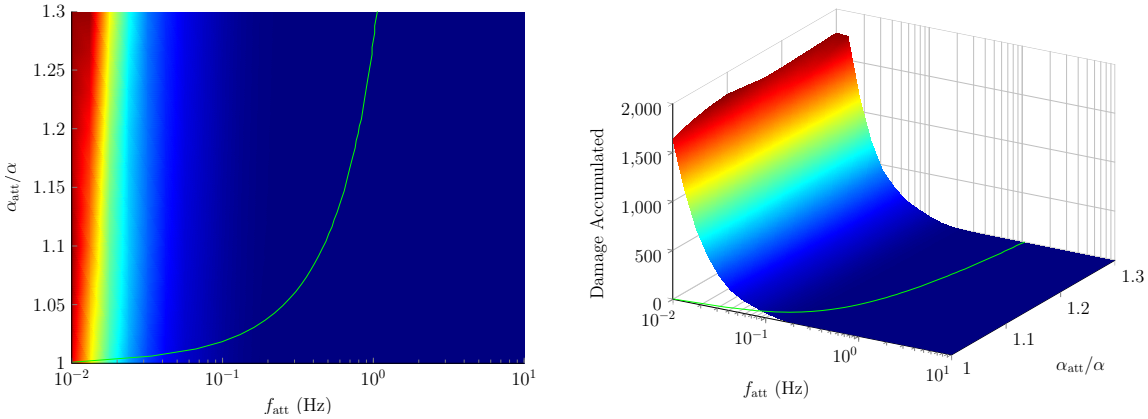


Fig. 5.6. Total damage across relative attacker power and frequencies calculated for a platoon using bidirectional controller without attack detection. Collision line in green [6].

from Fig. 5.3, and the fourth and fifth vehicles do not follow the attacker anymore; they attempt, instead, to recover to the reference trajectory.

5.4 Results Comparison

In this section, we compare the results obtained using linear bidirectional controller, sliding mode controller, and the modified sliding mode controller. In all the results shown in this section, we simulate a platoon with five vehicles, wherein the attacker is controlling the third vehicle. To calculate the effect of an attack, we assign a damage state to the platoon along the lines of 5.8. This damage state begins with a value of zero, and all the collisions' relative velocities are accumulated as the simulation progresses and cars collide. We also define a 'collision line' as follows: Given an attacking and a defending vehicle along with some initial conditions, with both applying maximum effort, it is possible to calculateb the time they collide (t_{col}) using the solution to ($x_{i+1}(t) - x_i(t) = 0$) and ($f_{\text{col}} = \frac{1}{2t_{\text{col}}}$), which is a function of relative attacker power and initial conditions [6]. Above this frequency we can avoid collisions if a suitable control scheme is adopted. In all the results shown in this section, the accumulated damage is calculated across a range of frequencies and a range of relative attacker power. The numbers on the x-axis correspond to the attack frequency while the numbers on the y-axis correspond to the ratio of attacker power over normal vehicle power.

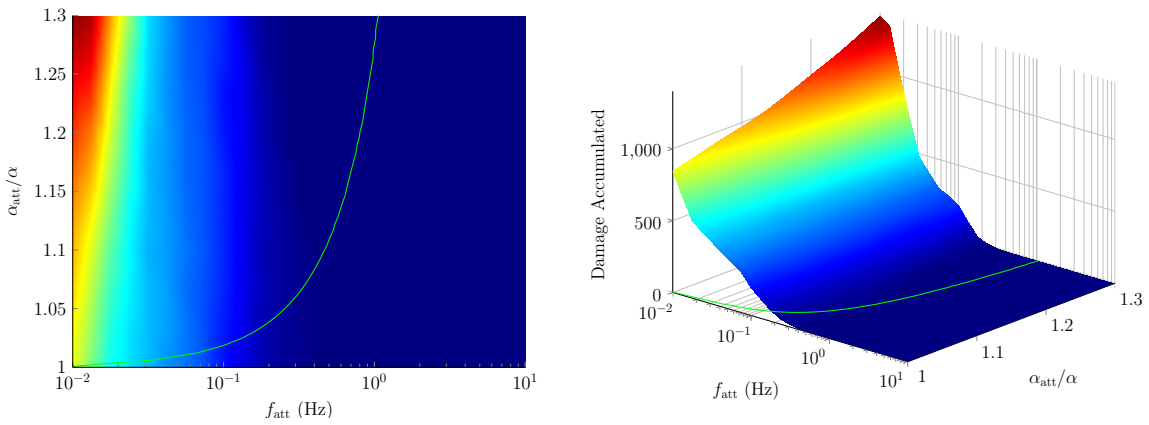


Fig. 5.7. Total damage across relative attacker power and frequencies calculated for a platoon using sliding mode controller without attack detection. Collision line in green [6].

