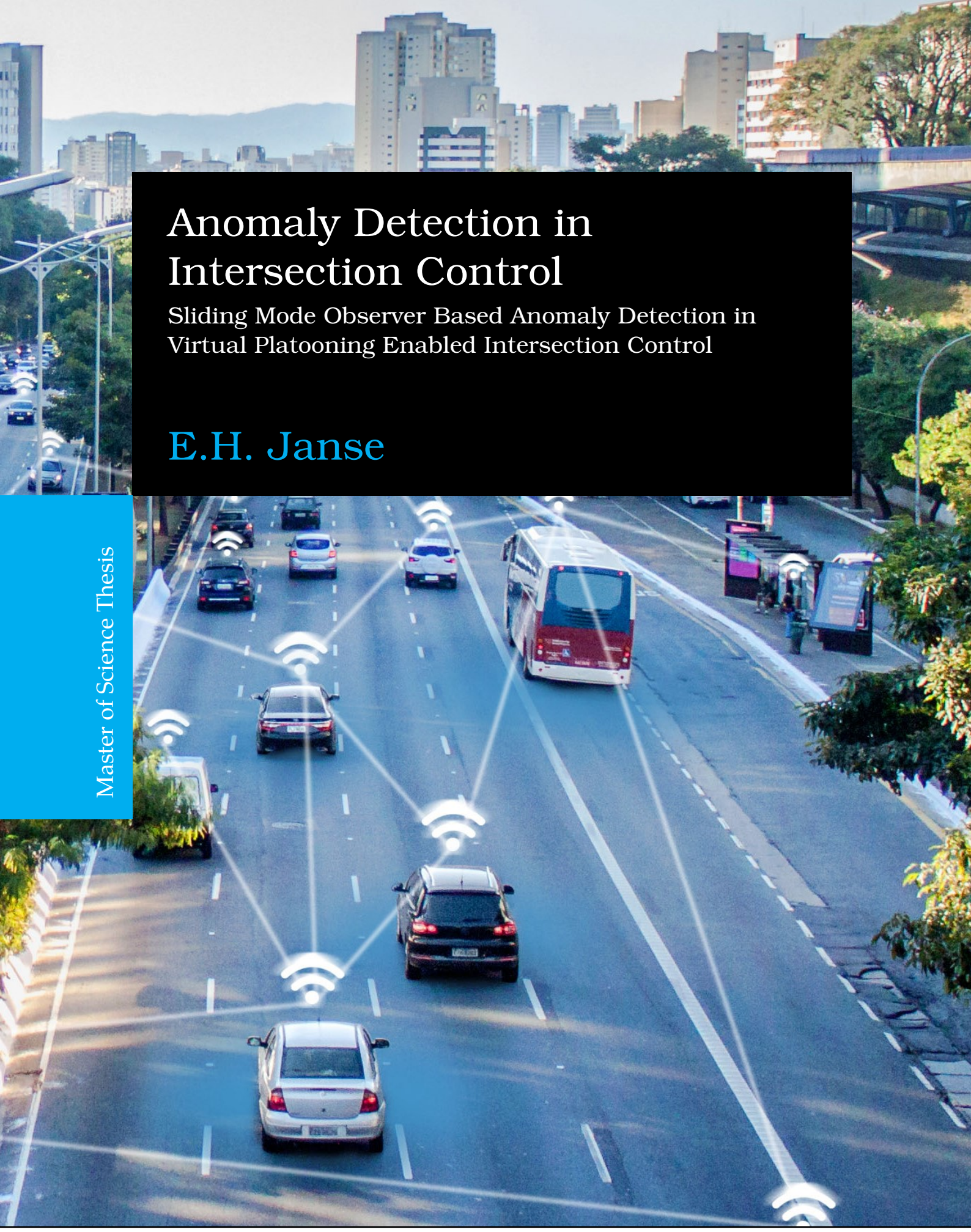


# Anomaly Detection in Intersection Control

## Sliding Mode Observer Based Anomaly Detection in Virtual Platooning Enabled Intersection Control

E.H. Janse

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

# Anomaly Detection in Intersection Control

## Sliding Mode Observer Based Anomaly Detection in Virtual Platooning Enabled Intersection Control

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

E.H. Janse

July 6, 2020

Thesis Committee: Prof.dr.ir. J.W. van Wingerden
Dr.ir. R. Ferrari
Dr.ir. F. Jarmolowitz
Dr.ir. L. Ferranti
Ir. T. Keijzer

*(Image courtesy of the Robert Bosch GmbH Mediabank)*

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis was supported by Robert Bosch GmbH. Their cooperation is hereby gratefully acknowledged.

# Abstract

Road intersections have a large impact on road accidents and travel delay. Applying infrastructure, such as stop signs and traffic lights, are supposed to prevent collisions on the intersection. However, such methods contribute to the travel delay, while accidents still occur due to human errors. Considering the rise of automation within vehicles, the automation of intersections becomes possible. Vehicles exchange information with each other (distributed techniques) or with a central unit (centralized techniques) to determine the velocity profile needed to efficiently cross the intersection without causing any collisions. This information exchange is established through the creation of spontaneous wireless networks between vehicles and infrastructure, also known as a Vehicular Ad-Hoc Network (VANET).

However, the addition of wireless communication to autonomous vehicles presents a dangerous vulnerability to cyber attacks. In the literature, many studies investigate what types of cyber attacks are most likely to occur on the VANET. From these studies, it is chosen to test the cyber attack called "false data injection" on Virtual Platooning based Intersection Control.

It is shown in simulations that an alteration in the broadcasted messages for a duration of 1.3 seconds already leads to a collision on the intersection. In order to prevent such events from happening, a Sliding Mode Observer (SMO) is designed to monitor the received data from the surrounding vehicles. Treating the false injected data as an unknown input to the system, the SMO is designed to reconstruct the anomalous data.

The accuracy of the unknown input reconstruction using an SMO depends on how well the model describes the dynamics of the vehicles. Thus the SMO is designed for both a linear as well as a non-linear vehicle model. To analyze the performance of the SMO, the observer is applied to both simulations and experiments. Under the assumption that the vehicles can measure the relative velocity to each other, both the simulations and the experiments show promising results.

# Preface

This master thesis was executed to conclude the study Systems and Control Engineering at the Technical University in Delft. Upon completing this thesis, the Master of Science degree will be obtained in the field of Control Engineering.

This graduation project was partially performed at Robert Bosch GmbH in Stuttgart, Germany. I would like to thank both my supervisor at Bosch, Dr. Fabian Jarmolowitz and my supervisor at the university Dr. Riccardo Ferrari for this incredible opportunity.

Furthermore, I want to thank my supervisors Dr. Riccardo Ferrari, Dr. Fabian Jarmolowitz and T. Keijzer for their insightful feedback and advise throughout the thesis project.

Finally, I would like to express my gratitude to Niels, Marlein and Flore for providing me with help and feedback on my report, and keeping me motivated throughout the whole project.

**E.H. Janse**
June 2020

# Table of Contents

# List of Figures

# List of Abbreviations

| | |
|---|---|
| **AV** | Autonomous Vehicle |
| **CACC** | Cooperative Adaptive Cruise Control |
| **DoS** | Denial of Service |
| **ETSI** | European Telecommunications Standards Institute |
| **GA** | Genetic Algorithm |
| **IC** | Intersection Control |
| **IM** | Intersection Manager |
| **OBU** | On Board Unit |
| **PKI** | Public Key Infrastructure |
| **RSU** | Road Side Unit |
| **SMC** | Sliding Mode Control |
| **SMO** | Sliding Mode Observer |
| **TPD** | Tamper Proof Device |
| **UIO** | Unknown Input Observer |
| **VANET** | Vehicular Ad-Hoc Network |
| **VP** | Virtual Platooning |
| **V2V** | Vehicle-to-Vehicle |
| **V2X** | Vehicle-to-Everything |

# List of Symbols

| | | | | | |
|---|---|---|---|---|---|
| $\square_i$ | Vehicle $i$ | [-] | $h$ | Time headway | [s] |
| $\square_t$ | Target vehicle | [-] | $H_{obs}$ | Observer sample time | [-] |
| $\Delta u_t$ | Altered signal | [m/s$^2$] | $K_d$ | Aerodynamic drag coefficient | [-] |
| $\delta$ | Inter-vehicle distance | [m] | $L$ | Length vehicle | [m] |
| $\delta_{\text{ref}}$ | Reference inter-vehicle distance | [m] | $m$ | Mass of vehicle | [kg] |
| $\tau$ | Time lag engine dynamics | [s] | $r$ | Stand-still distance | [m] |
| $\theta$ | Communication delay | [s] | $u$ | Control input | [m/s$^2$] |
| $\theta_a$ | Actuator delay | [s] | $u_{t,\text{rec}}$ | Received control input | [m/s$^2$] |
| $a$ | Longitudinal acceleration | [m/s$^2$] | $v$ | Longitudinal velocity | [m/s] |
| $a_{\text{ref}}$ | Reference acceleration | [m/s$^2$] | $v_{\text{ref}}$ | Reference velocity | [m/s] |
| $d_m$ | Mechanical drag | [N] | $x$ | System states | [-] |
| $H$ | Control sample time | [-] | $y$ | System outputs | [-] |

# Chapter 1

# Introduction

It is a well-known fact that junctions have a large contribution to road accidents and travel delay. In 2018, fatalities on intersections made up 20% of all fatal road accidents [21]. To prevent accidents on intersections, traffic lights are placed such that the vehicle flow can be controlled. However, the use of traffic lights cause a delay in travel time, while accidents still occur due to human errors. With the rise of automation in vehicles, concepts to automate intersections have been proposed in the literature. Vehicles exchange information with each other (distributed techniques) or with a central unit (centralized techniques) to determine the velocity profile needed to efficiently cross the intersection without causing any collisions. This information exchange is established through the creation of spontaneous wireless networks between vehicles and infrastructure, also known as a Vehicular Ad-Hoc Network (VANET). Vehicles that use the VANET to exchange messages are referred to as connected vehicles.

Since the automation of intersections relies heavily on wireless communication, a dangerous vulnerability arises. As is the case with any wireless application, wireless communication exposes the vehicles to cyber attacks. An undetected cyber attack on the communication channel can lead to inefficient and even dangerous situations [2, 4, 23, 122]. Therefore, multiple researchers have investigated the effects and the solutions of cyber attacks in VANET, each dedicated to a particular type of cyber attack. The type of cyber attack on the connected vehicles depends on how the VANET is used, and what the autonomy level of the vehicle is. Therefore, the rest of the chapter elaborates on the possible autonomy levels in the vehicle, and the details of the VANET. The chapter is concluded with the research aim of this thesis, and the further structure of the report.

## 1-1 Autonomous Vehicles and its Sensors

The type of communication, and thus the type of cyber attack, depends on the level of autonomy within vehicles. Autonomous vehicles are generally divided into six levels, the first level corresponds with the lowest level of autonomy, and the sixth level with the highest level [22]. The higher the level, the more tasks are transferred from the human driver to the vehicle

| SAE level | Name | Narrative Definition | Execution of Steering and Acceleration/ Deceleration | *Monitoring* of Driving Environment | Fallback Performance of *Dynamic Driving Task* | System Capability *(Driving Modes)* |
|---|---|---|---|---|---|---|
| ***Human driver* monitors the driving environment** | | | | | | |
| 0 | **No Automation** | the full-time performance by the *human driver* of all aspects of the *dynamic driving task*, even when enhanced by warning or intervention systems | Human driver | Human driver | Human driver | n/a |
| 1 | **Driver Assistance** | the *driving mode*-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task* | Human driver and system | Human driver | Human driver | Some driving modes |
| 2 | **Partial Automation** | the *driving mode*-specific execution by one or more driver assistance systems of both steering and acceleration/ deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task* | **System** | Human driver | Human driver | Some driving modes |
| ***Automated driving system* ("system") monitors the driving environment** | | | | | | |
| 3 | **Conditional Automation** | the *driving mode*-specific performance by an *automated driving system* of all aspects of the dynamic driving task with the expectation that the *human driver* will respond appropriately to a *request to intervene* | System | **System** | Human driver | Some driving modes |
| 4 | **High Automation** | the *driving mode*-specific performance by an automated driving system of all aspects of the *dynamic driving task*, even if a *human driver* does not respond appropriately to a *request to intervene* | System | System | **System** | Some driving modes |
| 5 | **Full Automation** | the full-time performance by an *automated driving system* of all aspects of the *dynamic driving task* under all roadway and environmental conditions that can be managed by a *human driver* | System | System | System | **All driving modes** |

**Figure 1-1:** Levels of autonomous driving according to SAE International from [22].

itself. From the third level, the vehicle is responsible for navigating through the environment, and is thus most commonly seen as autonomous driving. A short overview of the levels is shown in Figure 1-1.

To be able to achieve any kind of autonomy, multiple sensors are necessary. These sensors are the ears and eyes of the vehicle, and register the surroundings to enable the possibility of navigating through the environment without human interference. The most important sensors are given in Figure 1-2 and are further explained below.

- *LiDAR*: Light Detection And Ranging (LiDAR) is a mechanism which sends out laser pulses to measure the distance to the surrounding objects. The laser reflects on obstacles and is received back into the LiDAR system. The distance to the object is calculated by using the speed of light and the travel time of the laser. LiDAR sensors have a range of approximately 80 to 160 meters [100].

- *GPS*: Global Positioning System (GPS) communicates with satellites through radio waves containing the Time Of Transmission (TOT). Together with the received signal and the TOT, an exact location can be determined. The precision of the location is around a few centimeters, depending on the quality of the GPS [3].

- *Ultrasound*: Ultrasound sensors detect nearby objects with a range of 5 meters. The sensor emits a mechanically formed sound wave, and calculates the distance using the frequency and travel time of the wave [122].

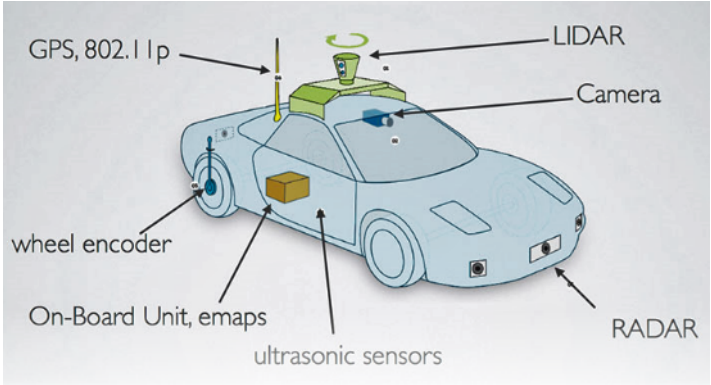



**Figure 1-2:** Important sensors in a vehicle from [103].

- *Radar*: Radars send radio waves which are reflected by objects and are received back into the Radar sensors. Mid Range Radars (MRR) have a range of up to 160 meters, while Long Range Radars (LRR) have a range of up to 250 meters [82, 77].
- *Camera*: Processed camera images enable the tracking of moving objects. However, getting useful data from raw camera images is quite difficult to achieve. This is caused by the constantly moving sensors, in both velocity and orientation. Furthermore, it is difficult to estimate the distance of objects in a two-dimensional (2D) image [56].
- *Wheel encoder*: Wheel encoders converts rotary information to velocity and distance information.

Although the three sensors LiDAR, Ultrasound and Radar have the same working principle, there are notable differences. First of all, the range and resolution of the sensors are different. Furthermore, all three sensors have different operating frequencies, causing different reflections on varying materials. This results in the fact that certain materials are not detected with radar waves, while they are detected with an ultrasound sensor [112].

## 1-2 Inter-Vehicle Communication

The vehicular Ad-Hoc Network enables the inter-vehicle communication by applying the principles of Mobile Ad-Hoc Network (MANET). Vehicles and road infrastructure act as nodes within the network and create a spontaneous wireless connection with each other through On Board Unit (OBU) and infrastructure using Road-Side Units (RSU) [108, 113]. Communication between vehicles is referred to as Vehicle-to-Vehicle (V2V) communication. When vehicles are also able to exchange information with Road Side Units (RSUs), it is referred to as Vehicle-to-Everything (V2X) communication. In Europe, the Vehicular Ad-Hoc Network communication is standardized as a layered protocol in ETSI ITS. The access layer is called ITS-G5 and applies the protocol IEEE 802.11p which uses the frequency band 5.855 - 5.925 MHz. The access layer is divided into a Data Link layer and a Physical layer. The physical layer consists of frequency channels with a bandwidth of 10 MHz each and supports eight transfer rates of which the highest transfer rate equals 27 Mbps. [52].

In the U.S., the standard of VANET is called the Dedicated Short Range Communication (DSRC), applying the IEEE 1609 Wireless Access in Vehicular Environments (WAVE) protocol which is based on the IEEE 802.11p protocol. The operating frequency is 5.9 GHz, divided into seven channels of 10 MHz each.

The level of autonomy in the vehicles influences the possible messages that can be sent via the VANET. In case of human operated vehicles, communication between vehicles may improve road safety by sharing informative messages. These messages, for example, may contain information about crashes ahead, collision warnings, traffic conditions and road conditions. These types of messages are referred to as "alert" or "warning" messages. In the case that the vehicle autonomy is at the third level (see Figure 1-1) and is thus able to automatically adjust its own velocity and acceleration, connected vehicles can share their state information and planned trajectories. Sharing state information, such as the current velocity and intended acceleration, enables the possibility to implement concepts such as Cooperative Adaptive Cruise Control (CACC) and Intersection Control (IC). Both concepts aim to improve the vehicle throughput and road safety. A more detailed description of both concepts is given in Chapter 2.

Wireless networks always pose a vulnerability to cyber attacks, the VANET being no exception. A cyber attack on connected autonomous vehicles could have fatal consequences, thus many studies have been conducted to determine what kind of attacks are possible on the wireless communication channel, and how to prevent and detect them. The effects of a cyber attack depend on the content of the transmitted messages between the vehicles. In the case that the transmitted messages contain alert information (e.g., a roadblock or traffic jam ahead), an attacker —also known as a malicious node— may forge false messages to create situations in their own benefit. If the messages are used to enable the formation of vehicle platoons, forged messages may lead to rear-end crashes by altering the positions or acceleration of the vehicles. Finally, in case of vehicles participating in an Intersection Control protocol, cyber attacks can lead to collisions on the intersection. It is therefore of high importance that prevention and detection techniques of cyber attacks are thoroughly researched.

## 1-3 Research Focus

The Intersection Control heavily depends on wireless communication in order to prevent collisions on the intersection. Receiving incorrect data from the neighbouring vehicles could have fatal consequences. The reasons for receiving incorrect data from neighbouring vehicles could be either intentional or unintentional. For example, if the anomalous data is a result of a sensor fault, the vehicle could unknowingly broadcast incorrect information. However, the most dangerous cause is when a cyber attack on the communication channel goes unnoticed.

Cyber attacks on alert messages and platooning vehicles are already a known research topic. However, no research has yet been conducted on cyber attacks on vehicles executing an Intersection Control protocol, even though intersections are most prone to fatal accidents. Therefore, the scope of this Master Thesis consists of two research questions.

The first objective is:

**If connected autonomous vehicles are participating in an Intersection Control protocol, to what extent should the broadcasted data from the surrounding vehicles be altered in order to cause a collision?**

And secondly:

**Is it possible to detect the presence of anomalous data and reconstruct it, in the case that the transmitted messages from connected autonomous vehicles participating in an Intersection Control concept are subject to data alteration?**

In order to answer both questions, a specific Intersection Control (IC) technique is chosen for further analysis. A short summary of the existing IC techniques, and an elaboration of the chosen technique, is given in Chapter 2. Next, to answer the first research question, an overview of the possible cyber attacks on the communication channel is given in Chapter 3, together with an analysis of the effects of anomalous data on the chosen IC technique. In Chapter 4, the existing detection an prevention techniques are briefly discussed, after which two detection techniques are chosen to answer the second research question. Chapter 4 also includes simulation results on one of the chosen detection techniques. The design of the second detection technique is discussed in detail in Chapter 5, and is tested in simulations in Chapter 6. Chapter 7 provides a performance analysis of the designed observers in experiments. Finally, we provide the final conclusion and discussion of this Master Thesis in Chapter 8.

# Chapter 2

# Intersection Control

Replacing the traffic lights at an intersection with a safer and more efficient solution is a topic that has been researched for many years [27]. Combining autonomous vehicles with V2X (Vehicle-to-Everything) communication provides the possibility to automate intersections. The automation of intersections is in general referred to as Intersection Control (IC). Vehicles share important information such as their current velocity or their time of arrival in order to safely cross the intersection with a minimum travel delay. Many different techniques have been considered, all with the main goal to avoid collisions and maximize the vehicle throughput.

In this chapter, a short summary of the existing IC techniques is given, after which the chosen concept is further elaborated.

## 2-1 State-of-the-Art

In a prior literature research, the different types of IC techniques have been researched and categorized. Figure 2-1 shows a summary of the Intersection Control techniques, divided into four different categories. The first important division is whether the coordination technique is optimization based or non-optimization based. Optimization based techniques translate the IC problem into an objective with constraints. The objective is often formulated as maximizing the vehicle throughput or minimizing energy usage. The constraints typically contain the collision avoidance terms and some limitations on the control input. Optimization based techniques have as advantage that at least a local optimum can be found, implying that the vehicles will efficiently cross the intersection. However, these techniques have the disadvantage that they are computationally heavy to perform, which is caused by the non-convex and non-linear constraints of the optimization objective. Furthermore, it needs to be guaranteed that a feasible solution is found in time in order to avoid collisions on the intersection. Therefore, multiple techniques are suggested to solve the intersection problem without using any optimization techniques.

The second division is based on the communication protocol of the vehicles.

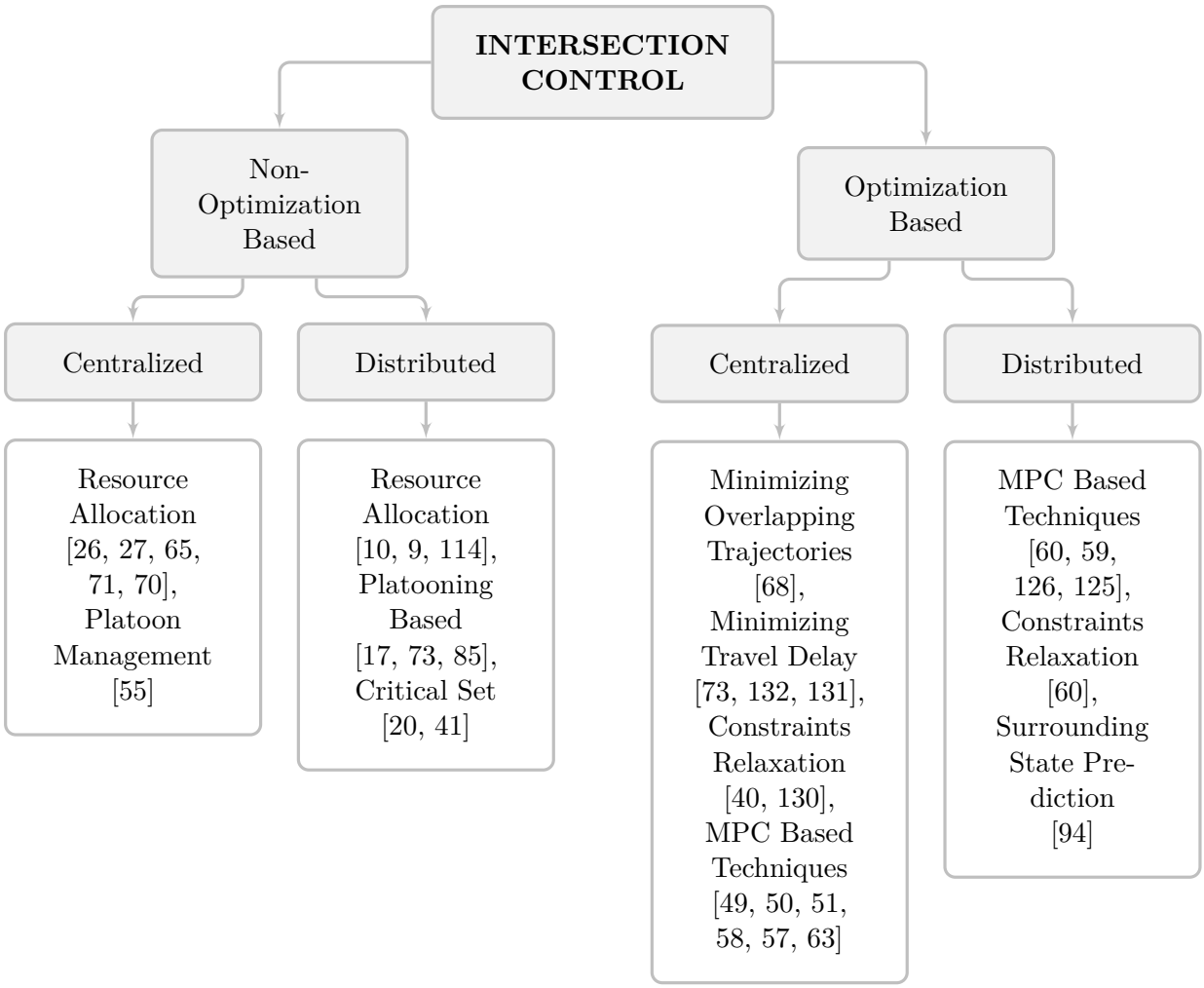



**Figure 2-1:** Overview Intersection Controllers.

In the case that the vehicles communicate to a central node, which makes at least one global decision, the controller is called centralized. The central node is often named the Intersection Manager (IM). Some researches implement a central node by temporarily assigning a vehicle as the IM. The method is referred to as decentralized control in the case that no central node is involved. Below, the often suggested non-optimization based and optimization based techniques are discussed.

From the beginning of IC research, centralized resource allocation was a popular concept. The essential idea behind resource allocation is to divide the intersection into tiles or collision points that can be reserved at a certain time. Upon approaching the intersection, vehicles communicate with the IM about their expected time of arrival and their planned trajectory. Next, some variant of a negotiation process takes place, after which the vehicle receives a reservation for a tile or collision point at a certain time. This idea was also introduced in a distributed manner. Instead of communicating with a central node, the vehicles claim a tile at a certain time, and keep broadcasting their reservation until they passed the tile.

Another interesting concept, both proposed in a distributed and centralized way, is platoon based Intersection Control. A standard platoon describes the situation where vehicles drive in sequence on a highway, and try to keep a small constant distance between each other. In order to maintain a constant distance, the vehicles share their intended control input, which is directly used in the calculation of their own control input. This idea was adjusted and proposed as an IC technique. Approaching vehicles form a so-called Virtual Platooning (VP) where the distance towards the common collision point is used to calculate a safe inter-vehicle distance.

Although the suggested non-optimization based techniques maintain a collision free intersection, many researches argued that such concepts are not efficient in terms of travel delay. This resulted in the idea of an optimization based IC. As mentioned in the introduction, the IC problem is translated into an objective function together with some constraints. Optimization based approaches often give a solution in the form of a schedule containing the arrival times for each vehicle. In case of a distributed IC technique, the optimization problem becomes even more challenging. The vehicles each find a solution to the objective function after which they broadcast their solution (for example their time of arrival) and negotiate until the conflicting arrival times are resolved.

For a more detailed summary on the different IC concepts, the interested reader is referred to the literature study executed prior to this Master Thesis, or to the papers provided in Figure 2-1.

After a thorough literature research, the concept of Virtual Platooning enabled Intersection Control [85] is chosen in this thesis to analyse the effects of the presence of anomalous data in the Vehicular Ad-Hoc Network (VANET). Virtual Platooning is a distributed non-optimization based IC concept. The first decision that needed to be made was between a centralized or distributed controller. Using a centralized approach, no negotiation process is necessary. The central unit collects all needed information to determine the crossing order and trajectories to avoid collisions. However, the central unit must be able to bear the high computational load caused by all the necessary calculations, therefore making it difficult to scale up a centralized control unit in larger scenarios. Furthermore, the process makes the central unit a very crucial node, meaning that a malfunction in the central node will have major impacts. Another disadvantage arises from the high costs of placing and maintaining central control units at each intersection. In a complete decentralized approach, these disadvantages are not present. However, the vehicles do need to negotiate in order to determine the intersection crossing order, which will lead to a higher communication rate. Considering these arguments, a distributed technique was chosen. The control level of the chosen VP technique is decentralized, however, there is a central node present to determine the crossing order of the vehicles. The authors do note that this centralized node should eventually be replaced with a decentralized process.

Another reason for choosing the VP technique for IC, is that it is a non-optimization based technique. The main disadvantage with optimization based protocols is that it cannot be guaranteed that a feasible solution can be found on time, every time, due to the complexity of the objective function with its constraints. Furthermore, even though the Virtual Platooning is not an optimization based protocol, a high vehicle throughput can still be established due to its working principle. The next section gives a more detailed description of the working principles of Virtual Platooning.

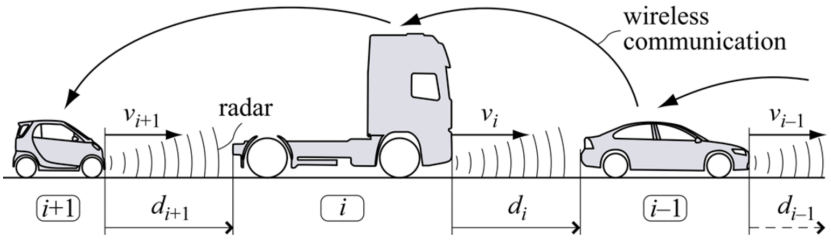



**Figure 2-2:** A platoon of CACC vehicles from [92].

## 2-2 Virtual Platooning based Intersection Control

In order to discuss the working principles of Virtual Platooning, it is first necessary to explain a distributed longitudinal control scheme calledCooperative Adaptive Cruise Control (CACC).

### 2-2-1 Cooperative Adaptive Cruise Control

CACC is an extension of Adaptive Cruise Control (ACC), which is a technique that is used such so that the vehicle automatically keeps a safe distance from the preceding vehicle by using data retrieved from on-board sensors. The smaller the inter-vehicle distance, the higher the vehicle throughput. However, when only sensor data is used, it is difficult to prove an important requirement, called string stability. A platoon of vehicles is called string stable if a disturbance in the velocity of the vehicles decrease as it moves upstream in the platoon [93]. Multiple researches showed promising results for string stability in an ACC enabled platoon [28, 120], but do not account for the actuator and sensor delay, which significantly degrade the string stability [83, 116]. String stability is mathematically defined as

$$\|z_i(t)\|_{\mathcal{L}_p} \leq \|z_{i-1}(t)\|_{\mathcal{L}_p}, \quad \forall t \geq 0,\ 2 \leq i \leq m\,, \tag{2-1}$$

where $z_i(t)$ is either the distance error, velocity or the acceleration of vehicle $i$. $\|\cdot\|_{\mathcal{L}_p}$ denotes the signal $p$-norm. Finally, a total of $m$ vehicles are participating in the platoon where vehicle $i = 1$ is called the platoon leader, (see Figure 2-2).

By sharing the intended control input via the VANET, vehicles can take velocity disturbances of the preceding vehicle into account, and are therefore able to achieve a smaller inter-vehicle distance while maintaining string stability. Most researches express the inter-vehicle distance using a constant time-headway policy, formulated as

$$d_{r,i}(t) = r_i + hv_i(t), \quad 2 \leq i \leq m, \tag{2-2}$$

where $d_{r,i}$ is the desired distance of vehicle $i$ to the preceding vehicle $i-1$, $r_i$ is the standstill distance, $h > 0$ is the time headway and $v_i$ the velocity of vehicle $i$. The time headway $h$ is known to improve string stability [92]. Using this spacing policy, the spacing error $e_i$ is defined as

$$\begin{aligned} e_i(t) &= d_i(t) - d_{r,i}(t) \\ &= (s_{i-1}(t) - s_i(t) - L_i) - (r_i + hv_i(t))\,, \end{aligned} \tag{2-3}$$

where $d_i(t)$ is the true inter-vehicle distance, $s_i(t)$ and $s_{i-1}(t)$ are the positions of the follower vehicle $i$ and the preceding vehicle $i-1$, and $L_i$ is the length of the vehicle.

To design a control law in order to maintain the correct inter-vehicle distance, Ploeg *et al.* [92] use the following linearized vehicle model,

$$\begin{pmatrix} \dot{d}_i \\ \dot{v}_i \\ \dot{a}_i \end{pmatrix} = \begin{pmatrix} v_{i-1} - v_i \\ a_i \\ -\frac{1}{\tau}a_i + \frac{1}{\tau}u_i \end{pmatrix}, \tag{2-4}$$

where $d_i(t)$, $v_i(t)$ and $a_i(t)$ are the inter-vehicle distance, velocity and acceleration of vehicle $i$, respectively. Furthermore $\tau$ is related to the motor dynamics and $u_i(t)$ is the intended control input.

Using Equations (2-3) and (2-4) and together with $e_{1,i}(t) = e_i(t)$, $e_{2,i}(t) = \dot{e}_i(t)$ and $e_{3,i}(t) = \ddot{e}_i(t)$, the error $\dot{e}_{3,i}(t)$ is given by

$$\dot{e}_{3,i} = -\frac{1}{\tau}e_{3,i} - \frac{1}{\tau}q_i + \frac{1}{\tau}u_{i-1}, \tag{2-5}$$

where the input $q_i = h\dot{u}_i + u_i$ should stabilize the error dynamics and compensate for the control input $u_{i-1}$ of the preceding vehicle. Thus a feedback control law is designed as

$$q_i = K \begin{pmatrix} e_{1,i} \\ e_{2,i} \\ e_{3,i} \end{pmatrix} + u_{i-1}, \tag{2-6}$$

where $K = \begin{pmatrix} k_p & k_d & k_{dd} \end{pmatrix}$. Using this control law together with the definition of $q_i$, the control law $\dot{u}_i$ is defined as

$$\dot{u}_i = -\frac{1}{h}u_i + \frac{1}{h}\left(k_p e_{1,i} + k_d e_{2,i} + k_{dd} e_{3.i}\right) + \frac{1}{h}u_{i-1}. \tag{2-7}$$

Extending the error dynamics with this control law leads to

$$\begin{pmatrix} \dot{e}_{1,i}(t) \\ \dot{e}_{2,i}(t) \\ \dot{e}_{3,i}(t) \\ \dot{u}_i(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{k_p}{\tau} & -\frac{k_d}{\tau} & -\frac{1+k_{dd}}{\tau} & 0 \\ \frac{k_p}{h} & \frac{k_d}{h} & \frac{k_{dd}}{h} & -\frac{1}{h} \end{pmatrix} \begin{pmatrix} e_{1,i}(t) \\ e_{2,i}(t) \\ e_{3,i}(t) \\ u_i(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{h} \end{pmatrix} u_{i-1}(t), \tag{2-8}$$

where $k_p$, $k_d$ and $k_{dd}$ are design parameters, $\tau$ represents the motor dynamics, and $u_{i-1}$ is the control input of the preceding vehicle. Since the intended acceleration of the target vehicle $u_{i-1}$ cannot be measured by on-board sensors, it is required that this variable is shared via the VANET.

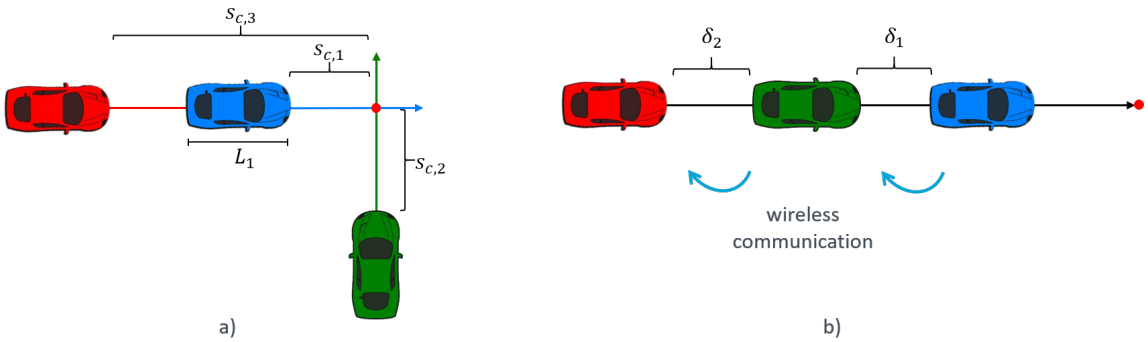



**Figure 2-3:** Virtual Platooning concept. a) Vehicles on distinct trajectories with a common collision point, b) vehicles projected into a virtual platoon, where the distance to the collision point $s_i$ and the length of a vehicle $L_i$ are used to calculate the inter-vehicle distance $\delta_i = s_{c,i} - s_{c,i-1} - L_{i-1}$.

### 2-2-2 Virtual Platooning

Virtual Platooning allows the formation of vehicle platoons, even though the vehicles are on different lanes of the intersection and have different planned trajectories. Virtual Platooning was already a concept suggested for lane merging on highways. In 2015, Morales Medina *et al.* [80] extended the Virtual Platooning concept to T-shaped intersection. They further developed their work in 2017 [85] to a more comprehensive study where they apply Virtual Platooning to any type of intersection. The authors argue that the safe passage of vehicles through the intersection does not only depend on a time schedule, but is mainly achieved by dynamic cooperation between vehicles. In 2019, Castiglione *et al.* [17] modified the concept from [85] to a Multi-Agent System (MAS) and mainly focused on experiments in urban environmental scenarios.

The virtual platoons are formed by projecting vehicles from distinct trajectories onto the same virtual path. Each vehicle follows a preceding vehicle with a small but safe inter-vehicle distance, which is calculated based on the distance to the common collision point. If the preceding vehicle is in reality on a different road, the safe inter-vehicle distance creates a "gap" such that the following and leader vehicle will not collide at the common collision point. The Virtual Platooning concept is schematically represented in Figure 2-3.

The authors from [85] show that it is possible to decouple the vehicle control in a lateral and a longitudinal part. Lateral control focuses on following the planned trajectory while longitudinal control enables the formation of Virtual Platoons. The authors do not focus on the ordering technique and assume a simple First Come First Serve (FCFS). However, they stress that other ordering techniques are more efficient. For example, when a vehicle approaches the intersection, the Target Vehicle Assignment (TVA) protocol checks if the vehicle has a conflicting trajectory. The vehicle that is closest to the common collision point will be the target vehicle due to the FCFS protocol. Castiglione *et al.* [17] propose a similar approach where the leader vehicle is assigned to the vehicle closest to the center of the intersection. However, the technique introduced in Ref. [85] would be more efficient in most scenarios.

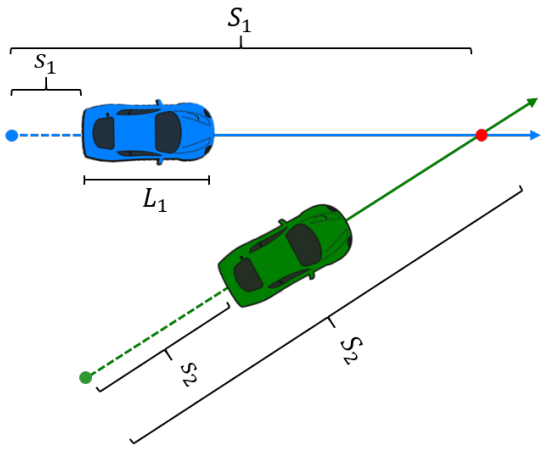



**Figure 2-4:** Locations in Virtual Platooning.

In order to use the CACC concept to form a Virtual Platoon, it is required to define the inter-vehicle distance. Figure 2-4 shows two vehicles approaching a common collision point, indicated by a red circle. The blue and green circles are the entrance points of the respective vehicles of the Intersection Zone, which is an area around the intersection from where the IC protocol and the communication between vehicles takes place. Since the green vehicle is closer to the common collision point, it is assigned to be the target vehicle of the blue vehicle. The inter-vehicle distance is given by

$$\delta = S_1 - s_1 - L_1 - (S_2 - s_2), \tag{2-9}$$

where the variables are as defined in Figure 2-4. Using the distance towards the collision point transforms the two-dimensional coordination problem into a one-dimensional one. Therefore, it is now possible to implement the same control law as with CACC.

The control law that ensures the correct inter-vehicle distance is defined by

$$\dot{u} = \frac{1}{h}\left(u_t - u + k_p(\delta - \delta_{\text{ref}}) + k_d(\dot{\delta} - \dot{\delta}_{\text{ref}})\right), \tag{2-10}$$

where $\delta_{\text{ref}}$ is the reference inter-vehicle distance, $k_p$ and $k_d$ are design parameters and $u_t$ is the intended longitudinal control input of the target vehicle (equal to the index $i-1$ in CACC). Again, the reference inter-vehicle distance is given by $\delta_{\text{ref}} = h + rv$. For $h > 0$, and $k_d > \tau k_p$ the control input from Equation (2-10) exponentially stabilizes the error between the virtual inter-vehicle distance and the virtual reference distance $e = \delta - \delta_{\text{ref}}$, see Ref. [85]. Thus the tracking error dynamics equals

$$\begin{pmatrix} \dot{e}_1(t) \\ \dot{e}_2(t) \\ \dot{e}_3(t) \\ \dot{u}(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{k_p}{\tau} & -\frac{k_d}{\tau} & -\frac{1}{\tau} & 0 \\ \frac{k_p}{h} & \frac{k_d}{h} & \frac{k_{dd}}{h} & -\frac{1}{h} \end{pmatrix} \begin{pmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \\ u(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{h} \end{pmatrix} u_t(t), \tag{2-11}$$

where $e_1 = e$, $e_2 = \dot{e}$ and $e_3 = \ddot{e}$. Even though the state-space of VP is equal to the CACC state-space, an important distinction in the state $e_1$ is present. Instead of representing the

actual inter-vehicle distance, $e_1$ now represent the virtual inter-vehicle distance. Thus, this state cannot be obtained with a frontal radar. The states are defined as

$$\begin{aligned} e_1 &= e = s - s_t + S - S_t - L - \delta_{\text{ref}}, \\ e_2 &= \dot{e} = v - v_t - \dot{\delta}_{\text{ref}}, \\ e_3 &= \ddot{e} = a - a_t - \ddot{\delta}_{\text{ref}}. \end{aligned} \tag{2-12}$$

Virtual platooning in intersections is furthermore also proposed by Castiglione *et al.* [17], who look at the problem as a Multi Agent System (MAS) and use an undirected graph to represent the communication between the vehicles. The authors focus on experimental application of the Virtual Platooning technique established by 5G communication.