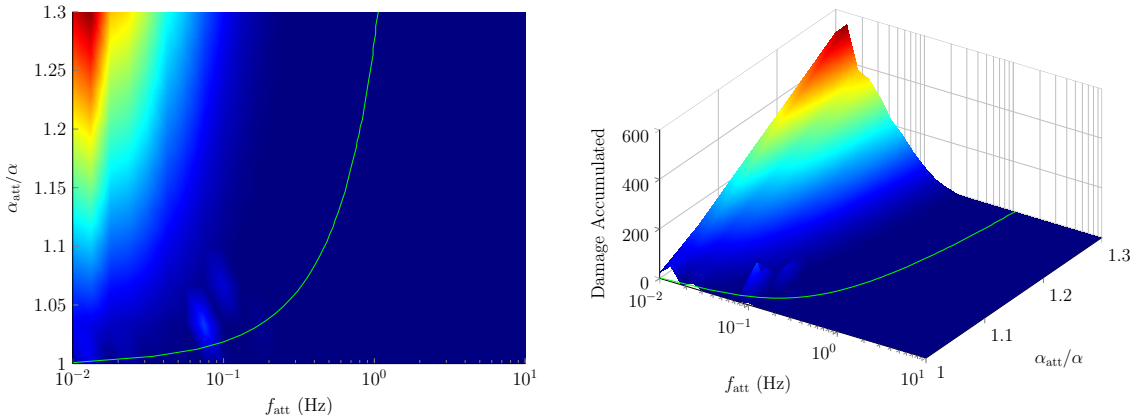


Fig. 5.8. Total damage across relative attacker power and frequencies calculated for a platoon using sliding mode controller with attack detection. Collision line in green [6].

For a reference, we start with total damage measurement using a bidirectional platooning control law, for which the results are shown in Fig. 5.6. The high level controller for this case is described as follows

$$u_i = k_p(x_{i+1} - x_i - \sigma_{\text{ref}}) + k_p(x_{i-1} - x_i + \sigma_{\text{ref}}) + \\ k_d(v_{i+1} - v_i) + k_d(v_{i-1} - v_i) \tag{5.15}$$

with $k_p = 1$ and $k_d = 3$. Fig. 5.6 and 5.7 show the total damage calculated for a platoon using the bidirectional and sliding mode controllers, respectively. We can see that high

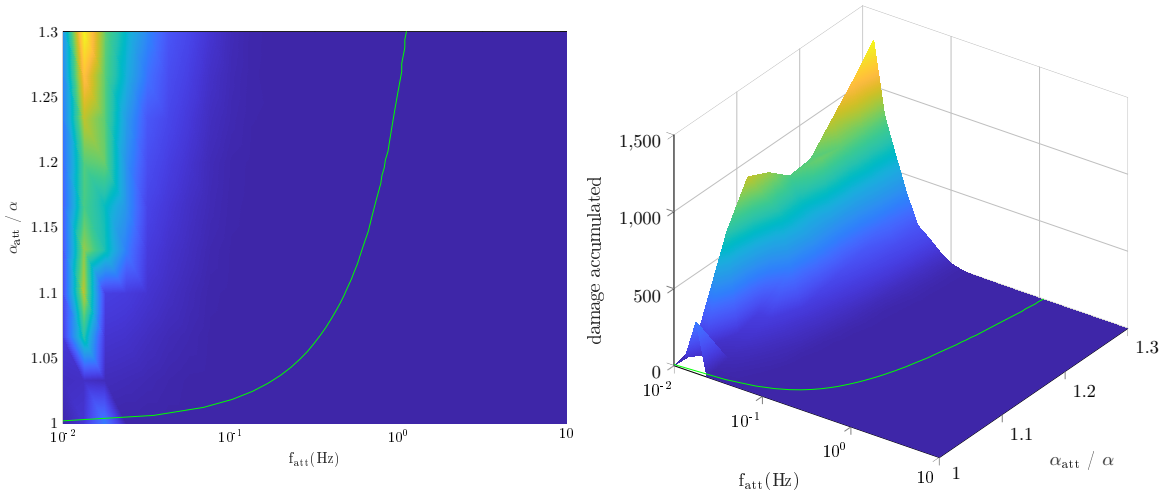


Fig. 5.9. Total damage across relative attacker power and frequencies calculated for a platoon using the modified siding mode controller with attack detection. For these results, disbanding begins when the inter-vehicle separation is increased by two meters.

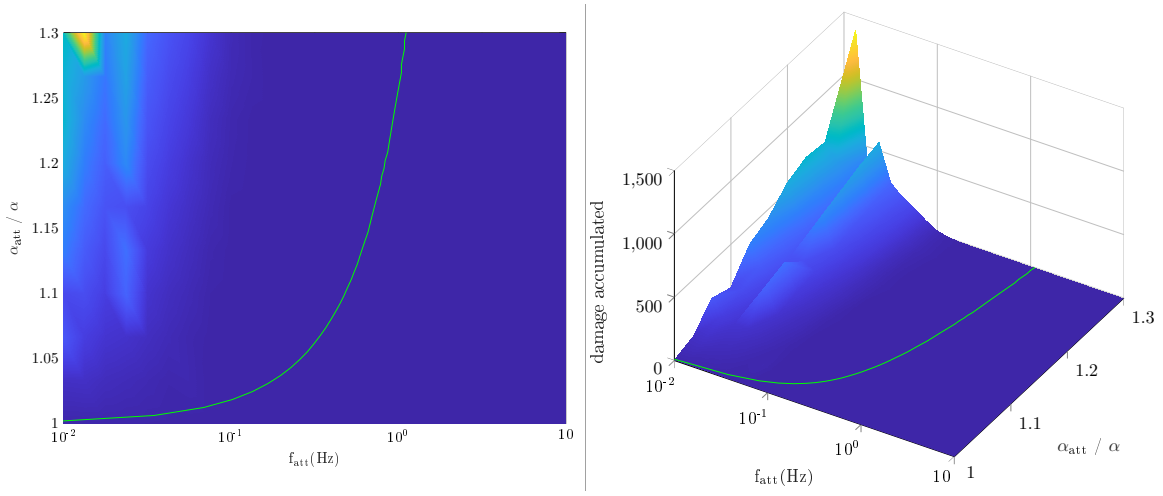


Fig. 5.10. Total damage across relative attacker power and frequencies calculated for a platoon using the modified siding mode controller with attack detection. For these results, disbanding begins when the inter-vehicle separation is increased by four meters.

damage results in both cases at low frequencies and then reduces greatly as the attack frequency increases. Furthermore, Fig. 5.8 shows the total damage for the same platoon, but uses attack detection with the sliding mode controller. Although we still can see high damage at a certain attack frequency and power, the maximum accumulated damage is

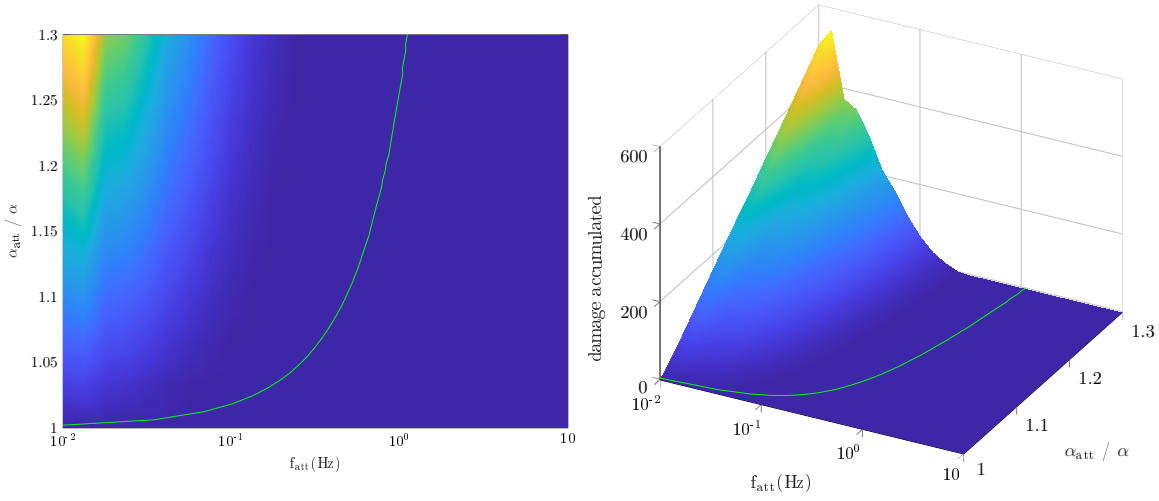


Fig. 5.11. Total damage across relative attacker power and frequencies calculated for a platoon using the modified siding mode controller with attack detection. For these results, disbanding begins when the inter-vehicle separation is increased by seven meters.

lower than that for the two previous cases, which means that adding detection helps with reducing the number of collisions.

On the other hand, Fig. 5.9, 5.10, and 5.11 show the total damage calculated for a vehicular platoon using the modified mitigation controller. As can be seen, the platoon disbands when the inter-vehicle separation, for any of the intact vehicles, reaches two, four, or seven meters with respect to the preceding vehicle. Compared to Fig. 5.8, the results shown in both Fig. 5.9 and 5.10 show higher total damage, even though the attack detection is engaged. This is due to the accomplished inter-vehicle separations which deploy once the disbanding begins, resulting in sudden brakes that can lead to collisions. However, we can see in Fig. 5.11 that the total damage is reduced when compared with the last two cases. In fact, the maximum total damage is nearly equivalent to that of the case shown in Fig. 5.8. The primary difference is that the control of intact vehicles is switched to human drivers, and, thus, the attacker is no longer able to influence the platoon.

5.5 Conclusion

In this chapter, we focused on an existing mitigation controller formulated with the

goal of reducing the collisions that result by attacking a vehicular platoon. Despite success in mitigating attack impacts, engaging the aforementioned controller resulted in a platoon influenced by the attacker. In order to overcome this drawback, we modified the mitigation controller such that the inter-vehicle separation of the intact vehicles increases gradually. Then, after reaching a certain threshold, the automation system of the intact vehicles switch control back to the human driver, hence disbanding the platoon. The last action guarantees that the attacker becomes unable to control the platoon, and the fewest collisions are resulting as possible.