# Chapter 3

# Anomalous Data in Connected Vehicles

All previously introduced Intersection Controllers (ICs) heavily depend on wireless communication in order to prevent collisions on the intersection. However, this presents a dangerous vulnerability, where receiving incorrect data have fatal consequences. In order to prevent or mitigate such events it is important to know what the possible causes are.

This chapter explains the causes and types of altered data in connected vehicles, whereafter a specific type of data alteration is chosen for further research. To answer the first research question, the needed data alteration to cause a collision at the intersection is analyzed in simulations. This type of analysis has not yet been performed in IC, and is thus a novel contribution.

## 3-1 Causes of Altered Data

The reasons for receiving incorrect data are classified into two categories: unintentional and intentional causes. In the case that the sending vehicle is transmitting incorrect data, without the interference of a third party, the cause is classified as unintentional. However, in the event that either the sender or a malicious outsider alters the broadcasted information on purpose, one speaks of an intentional cause.

### 3-1-1 Unintentional

An Autonomous Vehicle (AV) may unknowingly send incorrect information due to the presence of undetected faults. Although AVs will be developed such that the presence of an undetected fault is minimal, it is very unlikely that a perfect fault-free AV will exist. The faults may be caused by, for example, uncertain or false sensor readings [7, 97].

### 3-1-2 Intentional

Evidently, no matter what the use case is, wireless communication always induces a sensitivity to cyber attacks. This has resulted in high concerns for the implementation of connected vehicles in general [1, 2, 89, 90, 107]. A broad body of literature exists on the topic of cyber attacks on connected vehicles participating in a platoon, or connected vehicles making use of safety messages. A malicious node may either intercept and modify transmitted date, or broadcast falsely generated data. Not only does this form a problem for concepts such as IC, but also for all vehicles using wireless communication.

The attacker, also referred to as a malicious node, can be classified according to the following categories, as proposed in Refs. [30, 42, 91]

- Outsider versus Insider: An outside attacker is an attacker that lacks the correct authentication keys, and is therefor recognized as an intruder. An inside attacker however is an authorized member of the communication network;
- Irrational versus Rational: An irrational attacker is an attacker who gains no personal benefit in the attack, contradictory to a rational attacker who does seek personal benefit;
- Active versus Passive: An active attacker generates signals or packages in order to harm the connected vehicles. A passive attacker does not affect the network, thereby causing no visible consequences, making it difficult to detect;
- Local vs Extended: This involves the scope of the attack. When only a limited amount of vehicles are affected by the attack, the attack is considered local. In contrast to a local attack, an extended attack involves a greater amount of vehicles (for example a whole intersection, or multiple intersections at the same time).

An overview of possible cyber attacks on the wireless communication channel is given below [4, 30, 89].

- *Denial of Service (DoS)*
  The goal of a DoS attack is to prevent the targeted network from performing its expected function [119]. A typical approach is by flooding the network with packages. As a consequence, the targeted vehicle is occupied with processing the continuous stream of messages, and is thus unable to process "true" messages from other vehicles. The packages that could not be processed in time are dropped, resulting in information loss [91, 30].
- *Ghost vehicle*
  A ghost vehicle is created when a malicious node sends fake messages to surrounding vehicles containing the necessary but fake authorization information. The surrounding vehicles keep an unnecessary distance to the non-existing vehicle, affecting the vehicle throughput [13].
- *Replay attack*
  A malicious node listens to messages transmitted through the VANET and records them for later use. The attacker broadcasts the messages at a desired moment when

the content of the message is not correct anymore. Since the message contains the true authentication keys, the surrounding vehicles will not detect the intruder [4, 30].

- *Sybil attack*
  A Sybil attack is a type of attack where the malicious node forges multiple fake vehicles with a fake GPS signal to inject false data into the VANET. Various safety related messages use an honest majority before broadcasting the messages. Sybil nodes can take advantage of the honest majority by creating multiple false nodes with correct authentication [25, 88].

- *Eavesdropping*
  Vehicles broadcast a variety of messages to their surroundings, containing information about their identity and position. These messages are shared amongst a wireless network, making it sensitive to eavesdropping. Eavesdropping is an attack where a malicious node only listens to the messages, without further interference. Since eaves dropping does not show any (immediate) consequences, they are very difficult to detect [4, 30].

- *Forged alert messages*
  As explained in Chapter 1, the Vehicular Ad-Hoc Network (VANET) can be used to transmit alert messages. Alert messages may contain road conditions or information about road hazards. An attacker may modify these types of messages for their own benefit. The attacker could forge informative messages containing false information about a non-existing traffic jam ahead. Vehicles may decide to not follow the road due to this traffic jam, resulting in an empty road in favor of the attacker [4, 30, 96].

- *False Data Injection*
  A more intelligent type of attack is possible when the vehicles use the state information sent through the VANET for control related computations, for example vehicles forming a platoon or vehicles engaged in IC. A malicious node may intercept messages and modify the content, without the receiving vehicle knowing. Furthermore, a malicious node could imitate the identity of the target vehicle, and transmit false messages. If such an attack is performed while vehicles are forming a platoon, the falsified information leads to perturbations in velocity or rear-end crashes [4, 30, 46].

### 3-1-3 Choice of Altered Data Attack

From all the listed cyber attacks, the Denial of Service (DoS) attack is the most straightforward to execute. The malicious node does not require any knowledge of the control method of the vehicles, or knowledge of the communication protocol. Therefore, a wide body of literature exists on how to deal with DoS attacks in the VANET. The attack type *False Data Injection* does provoke interest amongst researchers [4, 30, 46], but has not yet been researched for IC concepts. The false data transmitted via the VANET could also be the result of a sensor fault at the sending vehicles' side. Therefore, even when the VANET is made extremely resilient against cyber attacks, false data could still be present. Thus, false data injection (also referred to as data alteration) is further researched on the chosen IC concept.

## 3-2 Effects of Anomalous Data on Virtual Platooning

Over the years as the actual implementation of connected vehicles became more realistic, concerns for the privacy and safety of the passengers rose with it. Considering the possible causes listed in Section 3-1, the effects of vehicles receiving anomalous data while using the VANET for a variety of applications have been thoroughly researched. The applications that have been studied so far mostly consist of human operated vehicles using the VANET for safety related messages (see Chapter 1), and autonomous vehicles participating in a platoon enabled by Cooperative Adaptive Cruise Control (CACC). However, the consequences of vehicles receiving incorrect data while obeying an IC concept has not yet been considered. Therefore, this section presents a simple analysis of the effects of vehicles receiving incorrect data while participating in a Virtual Platoon.

In the previous chapter, the choice for a Virtual Platooning (VP) enabled IC protocol was explained. On approaching the intersection, a target vehicle $t$ is assigned to the entering vehicle $i$. Vehicle $i$ keeps a safe inter-vehicle distance to the target vehicle by executing the following control law:

$$\dot{u}_i(t) = \frac{1}{h}\left(u_t(t) - u_i(t) + k_p e_1(t) + k_d e_2(t)\right) \ , \tag{3-1}$$

Again, $e_1$ represents the difference between the virtual inter-vehicle distance and the reference distance, and $e_2$ the relative velocity between the vehicles as shown in Equation (2-12). Furthermore, $u_t$ is the control input of the target vehicle and $u$ the control input of the host vehicle. To calculate the needed control input using Equation (3-1) an important assumption is made, which is elaborated below.

**Assumption 3.1 Measurements.**
*Vehicle i (the host vehicle) can compute the relative (inter-vehicle) velocity with respect to its target vehicle using on-board sensors.* □

**Remark 1**
*Assumption 3.1 is a reasonable assumption in vehicles participating in a CACC enabled platoon, since the distance and relative velocity to the preceding vehicle can be measured with a frontal radar. In the IC case however, this assumption is less reliable. The target vehicle may approach sideways and could be out of view due to an obstacle (e.g., another vehicle).*

Providing that this assumption holds, the only required information from the target vehicle is the control input $u_t$ to calculate Equation (3-1), since the first state $e_1$ can be derived from a regular observer using the measurements for $e_2$. For clarity, a schematic overview is given in Figure 3-1.

The target vehicle implements a simple Cruise Control mode to maintain a constant velocity, given by

$$u(t) = -k_{cc}(v(t) - v_{\text{ref}}) + a_{\text{ref}}, \tag{3-2}$$

where $k_{cc}$ is a design constant, $v(t)$ the velocity of the vehicle, $v_{\text{ref}}$ the reference velocity and $a_{\text{ref}}$ the reference acceleration.

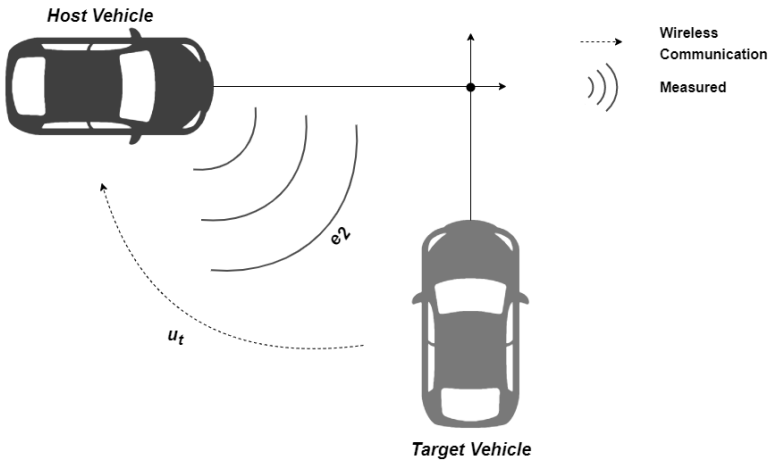



**Figure 3-1:** Measured and received data in Virtual Platooning

### Simulations without data alteration

This section presents the simulation results of a virtual platoon receiving the true value $u_t$. The simulation scenario consists of two homogeneous vehicles approaching an intersection, whose paths intersect at the intersection as shown in Figure 3-1. The target vehicle is the first of the Virtual Platoon, and is thus in Cruise Control mode. The vehicles behave according to a simple linearized vehicle dynamics model where the change in acceleration is represented by a time delay $\tau$. Since the considered vehicles are homogeneous, the time delay $\tau$ is equivalent for both vehicles. The simple vehicle dynamics model is given by

$$\begin{cases} \dot{\delta}(t) &= v_t(t) - v(t), \\ \dot{v}(t) &= a(t), \\ \dot{a}(t) &= -\frac{1}{\tau}a(t) + \frac{1}{\tau}u(t). \end{cases} \tag{3-3}$$

The variable $\delta$ is the inter-vehicle distance, $v_t$ and $v$ the velocity of the target vehicle and host vehicle, respectively, $a$ the acceleration and $u$ the control input.

The same parameters are used as in Ref. [85] to perform the simulation. The parameters of the target vehicle are $k_{cc} = 0.7\,s^{-1}$, $v_{\text{ref}} = 3\,m/s$, $a_{\text{ref}} = 0\,m/s$ and $\tau = 0.1\,s^{-1}$. The parameters of the host vehicle are equal to $r = 3\,m$, $h = 0.3\,s$, $k_p = 0.2\,s^{-2}$ and $k_d = 0.7\,s^{-1}$. The distance to the common collision point for the host vehicle is $S_1 = 39.5$, and for the target vehicle $S_t = 36.5$. Furthermore the following assumptions are made:

**Assumption 3.2 Sensor uncertainties.**
*There are no sensor uncertainties.* □

**Assumption 3.3 Communication.**
*There is no communication delay.* □

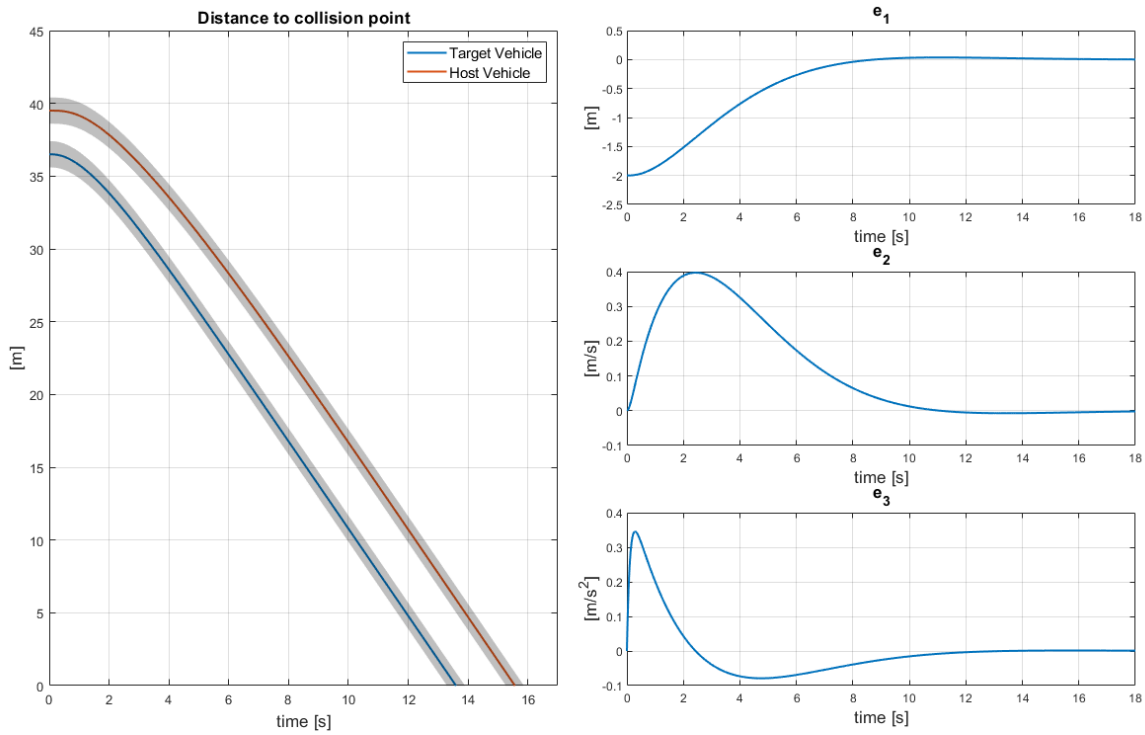



**Figure 3-2:** Left: distance to common collision point, the grey area represents the width of the car. Right: the three states $e_1$, $e_2$ and $e_3$ of the Virtual Platoon, where the host vehicle follows the target vehicle.

**Assumption 3.4 Lateral Control.**
*A different control system is responsible for the path following of the vehicles, which is not considered in these simulations.* □

Figure 3-2 shows the simulation results of two vehicles forming a Virtual Platoon when no attack is present. Both vehicles start from stationary conditions, $v(0) = 0m/s$. The left plot shows the distance towards the common collision point over time. As can be seen, a constant inter-vehicle distance is maintained under the previously defined assumptions and the collision point is safely crossed. The three plots on the right in Figure 3-2 show the three error states $e_1$, $e_2$ and $e_3$ of the Virtual Platoon, defined in Chapter 2, Equation (2-12). All three states converge to zero asymptotically.

### Simulations with data alteration

Under assumption 3.1, the only broadcasted variable using the VANET is $u_t$, therefore this is the only variable that may be incorrect. Jahanshahi and Ferrari [53] investigate the effects of vehicles receiving false data while following a CACC protocol. The authors showed that it is possible to model a wide range of cyber attacks as

$$\Delta u_t = u_{t,\text{rec}} - u_t, \tag{3-4}$$

where the signal $\Delta u_t$ is the difference between the true value $u_t$ and the received value $u_{t,\text{rec}}$ of the target vehicle.

The same concept is applied to Virtual Platooning using different $\Delta u_t$ signals to evaluate what is necessary to cause a collision on the intersection. Using the same simulation configurations as previously described, Figures 3-3 to 3-6 show the effects of a Virtual Platoon receiving incorrect data for different $\Delta u_t$ signals. The top plot in Figure 3-3 shows the distance to the common collision point over time while the host vehicle receives incorrect data. The received value is a summation of $u_t$ and $\Delta u_t = \text{step}(t)$, shown in the bottom plot. The grey areas represents the width of the vehicles, thus overlapping grey areas at the horizontal axis translate into a collision. It is very likely that the vehicles will perform a general check to see if the received value is within physical bounds. Therefore, the step signal has as final value of $4\,m/s^2$ such that the incorrect data would not be detected by the general check. It can be seen that a step signal with size $4\,m/s^2$ needs to be maintained for only 2.5 seconds to cause a crash at the intersection. Clearly, when a smaller step size is chosen it takes more time to cause a collision, as is shown in Figure 3-5.

In Figure 3-4 it can be seen that it is not necessary to maintain the step signal for $2.5\,s$ straight to cause a crash. The bottom plot shows that a step signal with final value $4\,m/s^2$ for only 1.3 seconds is enough for the vehicles to collide at the collision point.

Finally, Figure 3-6 shows the distance towards the collision point when not all consecutive messages are incorrect. The bottom plot shows $\Delta u_t$ over time, which is alternating between $0\,m/s^2$ and $4\,m/s^2$. Clearly this still causes a collision, however, more time is needed.

It must be noted that with this specific IC concept, the timing of the data alteration is important. For example, if the block signal in Figure 3-4 is implemented too early, the vehicle will recover from the unknown input before entering the intersection. This is due to the fact that the control law (3-1) also implements $e_2$, which is assumed to be measurable.

Furthermore, the duration needed to cause a collision depend on the reference inter-vehicle distance $\delta_{\text{ref}} = r + hv$. Choosing a larger stand-still distance $r$ or headway time $h$ will extend the needed duration of the data alteration.

## 3-3 Discussion

In this chapter, the type of anomalous data in connected vehicles was chosen. The chosen type of anomalous data is the alteration of the broadcasted messages from the target vehicle. The alteration of the broadcasted messages includes both the causes from a false data injection cyber attack, as well as the unintentional cause (e.g., the result of a sensor fault).

Furthermore, this chapter showed the needed data alteration in order to cause a collision at the intersection, while the vehicles are following a VP protocol. Even when the simulations are performed in the ideal situation, thus no communication delay, package loss or sensors noise, a data alteration with a duration of only $t = 1.3$ seconds was necessary to cause a collision at the intersection. However, the duration needed to cause a collision does depend on the reference virtual inter-vehicle distance $\delta_{\text{ref}}$.

However, the data alteration needs to be timed correctly in order to cause a collision in the specific case of VP enabled IC. The control law for the VP also uses data retrieved from on-board sensors, besides the received information from the target vehicle. Therefore, if the data alteration is temporarily and not timed correctly, the VP can recover from the received anomalous data.

This chapter showed the dangerous consequences of undetected anomalous data in the VANET. Thus, the next chapter elaborates on the existing prevention and detection techniques in the literature, of which one technique is chosen for further research.

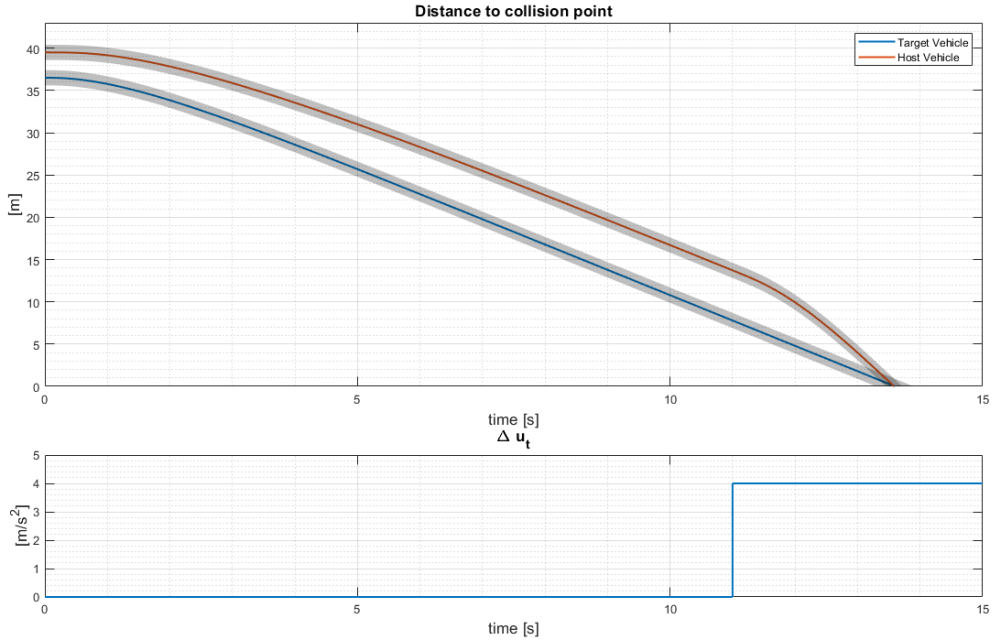



**Figure 3-3:** Top: distance to common collision point where the host vehicle receives $u_{t,\text{rec}} = u_t + \Delta u_t$. The grey area is the width of the vehicles, so overlapping grey areas already represent a collision. Bottom: difference between the true and received value from the target vehicle is a step signal of size $4\,m/s^2$.