REFERENCES

[1] ""The SARTRE project"," www.sartre-project.net, 2002, [Online; accessed 15-June-2015].

[2] T. Robinson, E. Chan, and E. Coelingh, "Operating platoons on public motorways: An introduction to the sartre platooning programme," in *17th world congress on intelligent transport systems*, vol. 1, 2010, p. 12.

[3] W. Ren and D. Green, "Continuous platooning: a new evolutionary operating concept for automated highway systems," in *American Control Conference (ACC), 1994*, June 1994.

[4] K.-Y. Liang, J. Mårtensson, and K. H. Johansson, "Fuel-saving potentials of platooning evaluated through sparse heavy-duty vehicle position data," *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 1061–1068, 2014.

[5] S. Dadras, R. M. Gerdes, and R. Sharma, "Vehicular platooning in an adversarial environment," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, ser. ASIA CCS '15. New York, NY, USA: ACM, 2015, pp. 167–178. [Online]. Available: http://doi.acm.org/10.1145/2714576.2714619

[6] I. Sajjad, D. D. Dunn, R. Sharma, and R. Gerdes, "Attack mitigation in adversarial platooning using detection-based sliding mode control," in *Proceedings of the First ACM Workshop on Cyber-Physical Systems-Security and/or PrivaCy*, ser. CPS-SPC '15. New York, NY, USA: ACM, 2015, pp. 43–53. [Online]. Available: http://doi.acm.org/10.1145/2808705.2808713

[7] A. Ferrara and C. Vecchio, "Sliding mode control for automatic driving of a platoon of vehicles," in *Variable Structure Systems, 2006. VSS'06. International Workshop on*, June 2006, pp. 262–267.

[8] J. Fax and R. Murray, "Information flow and cooperative control of vehicle formations," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1465–1476, Sept 2004.

[9] H. Tanner, "On the controllability of nearest neighbor interconnections," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 3, Dec 2004, pp. 2467–2472 Vol.3.

[10] R. Sharma, M. Kothari, C. Taylor, and I. Postlethwaite, "Cooperative target-capturing with inaccurate target information," in *American Control Conference (ACC), 2010*, June 2010, pp. 5520–5525.

[11] F. Pasqualetti, R. Carli, A. Bicchi, and F. Bullo, "Identifying cyber attacks via local model information," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, Dec 2010, pp. 5961–5966.

[12] R. Rajamani, *Vehicle Dynamics and Control*, ser. Mechanical Engineering Series. Springer, 2011. [Online]. Available: https://books.google.com/books?id=eoy19aWAjBgC

[13] P. Barooah and J. Hespanha, "Error amplification and disturbance propagation in vehicle strings with decentralized linear control," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, Dec 2005, pp. 4964–4969.

[14] M. Jovanovic and B. Bamieh, "On the ill-posedness of certain vehicular platoon control problems," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 4, Dec 2004, pp. 3780–3785 Vol.4.

[15] D. Swaroop and J. Hedrick, "String stability of interconnected systems," *Automatic Control, IEEE Transactions on*, vol. 41, no. 3, pp. 349–357, Mar 1996.

[16] D. Yanakiev and I. Kanellakopoulos, "A simplified framework for string stability analysis in ahs," in *Proceedings of the 13th IFAC World Congress, 1996*, 1996, pp. 177–182.

[17] S. S. Jackson, "Safety Aware Platooning of Automated Electric Transport Vehicles," Master's thesis, Utah State University, 2013.

[18] J. Fishelson, "Platooning Safety and Capacity in Automated Electric Transportation," Master's thesis, Utah State University, 2013.

CHAPTER 6

CONCLUSION

In this dissertation, we studied the performance of two examples of CPSs while operating in an environment where security and safety-violating attacks can happen. For each CPS, we analyzed the harmful impacts that could be generated by specific a attack against one of the system functionalities. Also, if possible, we suggested potential defenses, with the goal of reducing attack impacts. The first CPS we studied was a vehicular platoon wherein we analyzed attacks against the vehicular range and/or range-rate sensors, which are considered important, as their measurements are utilized by the platooning controller of each vehicle. For this part, we defined a "disbanding attack" with impacts can affect a stream of vehicular platoons, even those which were not specifically attacked. In addition, two solutions were suggested to handle such an attack. Then, we analyzed the impacts of FDI attacks against vehicular platoons using reachability analysis. We also considered an existing mitigation controller that was suggested to handle attacks in which the attacker controls one of the platooned vehicles. Finally, we studied the HVAC systems which used to regulate the indoor temperatures of smart buildings. For this part, we formulated an MPC-based controller to track the desired temperature, and then, assuming the possibility of an attacker with the capability of manipulating the measurements provided by temperature sensors in the building, we suggested MTD-based countermeasures to deter the impacts of such potential attacks. In summary, the following is a list of contributions of each chapter in this dissertation

- For Chapter 2
  - We study the effect of a "disbanding attack" that involves transition of control of multiple vehicles in a platoon. We show the harmful impacts such an attack can

induce, with special focus on how it can cause upstream (non-attacked) platoons to experience slowdowns and collisions.