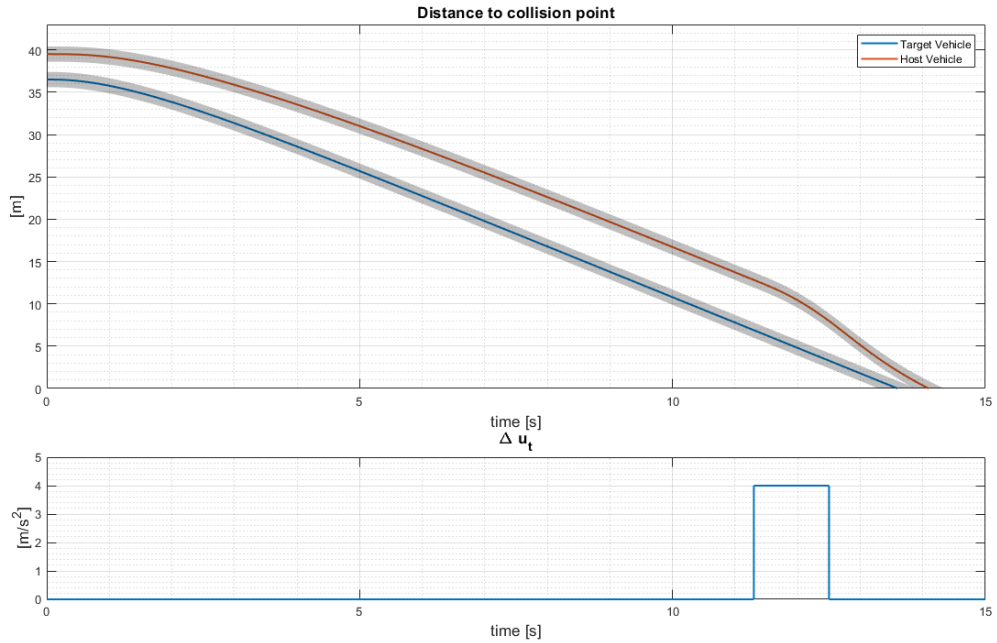



**Figure 3-4:** Top: distance to common collision point where the host vehicle receives $u_{t,\text{rec}} = u_t + \Delta u_t$. The grey area is the width of the vehicles, so overlapping grey areas already represent a collision. Bottom: difference between the true and received value from the target vehicle is a block signal of size $4\,m/s^2$.

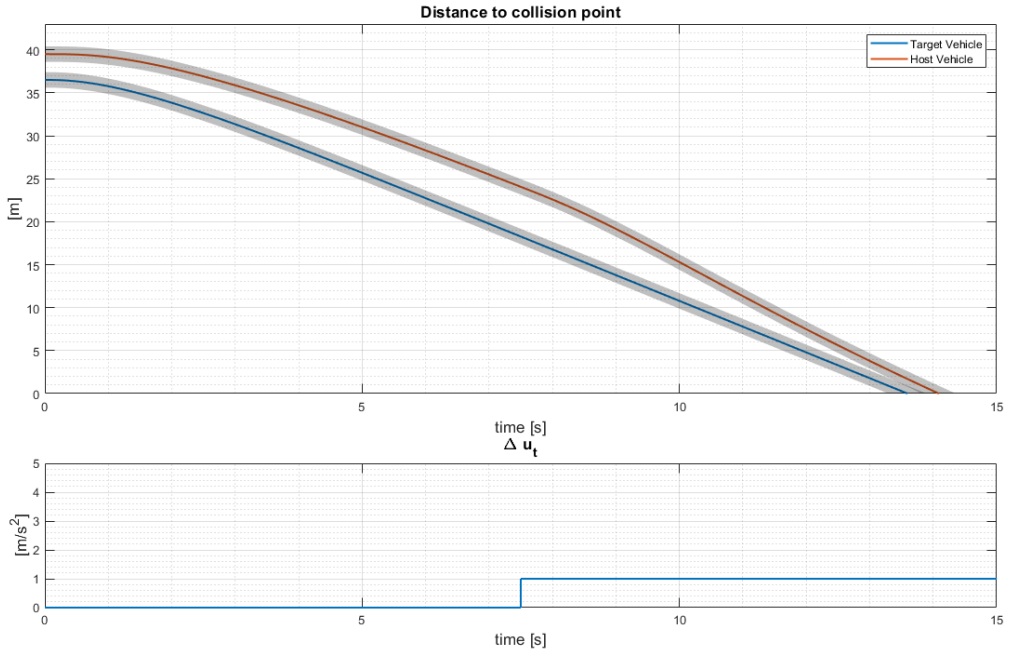



**Figure 3-5:** Top: distance to common collision point where the host vehicle receives $u_{t,\text{rec}} = u_t + \Delta u_t$. The grey area is the width of the vehicles, so overlapping grey areas already represent a collision. Bottom: difference between the true and received value from the target vehicle is a step signal of size $1\,m/s^2$.

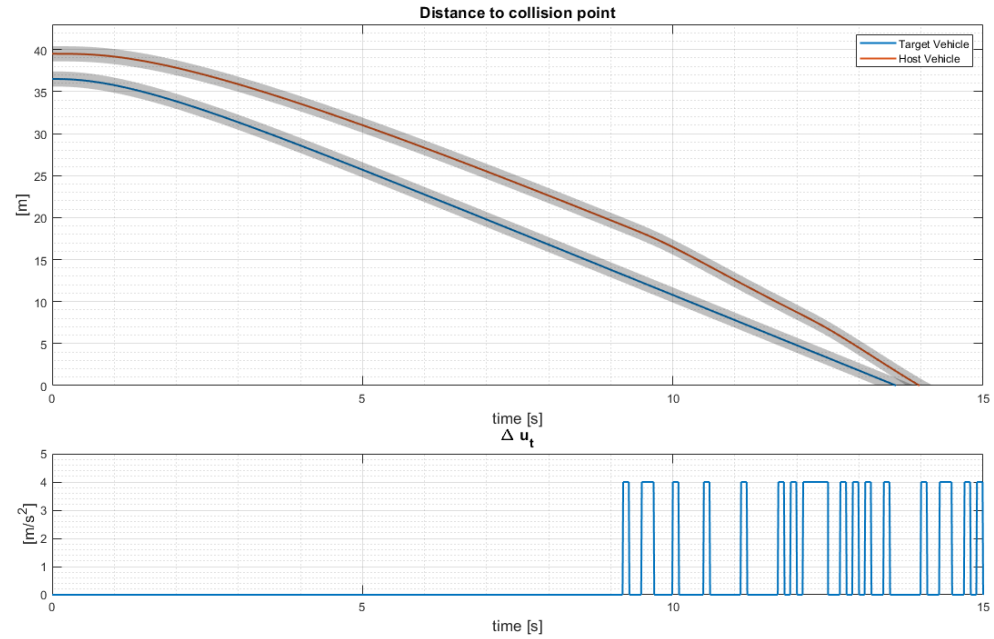



**Figure 3-6:** Top: distance to common collision point where the host vehicle receives $u_{t,\text{rec}} = u_t + \Delta u_t$. The grey area is the width of the vehicles, so overlapping grey areas already represent a collision. Bottom: difference between the true and received value from the target vehicle is a block wave of size $4\,m/s^2$.

# Chapter 4

# Detection and Estimation of False Data Injection Attacks

The previous chapter has shown the dangerous consequences of vehicles using false data in their Intersection Control (IC) protocol. It was shown in simulations that even in an ideal situation, receiving anomalous data for a period of one second could already lead to a collision at the intersection. It is therefore important that multiple safety measures are implemented in order to make it impossible for anomalous data to go undetected.

This chapter starts by shortly discussing the existing prevention and detection techniques in the literature, from which two techniques are chosen to detect the presence of false data in the Virtual Platooning (VP). Next, the state-space of the VP model is extended to explicitly contain the presence of false data.

The extended VP model is used in the design of both the false data detection techniques. First, an elaboration of the most popular technique, a Kalman filter, is given and is tested in simulations. Finally, an introduction for the second detection technique, a Sliding Mode Observer (SMO), is given. The SMO is further used throughout the thesis, of which the reasons are provided in this chapter.

## 4-1 State-of-the-Art and Chosen Concept

In a preliminary literature review as part of this thesis project, the most common procedures were researched on how to handle anomalous data in the Vehicular Ad-Hoc Network (VANET). This section provides a short summary of the found techniques, after which the chosen concept is elaborated.

### 4-1-1 State-of-the-Art

Figure 4-1 shows a summary of the state-of-the-art of the prevention and detection techniques of cyber attacks in the VANET. The techniques are split into two categories, which are in

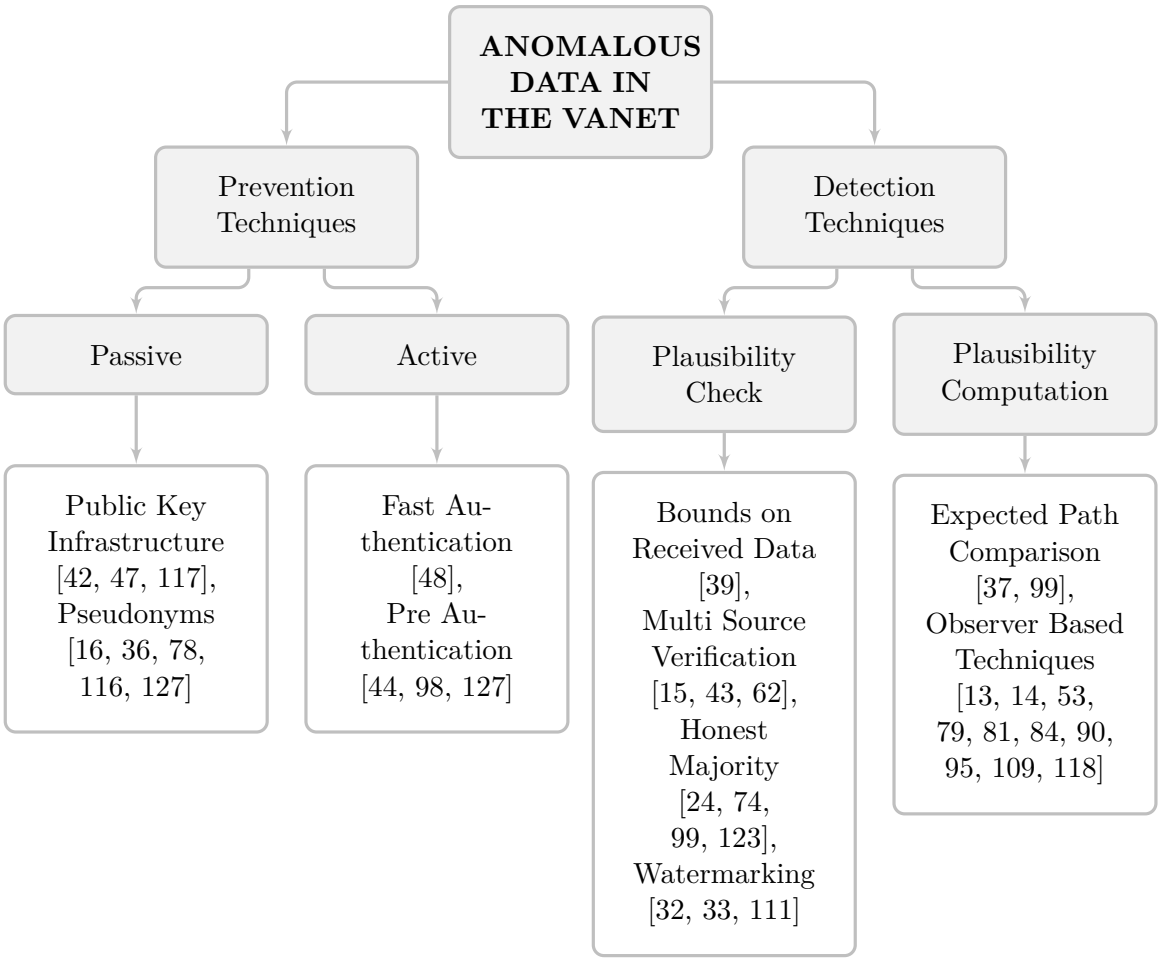



**Figure 4-1:** Schematic literature overview of false data prevention and detection techniques in a wireless network used by vehicles.

turn split into two classes each.

**Prevention Techniques** Prevention techniques mainly focus on keeping malicious nodes outside the communication channel. The prevention techniques require the vehicles to possess some resources, such as a certificate, in order to be included in the communication channel. The main goal of prevention techniques is to make it as difficult as possible for a malicious outside node to listen to transmitted messages and to modify or inject false data.

The prevention techniques are further categorized into two different classes: Passive and Active techniques. Passive techniques are implemented as a precaution in order to hinder a malicious node from achieving its goal. Passive techniques do not change based on the situation, meaning that every received and broadcasted message is subject to the same implemented prevention techniques. Examples that fall into this category are Pseudonyms and a Public Key Infrastructure (PKI). A PKI is an asymmetric cryptosystem, which is already widely used in different communication channels, such as e-mail and the Internet of Things [72, 129]. A Tamper Proof Device (TPD) installed at each vehicle contains encryption keys,

which are used to encrypt the broadcasted data. The encrypted message can only be decrypted with the public key of the same vehicle. This method confirms the identity of the source from the received message. The details on how an efficient PKI system should be implemented in the VANET is still under discussion. A few studies that focus on a PKI in the VANET are Refs. [45, 47, 110, 117].

However, researchers argue that such a PKI system enables malicious nodes to track vehicles on a large scale, as pointed out by Wang et al. [116]:

"*When communicating with each other, vehicles broadcast a range of highly sensitive information, such as position, speed and direction. Thus, if there is a malicious adversary, it can trace the vehicle and a violation of the privacy would happen.*"

Several Studies [16, 36, 78, 116, 127] suggest a procedure that uses a PKI system, but renews the encryption keys at a certain frequency such that the messages of one vehicle cannot be linked together. It has been pointed out, however, that this technique is not practical for vehicles participating in either a platoon or an IC technique, since in those protocols it is necessary to track a single vehicle.

In contrast to passive techniques, active techniques change the procedure based on the current situation. The active prevention techniques listed in Figure 4-1 are specifically developed to prevent a Denial of Service (DoS) attack, which is the most common cyber attack. A typical DoS attack seeks to disable the communication between vehicles by flooding the channel with messages. Since it is not necessary for a successful DoS attack to posses the required encryption keys or certificates, the passive prevention techniques will have no impact on avoiding such an attack. Multiple studies [48, 127, 44, 98] propose the addition of an extra step prior to the verification process. This extra step implements a technique that accelerates the authentication process, enabling the vehicle to filter out messages containing incorrect certificates or decryption keys.

**Detection Techniques** Detection techniques focus on detecting messages containing anomalous data while possessing the required encryption keys or certificates. Reasons for receiving this type of anomalous data could be unintentional e.g., sensor fault in the vehicle transmitting the data), or by a malicious inside node (see Section 3-1). Detection techniques actively screen messages to determine whether the received information can be trusted. The detection techniques are divided into two classes: Plausibility Checks and Computed Plausibility. The Plausibility Checks are in general easier to implement. For example, a simple check may be a comparison between the received data and the physical limits [76, 104]. However, more complex checks are also proposed in the literature. A common strategy is to verify the received data with multiple sources. These sources can either be information from your own sensors or from other vehicles. Combining the information from different sources gives an indication of the confidence level on the received message [15, 62, 37].

Computed Plausibility techniques use a model representing the dynamics of the vehicle(s) to detect an abnormality in the received data. The difference between the model outputs and the measurements is compared with a predefined threshold. If the difference surpasses the threshold, a decision is made to ignore the received message.

An often proposed concept that uses this technique is the Expected Path comparison. This technique is developed to detect a specific cyber attack that could occur in safety related

messages. As explained in Chapter 1, safety related messages are used to warn other vehicles about upcoming road obstacles. Researchers are worried, however, that individuals will inject false messages about non-existing obstacles with the aim to clear the road ahead [96]. Ruj et al. [99], Ghosh et al. [37], and Stübing et al. [106] argue that based on the contents of the safety message, the vehicle transmitting this message should follow a specific path to avoid the indicated obstacle. Using a model of the vehicle dynamics, a prediction can be made of the trajectory that the vehicle should follow in order to avoid the indicated obstacle. The difference between the actual and predicted path is compared with a predefined threshold. If the difference surpasses this threshold, a decision protocol is activated whether to ignore or trust the message.

Other techniques that fall in this category are observer based techniques. A vehicle model is used to compute whether the measurements and received information are coherent. This technique is especially useful if the dynamics of the host vehicle depend on the dynamics of the target vehicle. An example of such a concept is vehicles participating in a platoon or in an IC protocol. In both situations the vehicle receives crucial information in order to keep a safe distance from the target vehicle.

### 4-1-2 Chosen Anomalous Data Detection Technique

In order to implement an efficient IC protocol, it is crucial that the received information can be trusted. Therefore, it is likely that multiple prevention and detection techniques will be implemented in parallel. From the prevention and detection techniques listed in Figure 4-1, one technique is chosen in this thesis to be applied on the VP protocol.

Transmitted anomalous data may be the result of a sensor fault or from an inside attacker. Thus, it is possible that a trusted vehicle unintentionally transmits anomalous data. Furthermore, no matter how secure an online system is, it is still subject to cyber attacks [105]. Although the prevention techniques listed in Figure 4-1 will greatly hinder a malicious node in performing its cyber attack, it will not be able to completely prevent the presence of anomalous data in the VANET.

Furthermore, as shown in Chapter 3, the anomalous data that caused a collision at the intersection was altered such that it was within the physical limits. Therefore, implementing a simple plausibility check on the received data will not filter out these cases. Finally, all suggested IC protocols depend on transmitted data, such as time of arrival or the intended acceleration of the surrounding vehicles. Detection techniques such as Multi-Source Verification verify whether the location of the target vehicle at that time instant can be trusted. Such detection techniques are thus not able to confirm the trustworthiness of the received information, which are essential for collision avoidance. Therefore, it is chosen in this thesis to design an observer to monitor the received information for the presence of anomalous data. In order to incorporate the anomalous data in the observer design, the next section extends the state-space model of the VP such that it contains the presence of anomalous data.

## 4-2 Extended State-Space Virtual Platooning

In order to detect the presence of anomalous data in the chosen IC technique, it is necessary to first modify the Virtual Platooning model. The model needs to be extended such that

it includes the possibility of receiving incorrect data from the target vehicle. As shown in Section 3-2, it is possible to split the received data into two parts

$$u_{t,\text{rec}} = u_t + \Delta u_t. \tag{4-1}$$

The first part $u_t$ represents the true control input of the target vehicle. The second part $\Delta u_t$ is the difference between what is received and what is truly implemented by the target vehicle. When there is no difference between the received and implemented data, it is clear that $\Delta u_t$ is equal to zero. The presence of $\Delta u_t$ will directly influence the dynamics of the third state $e_3$. Substituting Equation (4-1) in the tracking error dynamics of Equation (2-11), the third state $\dot{e}_3 = \dddot{e}$ will extend to the following

$$\begin{aligned}
\dot{e}_3 &= \dddot{\delta} - \dddot{\delta}_{\text{ref}} \\
&= \dot{a}_t - \dot{a} - h\ddot{a} \\
&= -\frac{1}{\tau}a_t + \frac{1}{\tau}u_t - \dot{a} - h\ddot{a} \\
&= -\frac{1}{\tau}a_t + \frac{1}{\tau}u_{t,\text{rec}} - \dot{a} - h\ddot{a} - \frac{1}{\tau}\Delta u_t.
\end{aligned} \tag{4-2}$$

Now implementing the vehicle dynamics model (3-3) and the control law (3-1), the $e_3$ dynamics become

$$\begin{aligned}
\dot{e}_3 &= -\frac{1}{\tau}a_t + \frac{1}{\tau}u_{t,\text{rec}} - \dot{a} - h\left(-\frac{1}{\tau}\dot{a} + \frac{1}{\tau}\dot{u}\right) - \frac{1}{\tau}\Delta u_t \\
&= -\frac{k_p}{\tau}e_1 - \frac{k_d}{\tau}e_2 - \frac{1}{\tau}e_3 - \frac{1}{\tau}\Delta u_t.
\end{aligned} \tag{4-3}$$

Finally, including the control law (3-1) used in Virtual Platooning and the extended $e_3$ dynamics (4-3), the complete Virtual Platooning dynamics are equal to

$$\begin{pmatrix} \dot{e}_1(t) \\ \dot{e}_2(t) \\ \dot{e}_3(t) \\ \dot{u}(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{k_p}{\tau} & -\frac{k_d}{\tau} & -\frac{1}{\tau} & 0 \\ \frac{k_p}{h} & \frac{k_d}{h} & 0 & -\frac{1}{h} \end{pmatrix} \begin{pmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \\ u(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{h} \end{pmatrix} u_{t,\text{rec}} + \begin{pmatrix} 0 \\ 0 \\ -\frac{1}{\tau} \\ 0 \end{pmatrix} \Delta u_t(t), \tag{4-4}$$

which can be written in the compact form

$$\dot{x}(t) = Ax(t) + Bu(t) + D\Delta u_t(t), \tag{4-5}$$

where $x^T(t) = \begin{bmatrix} e_1(t) & e_2(t) & e_3(t) & u(t) \end{bmatrix}$. The extra input $\Delta u_t$ is an unwanted and unknown disturbance to the system that is the result of a cyber attack or a sensor fault at the target vehicle's side. A non-zero value of $\Delta u_t$ is referred to as the unknown input to the system.

## 4-3 Observer Based Anomaly Detection

A common technique to detect sensor or actuator faults is to compare the difference in estimated output $\hat{y}(t)$ and measured outputs $y(t)$ —called the residual— with a threshold. This threshold can be a simple fixed level, or can be a more complicated time varying threshold, that for example incorporates the measurement uncertainty. From Assumption 3.1 it follows that the tracking error of the inter-vehicle velocity $e_2$ can be computed from measurements and on-board sensors. Thus the output of the system (4-4) is given by

$$y(t) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \\ u(t) \end{pmatrix}. \tag{4-6}$$

However, there are multiple advantages in using the estimation of $\Delta u_t(t)$ for anomaly detection, instead of using the output estimate $\hat{y}(t)$. The first and foremost advantage is that the estimate of $\Delta u_t$ can directly be used for both the detection and mitigation process. In the case that the presence of anomalous data is detected, with no knowledge of the size, a very drastic mitigation technique is to come to a complete standstill and not cross the intersection. However, using the estimate of $\Delta u_t$, a less conservative mitigation technique could be implemented that does not require a complete standstill. For example, from the estimate of $\Delta u_t$ the true control input $u_t$ can be derived, as shown in Equation (4-1), which can be used to keep a large distance from the corrupt vehicle without coming to a standstill. Furthermore, the estimate of $\Delta u_t$ gives direct insight into how the received data is altered. Therefore, the patterns of $\Delta u_t$ can later on be analyzed to help classify the incident as a fault or a cyber attack. And finally, the estimate of $\Delta u_t$ is directly used for the anomaly detection, thus, more suitable thresholding techniques can be applied. The threshold now directly represents the size of anomalous data that is not detected, meaning that the control method needs to be robust enough to tolerate anomalous data below the threshold [61].

A popular technique to reconstruct the anomalous data —also referred to as the unknown input— is by using a Kalman filter. The next subsection elaborates on how to use a Kalman filter for anomaly detection, after which a different technique, called SMO, is introduced.

### 4-3-1 Kalman Filter for Anomaly Detection

A popular technique to estimate the unknown variables of a dynamical system is to use a Kalman filter. The unknown input is included in the system as an extra state that needs to be estimated. Since there is no knowledge on how the unknown input is generated, the unknown input is modeled as a so-called random-walk process [115]. In discrete-time, the process is given by

$$\Delta u_t(k+1) = \Delta u_t(k) + w_{\Delta u_t}(k)\,, \tag{4-7}$$

where $w_{\Delta u_t}(k)$ is a white noise sequence with zero mean. This random-walk process indicates that a change in the unknown input is only determined by the white noise sequence $w_f(k)$. Implementing the continuous time version of (4-7) to the system described in (4-4) leads to

$$\begin{aligned}\begin{pmatrix}\dot{x}\\ \Delta\dot{u}_t\end{pmatrix} &= \begin{pmatrix}A & D\\ 0 & 0\end{pmatrix}\begin{pmatrix}x\\ \Delta u_t\end{pmatrix} + \begin{pmatrix}B\\ 0\end{pmatrix} u_{t,\text{rec}} + \begin{pmatrix}w\\ w_{\Delta u_t}\end{pmatrix},\\ y &= \begin{pmatrix}C & 0\end{pmatrix}\begin{pmatrix}x\\ \Delta u_t\end{pmatrix} + v,\end{aligned} \tag{4-8}$$

where $w$ is the process noise of the normal states $x$ and $v$ is the measurement noise. Furthermore, $x$, $A$, $D$ and $C$ are as defined in Equations (4-4) and(4-5). To implement the Kalman filter for the reconstruction of the anomalous data, it is required to define the covariance matrices of the process and measurement noise. The covariance matrix of the process and measurement noise in continuous time is given by

$$E\left(\begin{bmatrix}w(t)\\ w_{\Delta u_t}(t)\\ v(t)\end{bmatrix}\begin{bmatrix}w^T(\tau) & w_{\Delta u_t}^T(\tau) & v^T(\tau)\end{bmatrix}\right) = \begin{bmatrix}Q & 0 & S\\ 0 & Q_{\Delta u_t} & 0\\ S^T & 0 & R\end{bmatrix}\delta(t-\tau). \tag{4-9}$$

The extent to which $\Delta u_t$ is accurately estimated depends on the covariance matrix $Q_{\Delta u_t}$. The simulation result in the next subsection shows the influence of the covariance matrix $Q_{\Delta u_t}$ on the $\Delta u_t$ estimation.

To be able to use the extended state-space (4-8) to reconstruct the unknown input $\Delta u_t$, it is required that the pair $\left(\bar{A}, \bar{C}\right)$ is observable, where $\bar{C} = \begin{pmatrix}C & 0\end{pmatrix}$ and $\bar{A} = \begin{pmatrix}A & D\\ 0 & 0\end{pmatrix}$. The observability is determined by verifying the rank of the observability matrix $\mathcal{O}$.

**Definition 4.1**
The system (4-8) is observable if

$$\text{rank}(\mathcal{O}) = n, \tag{4-10}$$

where $n$ is defined as $\begin{pmatrix}x\\ \Delta u_t\end{pmatrix} \in R^n$. □

The rank of the observability matrix, in our case, is equal to

$$\text{rank}\left(\begin{bmatrix}\bar{C}\\ \bar{C}\bar{A}\\ \bar{C}\bar{A}^2\\ \bar{C}\bar{A}^3\\ \bar{C}\bar{A}^4\end{bmatrix}\right) = 5. \tag{4-11}$$

Since $\begin{pmatrix}x\\ \Delta u_t\end{pmatrix} \in R^5$, the rank in Equation (4-11) is equal to $n$, and thus $(\bar{A}, \bar{C})$ is observable.

### Simulation Result Without Measurement Noise

Using the design from the previous section, we performed a simulation in Simulink to test the unknown input reconstruction with and without measurement noise. The simulation consists of two homogeneous vehicles approaching a common collision point, as can be seen in Figure 3-1. The dynamics of the vehicles are simulated according to the vehicle model described in Equation (3-3). The unknown input $\Delta u_t$ is equal to a step signal, initiated at $t = 2\,s$, with an initial value of $0\,m/s^2$ and a final step value of $1\,m/s^2$.

*Target vehicle*
The target vehicle implements a simple Cruise Control control law, given by

$$a = -k_{cc}\,(v - v_{\text{ref}}) + a_{\text{ref}}, \tag{4-12}$$

where $k_{cc} = 0.2$, $v_{\text{ref}} = 3\,m/s$ and $a_{\text{ref}} = 0\,m/s^2$. The parameters of the vehicle are $\tau = 0.1$, $L = 2\,m$ and $S_t = 38.5\,m$, where $\tau$ is related to the motor dynamics, $L$ the length of the vehicle and $S_t$ the distance towards the common collision point.

*Host vehicle*
The host vehicle follows the target vehicle using the Virtual Platooning control law. Since the two vehicles are homogeneous, the host vehicle has the same constant related to the motor dynamics $\tau = 0.1$, and length $L = 2\,m$. Furthermore, the parameters for the control law are $k_p = 0.2$, $k_d = 0.7$, $h = 0.3$, $r = 3$ and the distance to the collision point is $S = 41.5\,m$. The control law is given by

$$\dot{u} = \frac{1}{h}\left(u_t - u + k_p(\delta - \delta_{\text{ref}}) + k_d(\dot{\delta} - \dot{\delta}_{\text{ref}}\right), \tag{4-13}$$

where the inter-vehicle distance is defined as $\delta = s_t - s - S_t + S - L$ and the reference inter-vehicle distance $\delta_{\text{ref}} = r + hv$.

As previously described, the process noise covariance matrix has a large influence on how accurate the unknown input is reconstructed. The value of $Q_{\Delta u_t}$ translates to how uncertain the model (4-7) for the $\Delta \dot{u}_t$ dynamics is. A $Q_{\Delta u_t}$ close to zero implies that there is no uncertainty in the $\Delta u_t$ model, meaning that there is no change in $\Delta u_t$ over time. The higher the value for $Q_{\Delta u_t}$, the more uncertain the model is, meaning that the estimate for $\Delta u_t$ may vary more over time. To illustrate this effect, the $\Delta u_t$ estimation is shown for two different values for $Q_{\Delta u_t}$. Furthermore, the state vector $x$ is modelled without process noise, thus the covariance matrix is set close to zero $Q = 10^{-20} I^{4\times 4}$. Figures 4-2 and 4-3 show the state estimates and the $\Delta u_t$ estimation, respectively, where $Q_{\Delta u_t} = 1$. Figures 4-4 and 4-5 show the same estimates with $Q_{\Delta u_t} = 100$. When comparing the estimation of $\Delta u_t$ with different $Q_{\Delta u_t}$ values, it can be seen that the rise time of the $\Delta u_t$ is smaller with a higher $Q_{\Delta u_t}$ value.

### Simulation Result With Measurement Noise

The choice of $Q_{\Delta u_t}$ also influences the sensitivity to measurement noise. A White Gaussian noise with zero mean and variance $\sigma = 0.0367$ was added to the relative velocity measurement $e_2$. The variance was taken from the product sheet of a Mid-Range velocity radar developed by

Bosch [38]. Figures 4-6 and 4-7 show the state estimates and $\Delta u_t$ estimation where $Q_{\Delta u_t} = 1$, while measurement noise is present. Figures 4-8 and 4-9 show the state and $\Delta u_t$ estimation, again with measurement noise present, but with a process noise covariance matrix equal to $Q_{\Delta u_t} = 100$. In Figure 4-7, where $Q_{\Delta u_t} = 1$, it can be seen that while the measurement noise is present, the estimation of the unknown input is still reliable. With $Q_{\Delta u_t} = 100$, the unknown input $\Delta u_t$ estimation becomes noticeably more affected by the noise, as shown in Figure 4-9. However, it is still possible to detect the presence of anomalous data from the unknown input estimation.

### 4-3-2 Sliding Mode Observer for Anomaly Detection

Using a Kalman filter to reconstruct the unknown input depends on a linear vehicle model. However, the research that introduces the VP-based IC, [85] uses the vehicle model from Ref. [101], who states the following:

"*This simple model used to describe the engine dynamics has proved to be useful for preliminary system level studies in longitudinal control of a platoon of vehicles,*"

thereby indicating that the simple vehicle model should eventually be replaced with a more suitable non-linear vehicle model. Therefore, a technique called Sliding Mode Observer (SMO) is researched to reconstruct the unknown input. An SMO has as main advantage that it is not limited to a linear model. After designing a linear SMO, only minor adjustments are needed to transform the observer to a non-linear SMO.

An SMO is a non-linear high gain state observer, which pushes the dynamics of the observer towards a predefined sliding surface. The sliding surface is represented by the difference between the estimated and measured outputs (the residual), and is set to zero. The observer feedback that pushes the dynamics towards this zero residual surface is often defined as the sign of the surface. Since the design of the SMO is complex, the next chapter is devoted to the design of both the linear as well as the non-linear SMO.

## 4-4 Discussion

In this chapter, an overview was given of the state-of-the-art on prevention and detection techniques of cyber attacks on connected vehicles. The VP model was extended such that the effect of the anomalous data explicitly appears in the model. And finally, Kalman filter was designed to reconstruct the unknown input. As a contribution, it was shown in this chapter that a Kalman filter is able to reconstruct the unknown input $\Delta u_t$. Furthermore, it was shown that the process noise covariance matrix $Q_{\Delta u_t}$ needs to be high in order to reconstruct the unknown input, which has as a disadvantage that the measurement noise is amplified.

However, the vehicle model used in simulations to estimate the unknown input with a Kalman filter, is a simple linear model. As stated in Section 4-3-2, the linear vehicle model will eventually be replaced with a more suitable non-linear vehicle model. Therefore, the next chapter elaborates on the design of an SMO to reconstruct the unknown input, since the SMO has much more flexibility in the observer design.

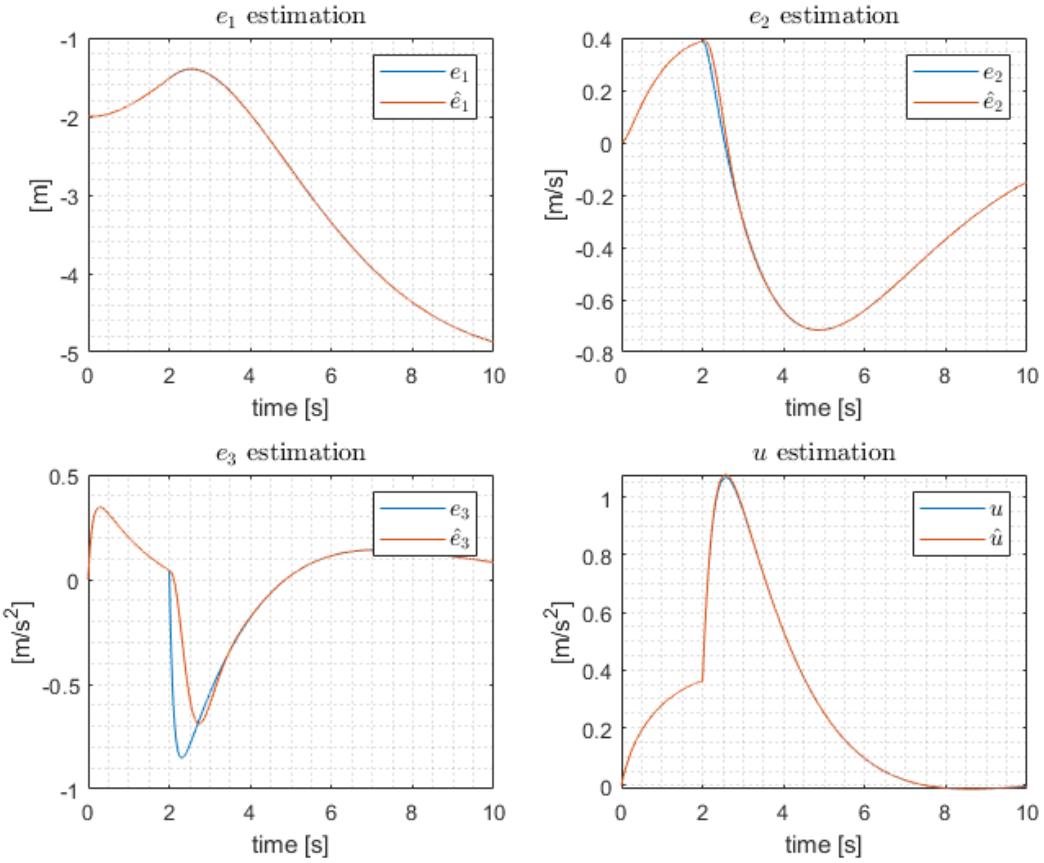



**Figure 4-2:** Estimation of the state vector using a Kalman filter while an unknown input is present, without measurement noise. The process noise covariance matrix $Q_{\Delta u_t} = 1$.

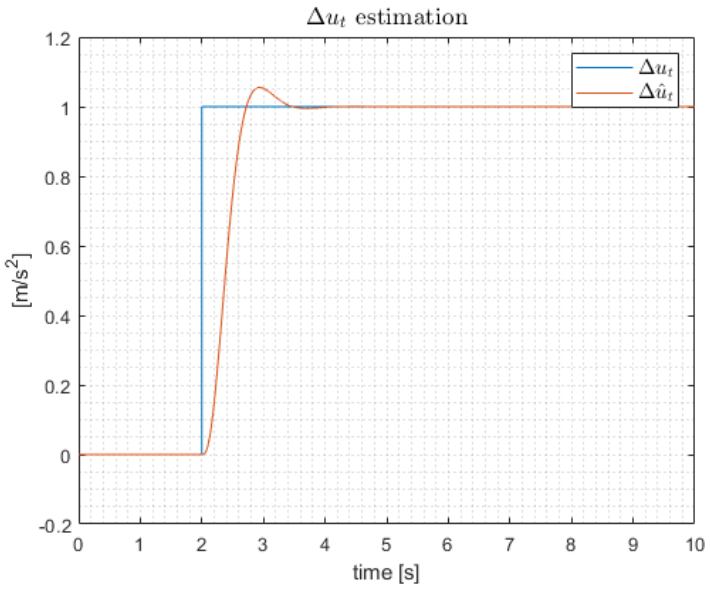



**Figure 4-3:** Estimation of the unknown input using a Kalman filter without measurement noise. The process noise covariance matrix $Q_{\Delta u_t} = 1$.

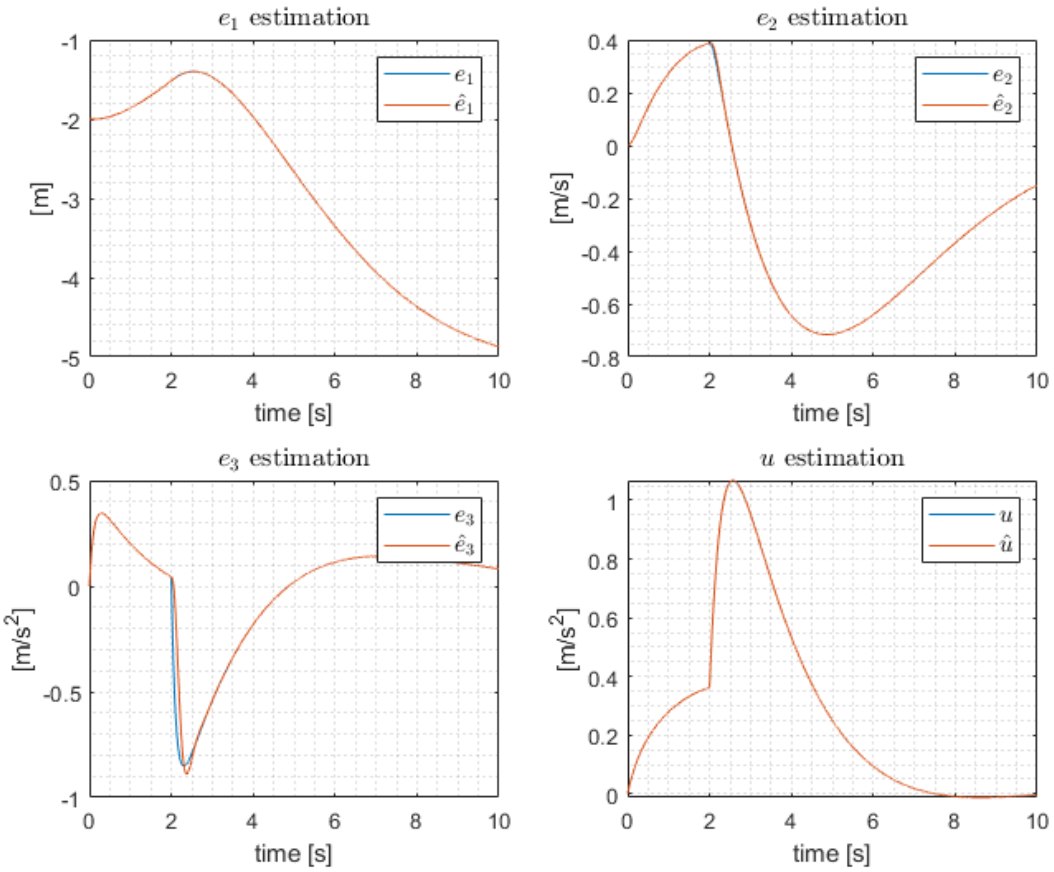



**Figure 4-4:** Estimation of the state vector using a Kalman filter while an unknown input is present, without measurement noise. The process noise covariance matrix $Q_{\Delta u_t} = 100$.