  - We define the disbanding attack by formulating it as an optimization problem wherein the objective is to maximize the deviation in vehicles' speeds, selected as a proxy for slowdowns and increased chances of colliding, by selecting platoon(s) to be disbanded and time(s) for disbanding.
  - To mitigate the aftermath of such an attack, we formulate an optimal solution using a Model Predictive Control (MPC) technique. However, as the optimal approach is not scalable in practice, since it is centralized and both information and communication-intensive, we also propose a heuristic algorithm to be used locally by vehicles of intact (non-disbanded) platoons. Our findings indicate that our algorithm produces nearly equivalent results in terms of reducing speed changes and avoiding accidents.
  - We also demonstrate the validity of the above attack and the suggested heuristic countermeasures with experiments on a hardware testbed consisting of a motion capture system and small mobile robots acting as vehicles.

- For Chapter 3

  - We analyze the performance of a vehicular platoon undergoing an FDI attack mounted against one or multiple locally-equipped range and range rate sensors.
  - To generalize the problem, we define threat models for targeting either the range sensors, range-rate sensors, or both. Also, we analyze the resulting impacts from those attacks.
  - After defining both the platoon and threat models, wherein the attack vector will act as the new control input to the system, we use the optimal control-based reachability approach to determine the final reachable set by the platoon. This will show whether collision(s) are possible due to an FDI attack.

- For Chapter 4

Assuming the presence of potential attacks against HVAC-equipped sensors, the main contribution of this chapter is to suggest two MTD-based proactive algorithms each of which determines a random set of installed sensors to be used for two tasks: First, a partial measurement of the temperatures in some of the building's zones. Second, a prediction (estimation) of the temperatures in the remaining zones. To implement the second task, we will formulate an optimal state observer that relies primarily on the collected data of inputs and outputs and the randomly-selected sensor set. Achieving the two aforementioned tasks guarantees the availability of the data (measurements) required for the operation of the HVAC system. By continuously selecting a random set at each time step, the attacker's ability to induce predictable effects by targeting one or multiple sensors is minimized.

- For Chapter 5

  In this chapter, our work builds upon the results presented in a previous study and also attempts to solve the safety problem in an adversarial environment. The results shown in that study indicate that the proposed sliding mode controller was able to considerably eliminate or at the very least, to significantly reduce accidents. But though the engagement of the proposed controller resulted in a collision-free platoon, it was still influenced by the attacker-controlled vehicle. For that reason, we modify the proposed sliding mode controller such that it will still be able to avoid collisions and at the same time lead the way to disbanding, switching control from auto to human, of the intact vehicles and, thus, releasing the platoon form the control of the attacker.

APPENDICES

APPENDIX A

Formulation of the MPC based mitigation approach for Chapter 2

Using MPC technique, the controller of the intact vehicle V_i will determine an acceleration required to alleviate disbanding impacts with respect to preceding vehicle V_{i-1}. Let us define the following

$$\begin{bmatrix} e(t) \\ \dot{e}(t) \\ \ddot{e}(t) \end{bmatrix} = \begin{bmatrix} x_i(t) - x_{i-1}(t) - d \\ v_i(t) - v_{i-1}(t) \\ a_i(t) - a_{i-1}(t) \end{bmatrix}, \tag{A.1}$$

(d is the desired separation) which can be rewritten in a discrete-time matrix form as

$$\begin{bmatrix} e(k+1) \\ \dot{e}(k+1) \\ \ddot{e}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T_s & 0 \\ 0 & 1 & T_s \\ 0 & 0 & 1 - \frac{T_s}{\tau} \end{bmatrix} \begin{bmatrix} e(k) \\ \dot{e}(k) \\ \ddot{e}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{-T_s}{\tau} \end{bmatrix} u_i(k) + \begin{bmatrix} 0 \\ 0 \\ \frac{T_s}{\tau} \end{bmatrix} u_{i-1}(k), \tag{A.2}$$

(T_s is the sampling time) which can then be rewritten in a state-space form as

$$\boldsymbol{e}(k+1) = \boldsymbol{A}\boldsymbol{e}(k) + \boldsymbol{B_1}u_i(k) + \boldsymbol{B_2}u_{i-1}(k). \tag{A.3}$$