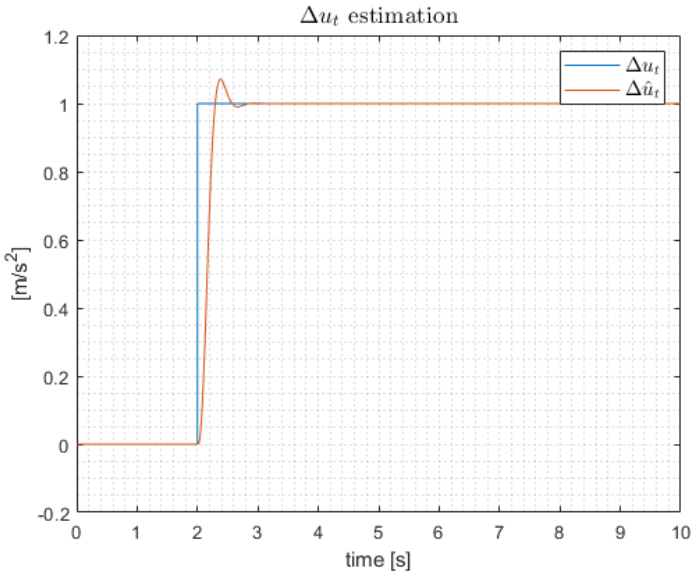



**Figure 4-5:** Estimation of the unknown input using a Kalman filter without measurement noise. The process noise covariance matrix $Q_{\Delta u_t} = 100$.

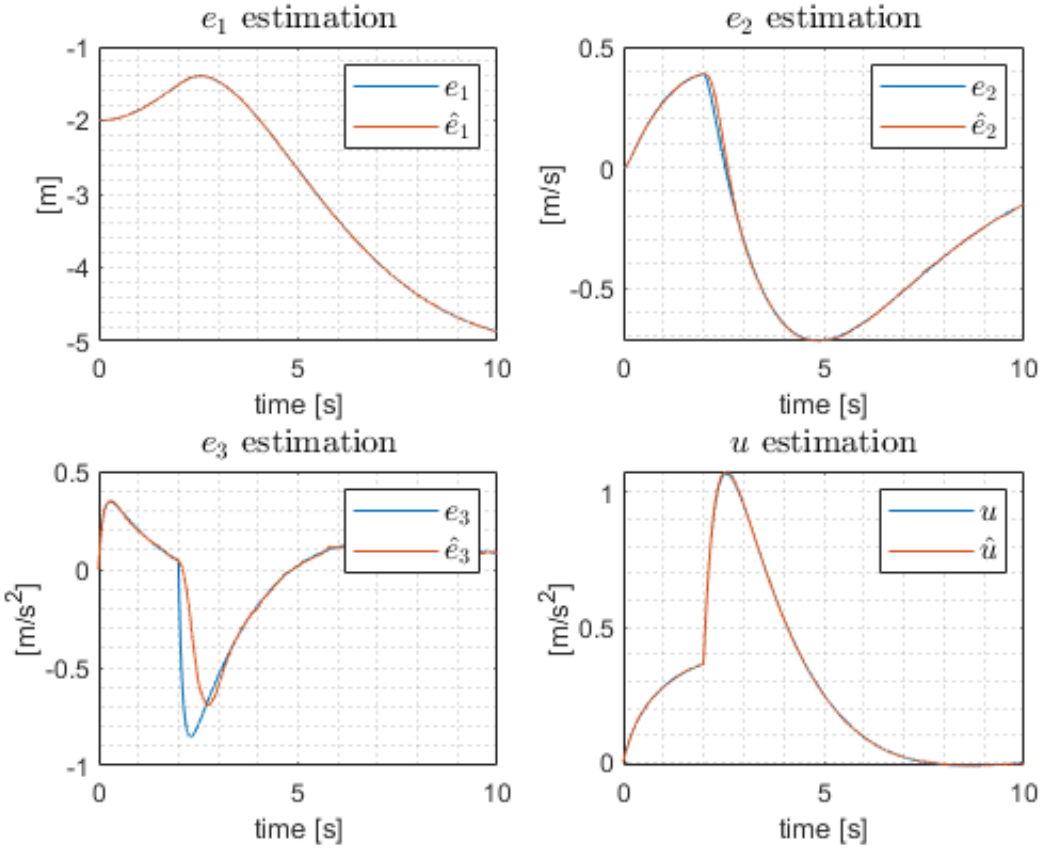



**Figure 4-6:** Estimation of the state vector using a Kalman filter while an unknown input is present, with measurement noise. The process noise covariance matrix $Q_{\Delta u_t} = 1$.

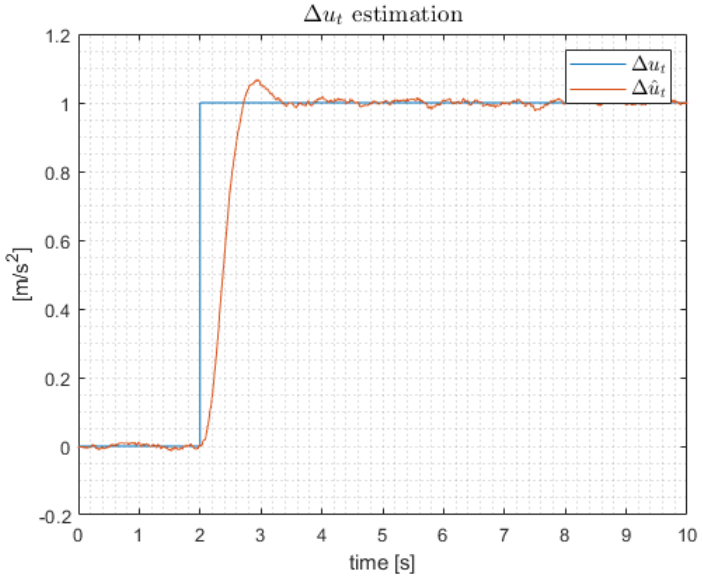



**Figure 4-7:** Estimation of the unknown input using a Kalman filter with measurement noise. The process noise covariance matrix $Q_{\Delta u_t} = 1$.

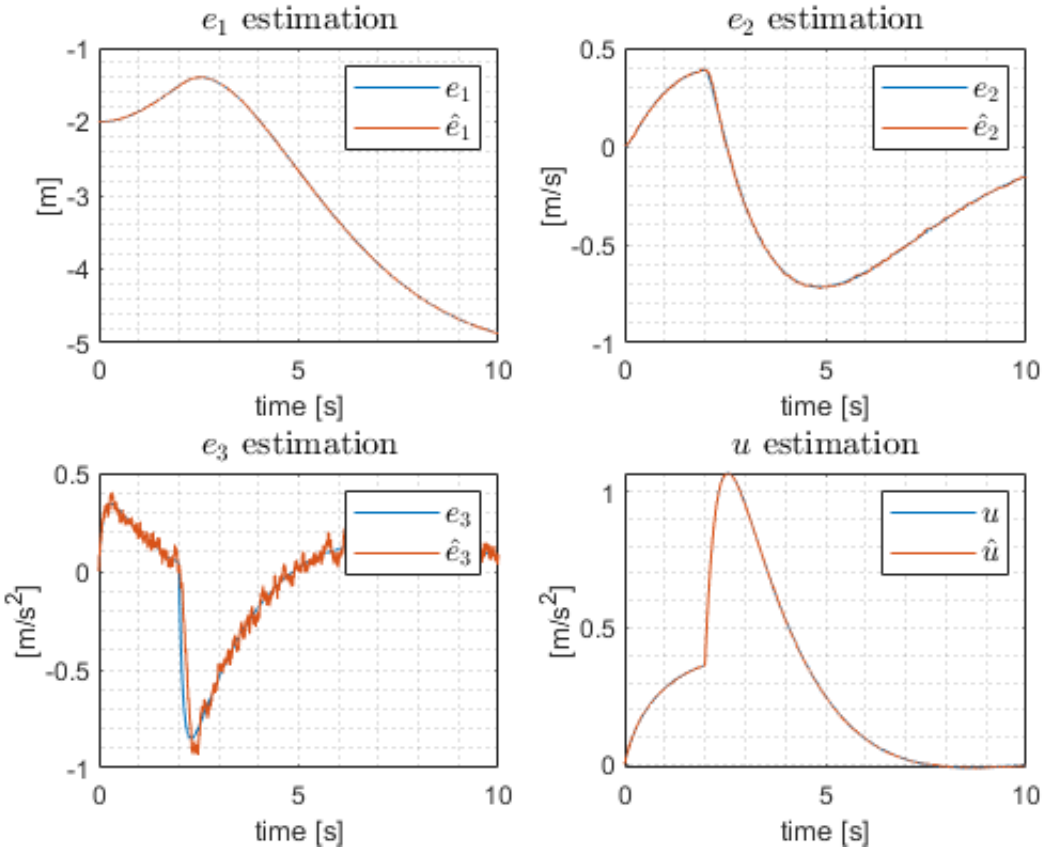



**Figure 4-8:** Estimation of the state vector using a Kalman filter while an unknown input is present, with measurement noise. The process noise covariance matrix $Q_{\Delta u_t} = 100$.

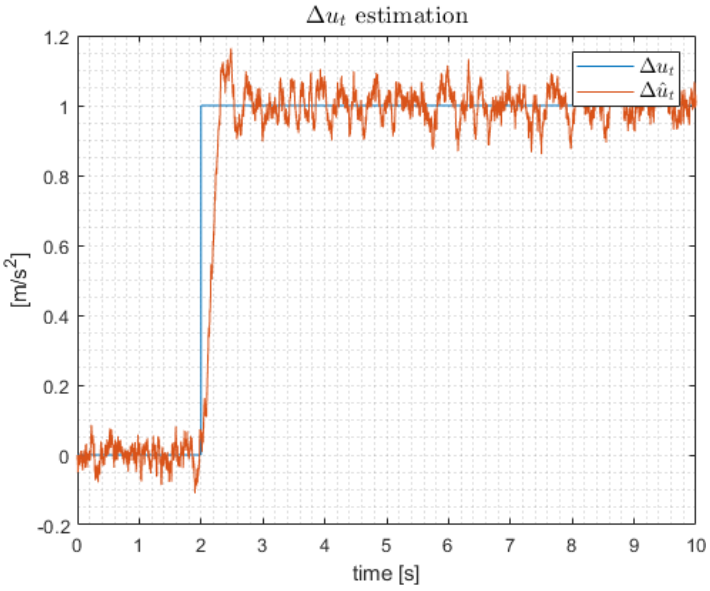



**Figure 4-9:** Estimation of the unknown input using a Kalman filter with measurement noise. The process noise covariance matrix $Q_{\Delta u_t} = 100$.

# Chapter 5

# Sliding Mode Observer for Anomaly Detection

The previous chapter illustrates how a Kalman filter can be used to estimate an unknown input of the system. However, as expressed in multiple papers [8, 86, 101], the simple linear vehicle model should eventually be replaced with a more suitable non-linear model. To this end, it was chosen to design a Sliding Mode Observer (SMO), which is not limited to a linear model to reconstruct the anomalous data of the system.

The design of the SMO is taken from the two studies by Floquet et al. [34] and Yu and Xu [124]. Thus, the novel contributions in this chapter are the combination of these two papers, and the application of the designed SMO to the Virtual Platooning (VP) protocol. The designed SMO is applied to both a linear as well as a non-linear vehicle model.

In this chapter, the design of the SMO is discussed. The chapter starts by shortly discussing the general working principle of an SMO in Section 5-1. Section 5-2 explains how an SMO can be designed such that it can be used to reconstruct unknown inputs in a linear system. Finally, Section 5-3 expends the SMO designed for a linear system to a non-linear vehicle model.

**Remark 2**
*Throughout the thesis, the SMO that uses a linear vehicle model in the observer design is referred to as the linear SMO, whereas the SMO with a non-linear vehicle model is called the non-linear SMO. These references are introduces for brevity, however, it must be kept in mind that both SMOs are non-linear.*

## 5-1 Working Principle Sliding Mode Observers

An SMO is a high gain state observer that originates from Sliding Mode Control (SMC). To better understand the working principle of an SMO, this section starts with a brief summary of SMC.

**Working Principle Sliding Mode Control** SMC is a non-linear control method, which forces the dynamics of the system to slide along a predefined surface [102, 124]. This predefined surface represents a desired behaviour of the system, for example, the desired equilibrium of the system. The sliding surface is reached by implementing a non-linear discontinuous control law.

Consider the following non-linear system

$$\begin{cases} \dot{x} &= f(x,t) + u(t), \\ y &= Cx(t), \end{cases} \tag{5-1}$$

where $x \in R^n$ is a state vector, $u \in R^m$ the control vector and $y \in R^p$ the system outputs. To push this system towards the origin $x = 0$, the sliding surface is chosen to be $x_1 + x_2 + ... + x_n = 0$, where $s(x) = x_1 + x_2 + ... + x_n$ is referred to as the switching function. The discontinuous control law switches from one continuous function to another based on the state of the switching function

$$u_i = \begin{cases} u_i^+(x,t), & \text{if } s_i(x) > 0, \\ u_i^-(x,t), & \text{if } s_i(x) < 0, \end{cases} \tag{5-2}$$

where $i$ stands for the $i$-th output $y$. The control input $u_i^+(x,t)$ ensures that the change in the switching function is negative ($\dot{s}_i(x) < 0$), which causes the value of $s_i(x)$ to decrease towards zero. The control input $u_i^-(x,t)$ represents a positive change in the switching function ($\dot{s}_i(x) > 0$).

**Advantages and Disadvantages Sliding Mode Control** Sliding Mode control has as the main advantages that it is robust against model uncertainties, can reach the sliding surface in finite time due to the discontinuous non-linear control law, and finally, the SMC is applicable to non-linear systems. However, the discontinuous control law introduces a chattering effect once the sliding surface is reached. Both functions that are part of the control law (5-2) push the switching function towards the other side of the sliding surface, which causes the chattering effect (see Figure 5-1). Systems where the control is executed by a mechanical component will experience faster wear due to the high frequency switching function. This initially resulted in a decrease in popularity of SMC. Recently, many techniques have been introduced in the literature to circumvent this chattering effect [11, 12, 31, 67].

**Working Principle Sliding Mode Observer** Observers designed for anomaly detection are not included in the control loop, thus, the chattering effect will not pose a limitation on the physical system. Therefore, using the SMO for fault detection is an interesting application.

Consider again the system defined in Equation (5-1). When designing an observer for this system, the desired dynamical behaviour is for the difference between the measured and estimated outputs —the residual— to go to zero. Thus the switching function is defined as

$$s(\hat{x}, y, t) = C\hat{x}(t) - y(t), \tag{5-3}$$

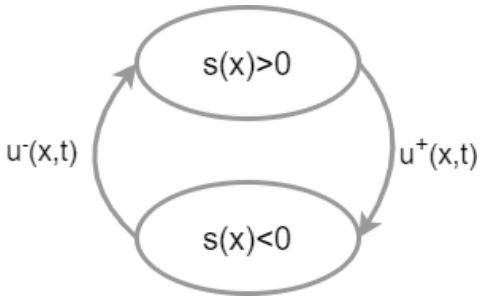



**Figure 5-1:** Switching function SMC.

and the sliding surface as

$$s(\hat{x}, y, t) = 0. \tag{5-4}$$

Including the state estimation with a simple feedback law that pushes the output estimation error (5-3) to zero is given by

$$\dot{\hat{x}} = f(\hat{x}, t) + u(t) + Lv(\hat{x}, y, t), \tag{5-5}$$

where $L$ is the feedback matrix and $v \in R^p$ is the discontinuous output injection signal equal to

$$v(\hat{x}, y, t) = -\rho \operatorname{sign}\left(s\left(x, y, t\right)\right), \tag{5-6}$$

where $\rho > 0$ is a design parameter. When the switching function has reached the sliding surface, the residual slides amongst the sliding surface defined in Equation (5-3).

**Definition 5.1**
After the switching function reaches the sliding surface, the injection signal that is necessary thereafter to maintain the sliding motion is called the equivalent injection signal [29]. □

## 5-2 Sliding Mode Observer Applied to Linear Systems

Edwards and Spurgeon [29] provide a method to use an SMO for the reconstruction of the unknown input. The authors show that in the case that an SMO meets two important requirements, it is possible to reconstruct the unknown input from the equivalent injection signal, defined in Definition 5.1. Unfortunately, for the system described in (4-4), one of the conditions —called the ranking condition— does not hold. Therefore, a technique introduced by Floquet et al. [34] is implemented to circumvent this condition.

Since the design of the SMO is quite complex, this section is divided into multiple parts. First, the concept Unknown Input Observer (UIO) is explained, and how the SMO should be designed to behave like a UIO. Next, it is shown how the unknown input can be retrieved from the equivalent input injection. Thereafter, the technique on how to circumvent the ranking condition is elaborated. And finally, the process on how to design the design matrices of the SMO is shown.

### 5-2-1 Unknown Input Sliding Mode Observer

A specific class of observers that is resilient against unknown inputs, is called a UIO. In order for an observer to behave like a UIO, two conditions must be met, which will be defined later on. Edwards and Spurgeon [29] translate these conditions into the design of the SMO, such that the SMO behaves like a UIO. Before the SMO is explained, the definition for a UIO is given below.

**Unknown Input Observer** UIOs are generally used in a specific fault detection set-up. In such a set-up, the unknown inputs are expected and should not lead to false positives. For example, a fault observer is designed for a wind turbine. However, there is no model for the effects of the wind on the blades, thus, the wind is an unknown input to the system. Using a UIO, the state estimates can be decoupled from the unknown input, such that the residual can be further analyzed for the presence of faults.

**Definition 5.2 Unknown Input Observer [19]**
An observer is defined as an unknown input observer if the state estimation error of the system approaches zero asymptotically regardless of the presence of the unknown input. □

**Unknown Input Sliding Mode Observer** To design an SMO for the system described in Equations (4-4) and (4-6) that behaves like a UIO, the necessary and sufficient conditions, given in [29], are

- rank(CD) = rank(D);
- Invariant zeros of (A,D,C) lie in the left half plane.

These conditions are the same as for a regular UIO, where the ranking condition is often referred to as the matching condition. Intuitively, the matching condition implies that the effects of the unknown input are directly visible in the system's output. Thus, making it possible to directly adjust the observer output independent of the unknown input. The second condition is necessary to design a stable SMO, since the invariant zeros of a system cannot be canceled out with a control law.

When these conditions hold, it is possible to design an SMO that stays on the sliding surface, independent of the unknown input [29]. Once the switching function reaches the sliding surface, the equivalent injection signal can be used to reconstruct the unknown input. The SMO, proposed by Floquet et al. [34], is given by

$$\dot{\hat{z}} = \bar{A}\hat{z} + \bar{B}u + G_l(y - \bar{C}\hat{z}) + G_n v_c(y - \bar{C}\hat{z}), \tag{5-7}$$

where $G_l$ and $G_n$ are design matrices, $z$ is a state transformation of $x$, and $v_c$ is the injection signal that pushes the observer dynamics towards the sliding surface. Transforming the state $x$ by $z = T_0 x$ will aid the stability proof of the SMO. The precise formulation of the matrices $G_l$ and $G_n$ will be given in Section 5-2-3. The sliding surface of the SMO (5-7) is defined as the output estimation error

$$s(x, y, t) = y(t) - \bar{C}\hat{z}(t) = 0. \tag{5-8}$$

In order to push the observer dynamics towards the sliding surface, a discontinuous injection signal is implemented. However, instead of using the injection signal suggested in Ref. [34], the injection signal from Ref. [124] is used in this thesis. The reason for using a different injection signal is that this injection signal guarantees quadratic stability of the sliding mode dynamics, as is shown in Appendix B. The injection signal is given by

$$v_c = \begin{cases} -\rho(t)||P_0\mathcal{D}_2||\frac{\epsilon_y}{||\epsilon_y||}, & \text{if } \epsilon_y \neq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{5-9}$$

where $\epsilon_y = y - \hat{y}$, and $\rho \geq \gamma + \mu$, with $\mu > 0$. Again, the matrices $P_0$ and $\mathcal{D}_2$ will be defined in Section 5-2-3. The variable $\gamma(t)$ represents the upper limit of the unknown input $\Delta u_t$

$$||\Delta u_t(t)|| \leq \gamma(t). \tag{5-10}$$

**Remark 3**
*The function $\gamma(t)$ represents the upper bound of the unknown input reconstruction. A logical choice for $\gamma$ would be to incorporate the physical limits on $u_t$. Anything that is below this limit cannot be detected with a simple data plausibility check, but can still cause dangerous situations on the intersection. The function $\gamma(t)$ can either be time varying or constant.*

**Reconstructing the unknown input** Once the sliding surface is reached, the unknown input can be reconstructed from the equivalent input injection. The dynamics of the sliding surface, using the SMO from Equation (5-7), are equal to

$$\dot{s} = \bar{C}\dot{\bar{\epsilon}} = \bar{C}(\bar{A} - G_l\bar{C})\bar{\epsilon} + \bar{C}\bar{D}\Delta u_t - \bar{C}G_n v(\bar{C}\bar{\epsilon}), \tag{5-11}$$

where $\bar{\epsilon}$ is the state estimation error $\epsilon = T_0^{-1}x - \hat{z}$. Designing $G_n$ and $G_l$ according to the procedure from Section 5-2-3, $\bar{\epsilon} = 0$ is a stable equilibrium of the state estimation error dynamics $\dot{\bar{\epsilon}}$. Thus the sliding surface dynamics (5-11) are simplified into

$$\dot{s} = \bar{C}\bar{D}\Delta u_t - \bar{C}_a G_n v_{eq}(\bar{C}\bar{\epsilon}) = 0, \tag{5-12}$$

where $v_{eq}(\cdot)$ indicates the equivalent output injection needed to maintain the sliding motion on $s = 0$. Thus, for the equality (5-12) to hold, it is necessary that the equivalent input injection compensates for the unknown input

$$\bar{C}G_n v_{eq}(\bar{C}\bar{\epsilon}) = \bar{C}\bar{D}\Delta u_t. \tag{5-13}$$

Multiplying both sides of Equation 5-13 with the pseudo inverse of $\bar{C}_a\bar{D}$ leads to an estimate of the unknown input $\Delta u_t$

$$\Delta\hat{u}_t = (\bar{C}\bar{D})^{+}\bar{C}G_n v_{eq}. \tag{5-14}$$

From Equation (5-13) it becomes clear why the ranking condition is so important. In the case that the ranking condition does not hold, the maximum amount of unknown inputs that can be reconstructed from Equation (5-14) is equal to rank$(CD)$. From equation (4-4) it follows that

$$(\bar{C}\bar{D})^{+} = \begin{bmatrix} 0 & 0 \end{bmatrix}. \tag{5-15}$$

Thus, the unknown input cannot be reconstructed from Equation (5-14). The next section implements a technique from Ref. [34] to still meet the requirement.

### 5-2-2 Fitting the Ranking Condition

A simple check shows that the Virtual Platooning dynamics (4-4) do not fulfill the ranking condition

$$\begin{aligned} \text{rank}(D) &= \text{rank}\left(\begin{bmatrix} 0 \\ 0 \\ -\frac{1}{\tau} \\ 0 \end{bmatrix}\right) = 1, \\ \text{rank}(CD) &= \text{rank}\left(\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{\tau} \\ 0 \end{bmatrix}\right) = 0 \neq \text{rank}(D). \end{aligned} \tag{5-16}$$

In order to fulfill the ranking condition, Floquet et al. [34] suggest to estimate extra auxiliary outputs until the ranking condition is fulfilled. The authors include a method that helps determine how many auxiliary outputs are needed, and from which available system outputs they need to be obtained. From the following inequality the variable $u_j$ is determined

$$\begin{aligned} C_j A^k D = 0, \quad \forall k < u_j - 1, \\ C_j A^{u_j - 1} D \neq 0. \end{aligned}$$

The notation $C_j$ denotes the $j$th row of the system output matrix $C$. The expression above indicates the amount of times $y_j$ needs to be differentiated in order for the unknown input to explicitly appear [34]. From the variables $u_j$, a new output matrix $C_a$ is formed that does fulfill the ranking condition. Selecting $u_1 = 2$ and $u_2 = 1$, the following $C_a$ matrix is retrieved

$$C_a = \begin{pmatrix} C_1 \\ C_1 A^{u_1 - 1} \\ C_2 \end{pmatrix} = \begin{pmatrix} C_1 \\ C_1 A \\ C_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{5-17}$$

which is not unexpected, since the dynamics of the vehicle are a chain of differentiators. Next, the new system output $y_a$ containing the extra auxiliary output is defined as

$$y_a = \begin{pmatrix} y_1 \\ v(y_1 - y_1^1) \\ y_2 \end{pmatrix}, \tag{5-18}$$

where the $y_1^1$ dynamics are

$$\dot{y}_1^1 = v(y_1 - y_1^1) + C_1 B u. \tag{5-19}$$

The signal $v(y_1 - y_1^1)$ is the injection input used in a step-by-step observer [35, 66]. The step-by-step observer is given as

$$\begin{cases} v(y_2 - y_2^1) & = \phi(y_1 - y_1^1) + \lambda |y_1 - y_1^1|^{\frac{1}{2}}, \\ \dot{\phi}(y_1 - y_1^1) & = \alpha \operatorname{sign}(y_1 - y_1^1), \end{cases} \tag{5-20}$$

where $\lambda, \alpha > 0$ are design constants. After a finite time, the auxiliary output will tend towards $C_1 A$, thus $y_a = C_a x$. Using the new $C_a$ matrix and state outputs $y_a$, the ranking condition is satisfied

$$\operatorname{rank}(C_a D) = 1 = \operatorname{rank}(D).$$

Replacing $C$ and $y$ from Equation (5-7) with $C_a$ and $y_a$, respectively, the new SMO is as follows

$$\dot{\hat{z}} = \bar{A}\hat{z} + \bar{B}u + G_l(y_a - \bar{C}_a\hat{z}) + G_n v_c (y_a - \bar{C}_a \hat{z}). \tag{5-21}$$

After a finite time, the auxiliary output vector $y_a$ will equal $y_a = \bar{C}_a \hat{z} = \bar{\epsilon}$. Thus, Equation (5-14) can be used to reconstruct the unknown input $\Delta u_t$.

### 5-2-3 Design of the Sliding Mode Observer

Using the new system containing the auxiliary output, it is possible to design an SMO conforming to the UIO requirements.

**State Transformation** The first step in designing an SMO is to apply a change of coordinates to the model of Equation (4-4), which helps the design process and stability proof of the observer. When the ranking condition is met, and there are no invariant zeros in the right-half plane, it is possible to apply the following change of coordinates to the Virtual Platooning model

$$\bar{A} = \left[\begin{array}{c|c} \bar{A}_{11} & \bar{A}_{12} \\ \hline \bar{A}_{211} \\ \bar{A}_{212} \end{array} \middle| \begin{array}{c} \\ \bar{A}_{22} \end{array}\right], \qquad \bar{D} = \begin{bmatrix} 0 \\ \bar{D}_2 \end{bmatrix}, \qquad \bar{C}_a = \begin{bmatrix} 0 & T \end{bmatrix}, \tag{5-22}$$

where $\bar{A}_{11} \in R^{(n-p)\times(n-p)}$ and $\bar{A}_{211} \in R^{(p-q)\times(p-q)}$ are a detectable pair, $\bar{D}_2 \in R^{q\times q}$ is nonsingular and $T \in R^{p\times p}$ is orthogonal. Furthermore, the invariant zeros of $(A, D, C_a)$ are the unobservable modes of $\bar{A}_{11}$ and $\bar{A}_{211}$. The control input matrix $\bar{B}$ does not require a special structure, thus is just the result of the matrix transformation to bring $(A, D, C_a)$ in the required structure.

The system as described in Equation (4-4) has no invariant zeros. However, in order to apply the design procedure to obtain the matrices $G_n$, $G_l$ and $P_0$, it is necessary that $\bar{A_{11}}$ is negative definite. Thus the matrix transformation, while obeying both the requirements for ref. [34] and [124] leads to

$$\bar{A} = \left[\begin{array}{c|ccc} -0.6667 & -2.6667 & -1.333 & 0 \\ \hline -0.6667 & -2.6667 & -2.333 & 0 \\ 0 & 0 & 0 & 1 \\ -2.0 & 2.0 & -7.0 & -10.0 \end{array}\right], \qquad \bar{D} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -10.0 \end{bmatrix},$$

$$\bar{C}_a = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \qquad \bar{B} = \begin{bmatrix} 3.333 \\ 3.333 \\ 0 \\ 0 \end{bmatrix}, \tag{5-23}$$

where the same values for the VP parameters are used as defined in Section 4-3-1. The state transformation matrix $T_0$ is equal to

$$T_0 = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}. \tag{5-24}$$

**Construction of the Design Matrices** In order to reconstruct the unknown input $\Delta u_t$, it is important that $\bar{\epsilon} = 0$ is a stable equilibrium of (5-11) [34]. To achieve the equilibrium $\bar{\epsilon} = 0$, the matrices $G_n$, $G_l$ and $P_0$ need to be designed appropriately. Therefore, the design procedure from Ref. [124] is followed to obtain the needed matrices.

In Appendix B, the stability is proved by using the Lyapunov candidate function $\mathcal{V}(\bar{\epsilon}) = \bar{\epsilon}\bar{P}\bar{\epsilon}$, where the matrix $\bar{P}$ is defined as

$$\bar{P} = \begin{bmatrix} \bar{P}_1 & \bar{P}_1\bar{L} \\ \bar{L}^T\bar{P}_1 & \bar{P}_2 + \bar{L}^T\bar{P}_1\bar{L} \end{bmatrix} \geq 0, \tag{5-25}$$

with $\bar{P}_1 \in R^{(n-p)\times(n-p)}$ and $\bar{P}_2 \in R^{p\times p}$. The feedback matrix $G_n$ is constructed by using the matrices $L$ and $P_2$ from (5-25)

$$G_n = \begin{bmatrix} -\bar{L}T^T \\ T^T \end{bmatrix} P_0^{-1}. \tag{5-26}$$

The matrix $T$ is part of the $\bar{C}_a$ matrix after the transformation (5-22), and $P_0$ is defined as

$$P_0 := T\bar{P}_2T^T. \tag{5-27}$$

Next, the matrices $G_l$ and $\bar{P}$ are obtained by solving the following inequality

$$\bar{A}^T\bar{P} + \bar{P}\bar{A} - \bar{P}G_l\bar{C}_a - \bar{C}_a^TG_l^T\bar{P} + \bar{P}W\bar{P} + \bar{P}G_l\bar{C}_aG_l^T\bar{P} < 0, \tag{5-28}$$

while minimizing trace$(P^{-1})$. The matrix $W$ is a symmetric positive definite design matrix. When this inequality holds, the Lyapunov candidate guarantees that the SMO is quadratically stable [124]. Substituting $G_l = P^{-1}C^TV^{-1}$ leads to

$$\bar{P}\bar{A} + \bar{A}^T\bar{P} - \bar{C}_a^TV^{-1}\bar{C}_a + \bar{P}W\bar{P} < 0, \tag{5-29}$$

where $V$ is again a symmetric positive definite design matrix. Using the Schur complement on (5-29), the inequality (5-29) can be solved while minimizing trace$(P^{-1})$ with the following LMIs

$$\begin{aligned} \underset{P_{11},P_{121},P_{22},\bar{X}}{\text{minimize}} \quad & \text{trace}(\bar{X}) \\ \text{subject to} \quad & \begin{bmatrix} \bar{P}\bar{A} + \bar{A}^T\bar{P} - \bar{C}_a^TV^{-1}\bar{C}_a & \bar{P} \\ \bar{P} & -W^{-1} \end{bmatrix} < 0, \\ & \begin{bmatrix} -\bar{P} & I \\ I & -\bar{X} \end{bmatrix} < 0, \\ & \bar{P} = \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix} > 0, \end{aligned} \tag{5-30}$$

where $P_{12}$ has the structure $P_{12} = \begin{bmatrix} P_{121} & 0 \end{bmatrix}$. The authors from Ref. [124] state that minimizing trace$(\bar{X})$ is equal to minimizing trace$(P^{-1})$ due to the condition $\left[\begin{smallmatrix} -\bar{P} & I \\ I & -\bar{X} \end{smallmatrix}\right] < 0$. Applying this design procedure to the model (5-23) leads to the following $G_n$ and $G_l$

$$G_l = \begin{bmatrix} 0 & 0 & 0 \\ -53.194 & -385250 & 172910 \\ -44478 & -153.48 & 53.194 \\ -153.48 & -1.2e+06 & 385250 \end{bmatrix}, \qquad G_n = \begin{bmatrix} 0.74471 & 5393.5 & -2420.7 \\ -0.53194 & -385.25 & 172.91 \\ -44.478 & -0.15348 & 0.053194 \\ -0.15348 & -1200 & 385.25 \end{bmatrix}. \tag{5-31}$$

## 5-3 Sliding Mode Observer Applied to Non-Linear Systems

In this section a non-linear extension of the SMO designed for linear systems is discussed.

### 5-3-1 Non-Linear Vehicle Model

The simple linear vehicle model from Equation (3-3) is replaced with the non-linear model suggested in Ref. [101]. The authors take the following model for the motor dynamics

$$\dot{F} = -\frac{F}{\tau} + \frac{u}{\tau}, \tag{5-32}$$

where $\tau$ is the lag related to the engine dynamics, $F$ is the driving force produced by the engine and $u$ is the control input. Next, the longitudinal forces on the vehicle are given by

$$ma = F - K_d v^2 - d_m, \tag{5-33}$$

where $m$ is the mass of the vehicle, $K_d$ is the aerodynamic drag coefficient and $d_m$ is the mechanical drag on the vehicle. It is notable that the mechanical drag is modeled as a constant, which is a simplification of the true mechanical drag. Differentiating Equation (5-33) over time, and substituting $\dot{F}$ from Equation (5-32) and $F$ from Equation (5-33), the following non-linear vehicle model is obtained

$$\dot{a} = -\frac{1}{\tau}a + \frac{1}{m\tau}u - \frac{2K_d}{m}va - \frac{K_d}{m\tau}v^2 - \frac{d_m}{m\tau}. \tag{5-34}$$

### 5-3-2 Sliding Mode Observer for Non-Linear Vehicle Model

As shown below, the dynamics from (5-34) can be split into a linear and a non-linear part

$$\dot{a} = -\frac{1}{\tau}a + \frac{1}{m\tau}u + \underbrace{\left(-\frac{2K_d}{m}va - \frac{K_d}{m\tau}v^2 - \frac{d_m}{m\tau}\right)}_{g}, \tag{5-35}$$

where $g$ represents the non-linear dynamics, and the remaining part the linear dynamics. Using this new formulation of $\dot{a}$, the VP dynamics are extended accordingly

$$\begin{aligned}
\dot{e}_3 &= \dot{a}_t - \dot{a} - h\ddot{a} \\
&= -\frac{1}{\tau}a_t + \frac{1}{m\tau}u_t + g_t(v_t, a_t) + \frac{1}{\tau}a - \frac{1}{m\tau}u - g(v,a) - h\left(-\frac{1}{\tau}\dot{a} + \frac{1}{m\tau}\dot{u} + \dot{g}(v,a)\right) \\
&= -\frac{k_p}{m\tau}e_1 - \frac{k_d}{m\tau}e_2 - \frac{1}{\tau}e_3 + \underbrace{g_t(v_t, a_t) - g(v,a) - \dot{g}(v,a)}_{g_{\text{tot}}} - \frac{1}{m\tau}\Delta u_t.
\end{aligned} \tag{5-36}$$

Using Equation (5-36), the complete state dynamics are given by

$$\begin{pmatrix} \dot{e}_1(t) \\ \dot{e}_2(t) \\ \dot{e}_3(t) \\ \dot{u}(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\frac{k_p}{m\tau} & -\frac{k_d}{m\tau} & -\frac{1}{\tau} & 0 \\ \frac{k_p}{h} & \frac{k_d}{h} & 0 & -\frac{1}{h} \end{pmatrix} \begin{pmatrix} e_1(t) \\ e_2(t) \\ e_3(t) \\ u(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ g_{\text{tot}}(v_t, a_t, v, a) \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -\frac{1}{m\tau} \\ 0 \end{pmatrix} \Delta u_t(t) + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{h} \end{pmatrix} u_{t,\text{rec}}, \tag{5-37}$$

which is represented in a compact form by

$$\dot{x}(t) = Ax(t) + Bu_{t,\text{rec}} + G(v_t, a_t, v, a) + D\Delta u_t. \tag{5-38}$$

The new non-linear SMO is simply the linear SMO, extended by the non-linear dynamics $\bar{G}(T_0^{-1}\hat{z})$

$$\dot{\hat{z}} = \bar{A}\hat{z} + \bar{B}u + G_l(y - \bar{C}_a\hat{z}) + G_n v_c(y - \bar{C}_a\hat{z}) + \bar{G}(T_0^{-1}\hat{z}), \tag{5-39}$$

where $\bar{G} = T_0 G$ is obtained from using the state transformation $T_0$ as defined in Equation (5-24).

**Estimating the Non-Linear Dynamics** In order to include the non-linear dynamics $G(v_t, a_t, v, a)$ in the SMO, the variables $a_t$ and $v_t$ need to be retrieved from the state estimates $\hat{e}$. One way to do this is by using the following equations

$$\hat{v}_t = \hat{e}_2 + v + ha, \tag{5-40}$$
$$\hat{a}_t = \hat{e}_3 + a + h\dot{a}. \tag{5-41}$$

As can be seen from the equations defined above, only locally available variables are required. Next, an estimate of $g$ is obtained

$$\hat{g}_{tot}(\hat{v}_t, \hat{a}_t, v, a) = \hat{g}_t(\hat{v}_t, \hat{a}_t) - g(v, a) - \dot{g}(v, a), \tag{5-42}$$

where

$$\dot{g}(v, a) = -\frac{2K_d}{m\tau}va - \frac{2K_d}{m}a^2 - \frac{2K_d}{m}\dot{a}v. \tag{5-43}$$

## 5-4 Discussion

In this chapter, the design for both the linear and the non-linear SMO are provided. One thing that immediately becomes clear, is that the design process is more complex than the Kalman filter from Chapter 4. However, after implementing the design process, the only two design matrices that need to be tuned are $V$ and $W$. The matrices $V$ and $W$ are symmetric positive definite matrices, with the dimensions of $V \in \mathbb{R}^{p \times p}$ corresponding to the output size $p$ and $W \in \mathbb{R}^{n \times n}$ to the state dimensions $n$. Furthermore, to construct a non-linear SMO, it is only necessary to extend the linear SMO with the non-linear dynamics, which is not possible with the Kalman filter.

Furthermore, it must be noted that even though the observers are specifically designed for the VP concept, the approach can be used for other Intersection Controllers (ICs) as well. The observer for the unknown input must be redesigned such that it estimates the variable crucial for the anti-collision constraint, shared via the Vehicular Ad-Hoc Network (VANET). For example, the MPC based IC from Ref. [63] relies on the safe distance $d_{i,j}$ between vehicles $i$ and $j$ on the intersection. The observer can be redesigned to obtain an estimation of $\Delta d_{i,j}$.

In this chapter, the design concepts where mainly taken from Ref. [34] and [124]. Thus, the contributions in this chapter were the combination of the SMO design from both researches, and the application of the SMO to the VP protocol. To combine the papers, the state transformation $z = T_0 x$ was adjusted to fit the design for both the papers [34, 124]. Furthermore, a different injection signal than suggested in [34] was implemented, since the new injection signal guarantees quadratic stability of the SMO.

Furthermore, the SMO was extended with a non-linear vehicle model to reconstruct the unknown input transmitted by the target vehicle, which has not been done before in the literature. To test the performance of the SMO, the next chapter provides simulations of both the linear as well as the non-linear SMO.

# Chapter 6

# Simulation Results

In this chapter, the performance of the Sliding Mode Observer (SMO) is analyzed using simulation results. The simulation consists of two vehicles approaching an intersection towards a common collision point, while the transmitted data of the target vehicle is subject to data alteration. The goal is to reconstruct the altered data with an SMO, while only using locally available data and the data received from the target vehicle.

Multiple novel contributions are provided in this chapter. First of all, for the first time the SMO as designed in Chapter 5 is applied to an Intersection Control (IC) technique subject to data alteration in simulations. Secondly, the SMO is adjusted such that an adequate unknown input reconstruction is obtained while the used data is subject to measurement noise. And finally, the non-linear SMO is tested on a non-linear vehicle model, which has not been done before in the literature.

The chapter is divided into two parts. The first part provides the simulation results of the SMO while the dynamics of the vehicle are produced with the simple linear vehicle model from Equation (3-3). The second part shows the simulation results of the SMO while the dynamics of the vehicle are represented by a non-linear vehicle model, given in Equation (5-34).

**Remark 4**
*As a reminder, the following remark is repeated. The SMO that uses a linear vehicle model in the observer design is referred to as the linear SMO, whereas the SMO with a non-linear vehicle model is called the non-linear SMO. These references are introduces for brevity, however, it must be kept in mind that both SMOs are non-linear.*

## 6-1 Simulation Result Linear Sliding Mode Observer

The same simulation set-up and settings as described in Section 4-3-1 are used throughout this chapter. In Appendix A, the Simulink set-up of the SMO is included. Again, for clarity, the simple linear vehicle dynamics used in the simulations are repeated below

$$\dot{a}_i = -\frac{1}{\tau}a_i + \frac{1}{\tau}u_i, \tag{6-1}$$

where the subscript $i$ stand for the $i$-th vehicle.