Then, the objective is to determine the control sequence that will minimize the following quadratic cost function

$$J = \boldsymbol{e}^T(N)\boldsymbol{Y}\boldsymbol{e}(N) + \sum_{k}^{k+N-1} \{\boldsymbol{e}^T(k)\boldsymbol{Q}\boldsymbol{e}(k) + u_i^T(k)\boldsymbol{R}u_i(k)\}, \tag{A.4}$$

subject to the dynamics given in (A.3) and the following state and control constraints

$$\boldsymbol{C_1 e}(k) \leq d \tag{A.5}$$

$$\boldsymbol{C_2 e}(k) \leq v_{\max} - v_{i-1}(k) \tag{A.6}$$

$$\boldsymbol{C_3 e}(k) \leq v_{\min} + v_{i-1}(k) \tag{A.7}$$

$$u_{\min} \leq u_i(k) \leq u_{\max} \tag{A.8}$$

where

$$\boldsymbol{C_1} = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}$$
$$\boldsymbol{C_2} = \begin{bmatrix} 0 & -1 & 0 \end{bmatrix}$$
$$\boldsymbol{C_3} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

State constraints ensures no collisions, (A.5), velocity is within bounds, (A.6) and (A.7), and resulting commanded acceleration is within bounds too, (A.8). Furthermore, $\boldsymbol{Y}$, $\boldsymbol{Q}$, and $\boldsymbol{R}$ are weighing matrices selected to satisfy the following [**?**]

$$\begin{aligned} \boldsymbol{Y} &= \boldsymbol{Y}^T \geq \boldsymbol{0} \\ \boldsymbol{Q} &= \boldsymbol{Q}^T \geq \boldsymbol{0} \\ \boldsymbol{R} &= \boldsymbol{R}^T > \boldsymbol{0} \end{aligned} \tag{A.9}$$

The predicted error vector $\bar{\boldsymbol{e}}$ can be written as a function of the current (measured) error vector $\boldsymbol{e}(k)$ as follows

$$\bar{\boldsymbol{e}} = \bar{\boldsymbol{A}}\boldsymbol{e}(k) + \bar{\boldsymbol{B}}_{\boldsymbol{1}}\boldsymbol{U} + \bar{\boldsymbol{B}}_{\boldsymbol{2}}\boldsymbol{U}_{\boldsymbol{i-1}} \tag{A.10}$$

where

$$\bar{\boldsymbol{e}} = \begin{bmatrix} \boldsymbol{e}(k+1) & \dots & \boldsymbol{e}(k+N) \end{bmatrix}^T$$

$$\bar{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A} & \dots & \boldsymbol{A}^N \end{bmatrix}^T$$

$$\bar{\boldsymbol{B}}_{\boldsymbol{m}} = \begin{bmatrix} \boldsymbol{B_m} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{AB_m} & \boldsymbol{B_m} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{N-1}\boldsymbol{B_m} & \boldsymbol{A}^{N-2}\boldsymbol{B_m} & \dots & \boldsymbol{B_m} \end{bmatrix} \quad \{\text{for m} = 1,2\}$$

$$\boldsymbol{U} = \begin{bmatrix} u_i(k) & \dots, & u_i(k+N-1) \end{bmatrix}^T$$

$$\boldsymbol{U_{i-1}} = \begin{bmatrix} u_{i-1}(k) & \dots & u_{i-1}(k) \end{bmatrix}^T$$

The cost function A.4 can be rewritten in a matrix form as

$$J = \bar{\boldsymbol{e}}^T\bar{\boldsymbol{Q}}\bar{\boldsymbol{e}} + \boldsymbol{U}^T\bar{\boldsymbol{R}}\boldsymbol{U} + \boldsymbol{e}^T(k)\boldsymbol{Q}\boldsymbol{e}(k) \tag{A.11}$$

where

$$\bar{\boldsymbol{Q}} = \begin{bmatrix} \boldsymbol{Q} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \ddots & \dots & \vdots \\ \vdots & \vdots & \boldsymbol{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \boldsymbol{Y} \end{bmatrix}$$

$$\bar{\boldsymbol{R}} = \begin{bmatrix} \boldsymbol{R} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{R} & \dots & \mathbf{0} \\ \vdots & & & \\ \mathbf{0} & \mathbf{0} & \dots & \boldsymbol{R} \end{bmatrix}$$

If (A.10) is substituted into (A.11), then we get

$$J = 2\boldsymbol{M_1 U} + \boldsymbol{U}^T \boldsymbol{M_2 U}, \tag{A.12}$$

where

$$\boldsymbol{M_1} = \boldsymbol{e}^T(k)\bar{\boldsymbol{A}}^T\bar{\boldsymbol{Q}}\bar{\boldsymbol{B}}_{\mathbf{1}} + \boldsymbol{U_{i-1}}^T\bar{\boldsymbol{B}}_{\mathbf{2}}{}^T\bar{\boldsymbol{Q}}\bar{\boldsymbol{B}}_{\mathbf{1}}$$

$$\boldsymbol{M_2} = \bar{\boldsymbol{R}} + \bar{\boldsymbol{B}}_{\mathbf{1}}{}^T\bar{\boldsymbol{Q}}\bar{\boldsymbol{B}}_{\mathbf{1}}$$

It should be noted that few terms are omitted from (A.12) because they are constant. The state and control constraints (A.5)-(A.8) can also be rewritten in a matrix form as

$$\boldsymbol{Ce}(k) \leq \boldsymbol{W} \tag{A.13}$$

and

$$\boldsymbol{U} \leq \boldsymbol{U}_{\max} \tag{A.14}$$

$$-\boldsymbol{U} \leq -\boldsymbol{U}_{\min} \tag{A.15}$$

where

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C_1} & \boldsymbol{C_2} & \boldsymbol{C_3} \end{bmatrix}^T$$
$$\boldsymbol{W} = \begin{bmatrix} d & v_{\max} - v_{i-1}(k) & v_{\min} + v_{i-1}(k) \end{bmatrix}^T$$

$$\boldsymbol{U}_{\max} = \begin{bmatrix} u_{\max} & \dots & u_{\max} \end{bmatrix}^T$$
$$\boldsymbol{U}_{\min} = \begin{bmatrix} u_{\min} & \dots & u_{\min} \end{bmatrix}^T$$

Equation (A.13) can be expressed using the predicted error vector as

$$\bar{\boldsymbol{C}}\bar{\boldsymbol{e}} \leq \bar{\boldsymbol{W}} \tag{A.16}$$

where

$$\bar{\boldsymbol{C}} = \begin{bmatrix} \boldsymbol{C} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{C} \end{bmatrix}$$
$$\bar{\boldsymbol{W}} = \begin{bmatrix} \boldsymbol{W} & \boldsymbol{W} & \dots & \boldsymbol{W} \end{bmatrix}^T$$

If (A.10) is substituted into (A.16), we get

$$\bar{\boldsymbol{C}}\bar{\boldsymbol{B}}_{\boldsymbol{1}}\boldsymbol{U} \leq \bar{\boldsymbol{W}} - \bar{\boldsymbol{C}}\bar{\boldsymbol{A}}\boldsymbol{e}(k) - \bar{\boldsymbol{C}}\bar{\boldsymbol{B}}_{\boldsymbol{2}}\boldsymbol{U}_{\boldsymbol{i-1}} \tag{A.17}$$

Equations (A.14), (A.15), and (A.17) can be written in a compact form as

$$\boldsymbol{M_3}\boldsymbol{U} \leq \boldsymbol{M_4} \tag{A.18}$$

where

$$\boldsymbol{M_3} = \begin{bmatrix} \bar{\boldsymbol{C}}\bar{\boldsymbol{B}}_{\boldsymbol{1}} & \boldsymbol{I} & -\boldsymbol{I} \end{bmatrix}^T$$

$$\boldsymbol{M_4} = \begin{bmatrix} \bar{\boldsymbol{W}} - \bar{\boldsymbol{C}}\bar{\boldsymbol{A}}\boldsymbol{e}(k) - \bar{\boldsymbol{C}}\bar{\boldsymbol{B}}_{\boldsymbol{2}}\boldsymbol{U_{i-1}} & \boldsymbol{U}_{\max} & -\boldsymbol{U}_{\min} \end{bmatrix}^T$$

APPENDIX B

Definitions of Matrices from Chapter 3

$$A_1 = \begin{bmatrix} \mathbf{0}_{(n\times n)} & \boldsymbol{I}_{(n\times n)} \\ \mathbf{0}_{(n\times n)} & \mathbf{0}_{(n\times n)} \end{bmatrix}$$

$$B_1 = \begin{bmatrix} & \mathbf{0}_{(n\times n)} & & & \\ -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \dots & -1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} k_p & 0 & \dots & 0 & k_d & 0 & \dots & 0 \\ -k_p & k_p & \dots & 0 & -k_d & k_d & \dots & 0 \\ \vdots & & & & & & & \\ 0 & 0 & \dots & k_p & 0 & 0 & \dots & k_d \end{bmatrix}$$

$$B_{2,x} = \begin{bmatrix} k_p & 0 & \dots & 0 \\ 0 & k_p & \dots & 0 \\ \vdots & & & \\ 0 & 0 & \dots & k_p \end{bmatrix}$$

$$B_{2,v} = \begin{bmatrix} k_d & 0 & \dots & 0 \\ 0 & k_d & \dots & 0 \\ \vdots & & & \\ 0 & 0 & \dots & k_d \end{bmatrix}$$

$$B_{2,xv} = \begin{bmatrix} B_{2,x} & B_{2,v} \end{bmatrix}$$

$$K_1 = \begin{bmatrix} A_2 & A_2A & \dots & A_2A^{m-1} \end{bmatrix}^T$$

$$K_2 = \begin{bmatrix} B_2 & 0 & \dots & 0 \\ A_2B & B_2 & \dots & 0 \\ \vdots & & & \\ A_2A^{m-2}B & A_2A^{m-3}B & \dots & B_2 \end{bmatrix}$$

$$K_3 = \begin{bmatrix} k_3 & k_3A & \dots & k_3A^{m-1} \end{bmatrix}^T$$

$$K_4 = \begin{bmatrix} I & 0 & \dots & 0 \\ k_3B & I & \dots & 0 \\ \vdots & & & \\ k_3A^{m-2}B & k_3A^{m-3}B & \dots & I \end{bmatrix}$$

$$K_5 = \begin{bmatrix} k_5 & k_5 A & \dots & k_5 A^{m-1} \end{bmatrix}^T$$

$$K_6 = \begin{bmatrix} 0 & 0 & \dots & 0 \\ k_5 B & 0 & \dots & 0 \\ \vdots & & & \\ k_5 A^{m-2} B & k_5 A^{m-3} B & \dots & 0 \end{bmatrix}$$

$$A_{ineq} = \begin{bmatrix} I & -I & K_2 & -K_2 & K_4 & -K_4 & K_6 \end{bmatrix}^T$$

$$\begin{aligned} b_{ineq} = \big[& \boldsymbol{\delta}_{\max} \ -\boldsymbol{\delta}_{\min} \ \boldsymbol{u}_{\max} - K_1 e(0) \ -\boldsymbol{u}_{\min} + K_1 e(0) \\ & \boldsymbol{s}_{\max} - K_3 e(0) \ -\boldsymbol{s}_{\min} + K_3 e(0) \ \Psi - K_3 e(0) \big]^T \end{aligned}$$

APPENDIX C

Definitions of Matrices from Chapter 4

$$\bar{Q} = \begin{bmatrix} Q & \mathbf{0}_{(n\times n)} & \dots & \mathbf{0}_{(n\times n)} \\ \mathbf{0}_{(n\times n)} & Q & \dots & \mathbf{0}_{(n\times n)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{(n\times n)} & \mathbf{0}_{(n\times n)} & \dots & Q \end{bmatrix}$$

$$\bar{R} = \begin{bmatrix} R & \mathbf{0}_{(n\times n)} & \dots & \mathbf{0}_{(n\times n)} \\ \mathbf{0}_{(n\times n)} & R & \dots & \mathbf{0}_{(n\times n)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{(n\times n)} & \mathbf{0}_{(n\times n)} & \dots & R \end{bmatrix}$$

$$M_1 = \begin{bmatrix} I_{(n\times n)} & I_{(n\times n)} & \dots & I_{(n\times n)} \end{bmatrix}^T$$

$$M_2 = \begin{bmatrix} A \\ A^2 + A \\ \vdots \\ A^N + A^{N-1} + \cdots + A \end{bmatrix}$$

$$M_3 = \begin{bmatrix} B & \mathbf{0}_{(n\times p)} & \dots & \mathbf{0}_{(n\times p)} \\ AB+B & B & \dots & \mathbf{0}_{(n\times p)} \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B+\dots+B & A^{N-2}B+\dots+B & \dots & B \end{bmatrix}$$

$$M_4 = \begin{bmatrix} G & \mathbf{0}_{(n\times n)} & \dots & \mathbf{0}_{(n\times n)} \\ AG+G & G & \dots & \mathbf{0}_{(n\times n)} \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}G+\dots+G & A^{N-2}G+\dots+G & \dots & G \end{bmatrix}$$

$$\bar{A} = \begin{bmatrix} CA^{-N} & CA^{-N+1} & \dots & CA^{-1} \end{bmatrix}^T$$

$$\bar{B} = \begin{bmatrix} -CA^{-1}B & -CA^{-2}B & \dots & -CA^{-N}B \\ \mathbf{0}_{(n\times p)} & -CA^{-1}B & \dots & -CA^{-N+1}B \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{(n\times p)} & \mathbf{0}_{(n\times p)} & \dots & -CA^{-1}B \end{bmatrix}$$

$$\bar{G} = \begin{bmatrix} -CA^{-1}G & -CA^{-2}G & \dots & -CA^{-N}G \\ \mathbf{0}_{(n\times n)} & -CA^{-1}G & \dots & -CA^{-N+1}G \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{(n\times n)} & \mathbf{0}_{(n\times n)} & \dots & -CA^{-1}G \end{bmatrix}$$

CURRICULUM VITAE

Ali Al-Hashimi

Career Objective

To obtain a university professor position at a public university in Iraq that will enable the use of technical expertise, leadership, and communication, computer, technical and teaching skills. Areas of interest include: security of cyber-physical systems, secure and reliable control systems, optimal control, and fuzzy logic-based control.

Education

- BSc in Electrical Engineering, University of Basrah 2008.
- MSc in Electrical Engineering, University of Basrah 2011.
- PhD in Electrical Engineering, Utah State Univeristy 2019.

Conference Papers

- The Disbanding Attack: Exploiting Human-in-the-loop Control in Vehicular Platooning, Ali Al-Hashimi, Pratham Oza, Ryan Gerdes, and Thidapat Chantem, in *Security and Privacy in Communication Networks. SecureComm*, 2019 (accepted).

Journal Papers

- Impacts of Constrained Sensing and Communication based Attacks on Vehicular Platoons , Mingshun Sun[1], Ali Al-Hashimi[1], Ming Li, and Ryan Gerdes, in *Proc. IEEE Transactions on Vehicular Technology*, 2019 (submitted).

[1]equally contributing authors.