First, the simulation result for the unknown input reconstruction is shown without measurement noise, whereafter the result with measurement noise is shown.

### 6-1-1 Simulations without measurement noise

The linear SMO designed in the previous chapter, containing the auxiliary output matrix $C_a$ and vector $y_a$ are given by $\dot{\hat{z}} = \bar{A}\hat{z} + \bar{B}u + G_l(y - \bar{C}_a\hat{z}) + G_n v_c(y_a - \bar{C}_a\hat{z})$, after which the unknown input $\Delta u_t$ can be reconstructed by using $\Delta\hat{u}_t = (\bar{C}_a\bar{D})^+\bar{C}_a G_n v_{eq}$.

To reconstruct the unknown input, the first step is to estimate the auxiliary output to be able to form $y_a$. In the absence of measurement noise, it is possible to directly apply the step-by-step observer given in Equation (5-20). The top panel of Figure 6-1 shows the estimation of the auxiliary output $e_3$ and the bottom panel shows the estimation error. The estimation error has a sudden spike at $t = 2s$, which is caused by the initialization of a nonzero $\Delta u_t$. Using $y_a$, the states are estimated, as is shown in Figure 6-2. The chattering effect, which is introduced by the discontinuous injection signal, is clearly visible in the sliding surface dynamics $\dot{s}$, as is shown in Figure 6-3. Finally, the estimation of the unknown input $\Delta u_t$ is reconstructed from the equivalent input injection. As can be seen in Figure 6-4, the unknown input $\Delta u_t$ is almost exactly reconstructed.

### 6-1-2 Simulations with measurement noise

Both the step-by-step observer and the SMO are high-gain observers, which results in fast observer dynamics. However, a big disadvantage with high-gain observers is that the measurement noise is amplified. Therefore, the step-by-step observer is replaced by a robust exact differentiator [69], in order to obtain an estimate of $e_3$ while measurement noise is present. Figures 6-5, 6-6 and 6-7 show $\hat{e}_3$, $\hat{x}$, $\dot{s}$ and $\Delta\hat{u}_t$, respectively. As can be seen in Figure 6-5, there is a large estimation error at $t = 0s$ and $t = 2s$. Both estimation errors are caused by the high rate of change of the variable $e_3$.

Even though the robust exact differentiator is able to estimate $e_3$, the estimate still contains noise, as can be seen in Figure 6-5. The SMO was not designed to handle measurement noise, thus the injection signal (5-9) tries to compensate for the estimation error caused by the measurement noise. This increases the noise in the $\Delta u_t$ estimation. In order to remove the high frequency noise, a first order low-pass filter was added to $\Delta u_t$. A low-pass filter removes the high frequency signal at the specified cut-off frequency. The transfer function from input to output is given by

$$\frac{Y(s)}{U(s)} = \frac{1}{\frac{1}{w_c}s + 1}, \tag{6-2}$$

where $w_c$ is the cut-off frequency. The lower the cut-off frequency, the higher the delay in the $\Delta u_t$ estimation. Figure 6-8 shows the $\Delta u_t$ estimation for different cut-off frequencies. A

clear trade-off is visible between noise reduction and estimation delay. For now, the cut-off frequency $w_c = 20$ is chosen, since the $\Delta u_t$ estimation is still fast while the noise is reduced.

From figure 6-8, two effects are immediately noticeable. Firstly, the $\Delta u_t$ estimation has a large peak at $t = 0\,s$, and secondly, there is a slight miss-match in the estimated and true $\Delta u_t$ from $t = 0.5\,s$ to $t = 2\,s$. Both effects are explained when looking at the $e_3$ estimation in Figure 6-5. A large miss-match is observable in the $e_3$ estimation at $t = 0\,s$, whereafter the miss-match becomes smaller, but is still non-zero. The same phenomenon is also visible in the sliding dynamics, as shown in Figure 6-7. The sliding dynamics show a large peak at $t = 0\,s$, after which a small miss-match is visible in the bottom panel until $t = 2\,s$.

### 6-1-3 Discussion Linear Sliding Mode Observer

In the absence of measurement noise, the SMO can almost exactly reconstruct the unknown input. However, while measurement noise is present, the step-by-step observer and the SMO cannot reconstruct the unknown input sufficiently. Therefore, we replaced the step-by-step observer with a robust exact differentiator. Additionally, the $\Delta u_t$ estimation is filtered with a low-pass filter to reduce the high frequency noise.

Instead of using a low-pass filter, multiple concepts are suggested in the literature in order to make the SMO more robust [18, 54, 121, 128]. For example, Xing-Gang Yan and Edwards [121] suggest to model the measurement uncertainties as an extra unknown input, and decouple the estimated measurement noise from the fault reconstruction. However, due to time limitations, in this thesis the low-pass filter was used.

In the simulations, the dynamics of the vehicles are now produced according to the simple linear vehicle model. Therefore, the linear SMO is able to almost perfectly reconstruct the unknown input. In a more realistic setting, the vehicles will not behave according to the simple linear vehicle model. Furthermore, in the simulations, the vehicles were assumed to be homogeneous, which is also not expected in a more realistic setting.

## 6-2 Simulation Result Non-Linear Sliding Mode Observer

The performance of the non-linear SMO was tested by comparing the $\Delta u_t$ estimation obtained from the SMO described in Equation (5-39) with the result obtained from a linear Kalman filter. Two homogeneous vehicles approach an intersection from distinct trajectories, with a common collision point.

**Non-Linear Vehicle Model** The dynamics of both the target vehicle and the host vehicle are generated by using the non-linear vehicle model described in Equation (5-34). For clarity, the non-linear model is provided again below

$$\dot{a} = -\frac{1}{\tau}a + \frac{1}{m\tau}u - \frac{2K_d}{m}va - \frac{K_d}{m\tau}v^2 - \frac{d_m}{m\tau} \tag{6-3}$$

The simulations are performed without the presence of measurement noise. The coefficients are chosen to equal $m = 1\,kg$, $K_d = 0.5$, $d_m = 0\,N$ and $\tau = 0.1\,s$. For simplicity, only the aerodynamic drag is considered, which leads to the mechanical drag $d_m = 0\,N$.

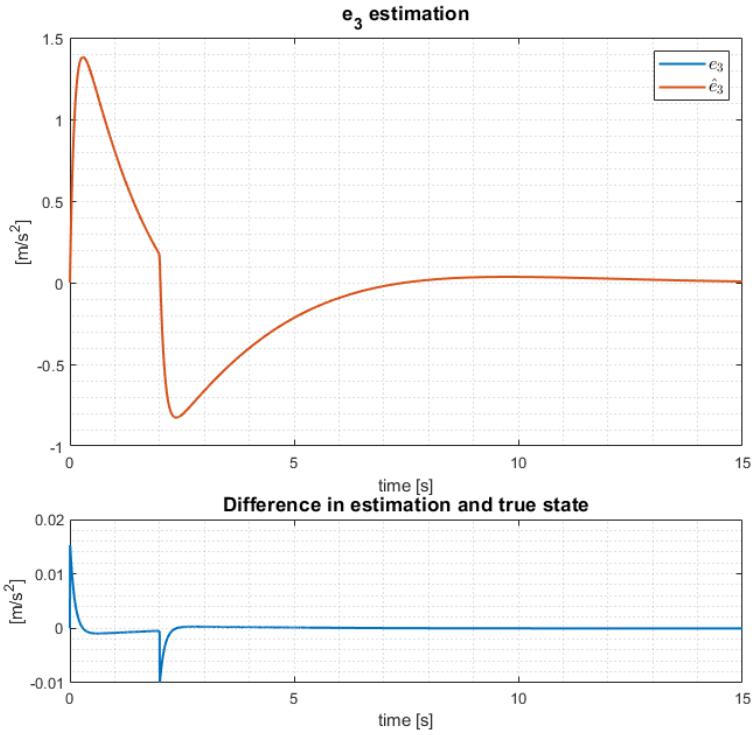



**Figure 6-1:** Auxiliary output $e_3$ estimation using the step by step observer. Top: real (blue) and estimated (red) $e_3$. Bottom: estimation error.

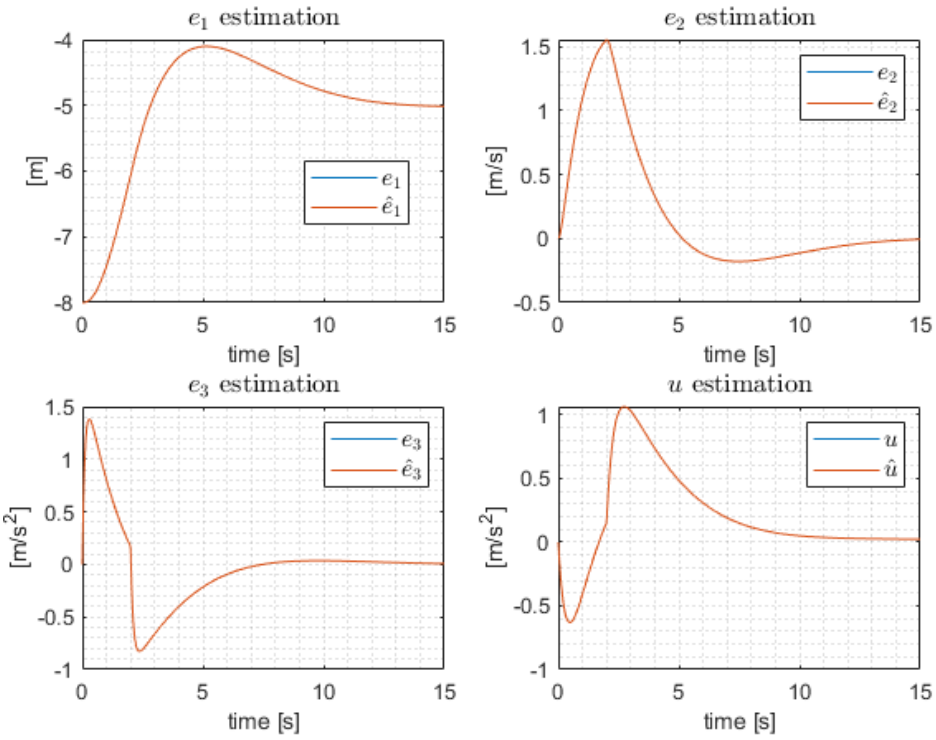



**Figure 6-2:** The true states (blue) of the Virtual Platooning (VP), and the estimated states (red) using the SMO.

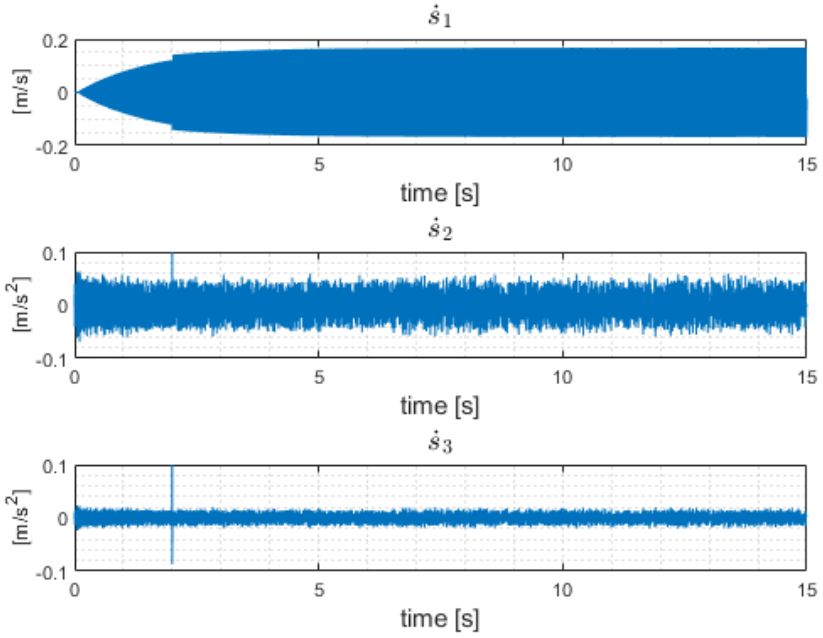



**Figure 6-3:** Sliding surface of $C_a\epsilon$. The sudden spike at $t = 2s$ is caused by the initialization of $\Delta u_t$.

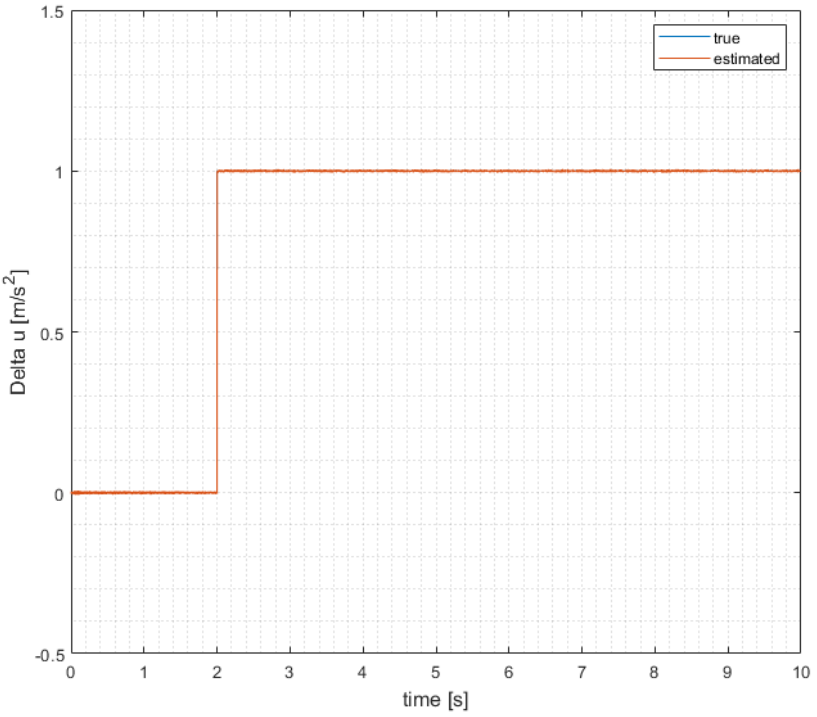



**Figure 6-4:** The true (blue) and estimated (red) $\Delta u_t$ using the SMO.

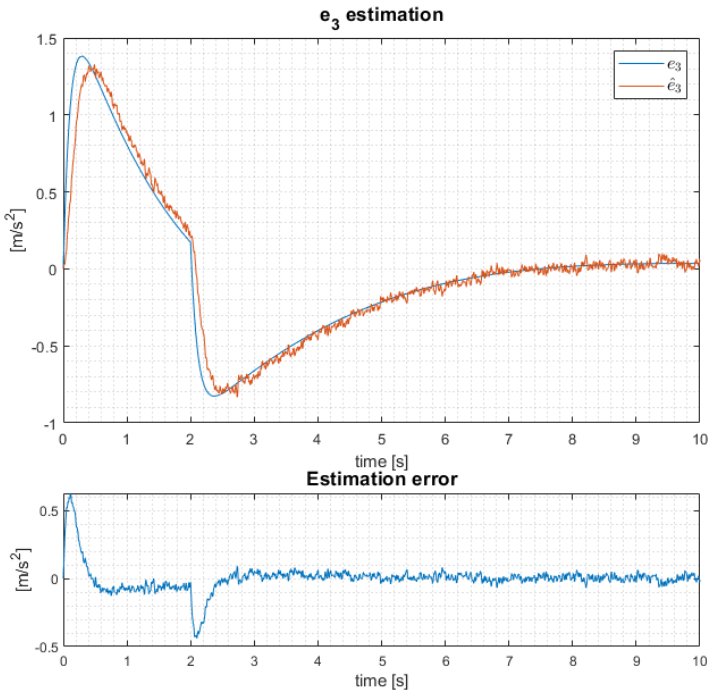



**Figure 6-5:** $e_3$ estimation using the robust differentiator. Top: real (blue) and estimated (red) $e_3$. Bottom: estimation error.

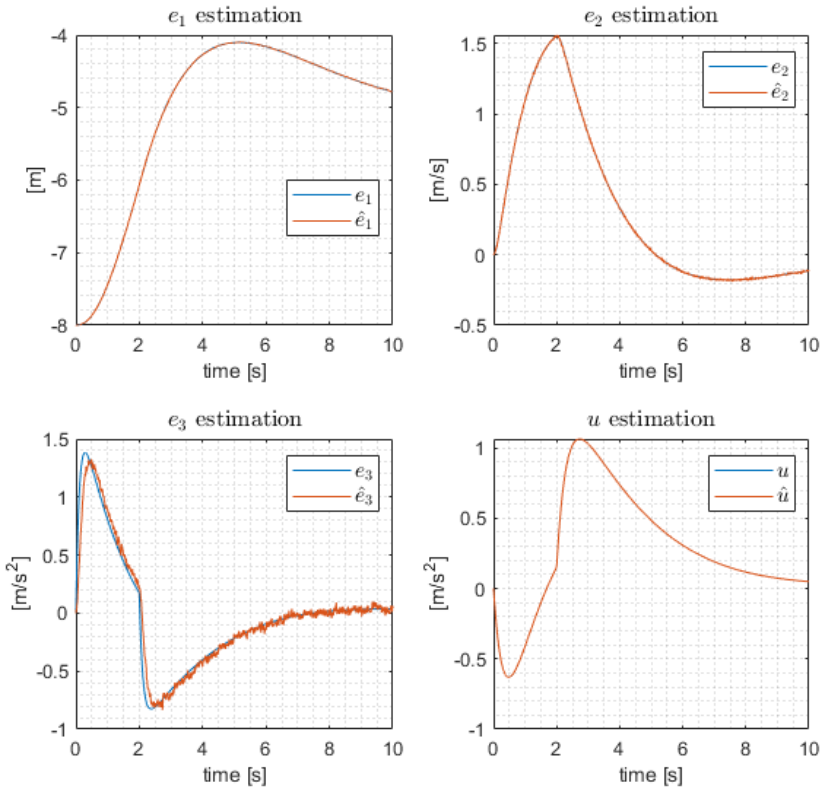



**Figure 6-6:** The true states (blue) of the VP, and the estimated states (red) using the SMO while measurement noise is present.

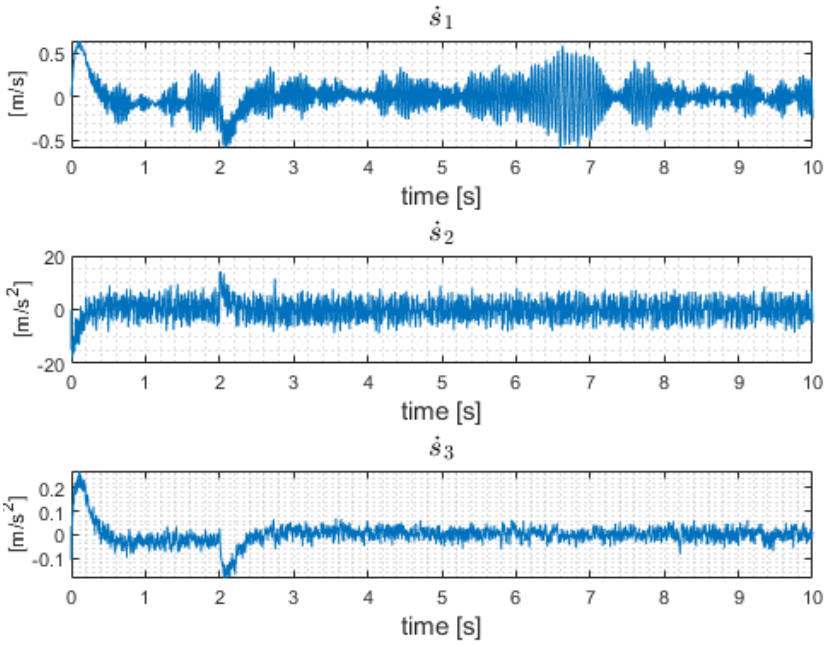



**Figure 6-7:** Sliding surface of $C_a\bar{\epsilon}$ while measurement noise is present. The sudden spike at t=2s is caused by the initialization of $\Delta u_t$. The spike is out of scope of the figures to clearly show the chattering effect of the switching function.

The mass of the vehicle is chosen to equal $1\,kg$, since a higher value would require the redesigning of the parameters $k_p$, $k_d$ and $k_{cc}$, which is not the aim of this simulation. Indeed, a higher value of $m$ would lead to a redesign of the parameters due to the fact that the control input is multiplied with $\frac{1}{m\tau}$, as can be seen in Equation (6-3).

The coefficient $K_d = 0.5$ is an average value for a standard passenger car calculated by $K_d = \frac{2F_d}{\rho A v^2}$, where $F_d$ is the drag force on the vehicle, $\rho$ the density of the air and $A$ the frontal area of the vehicle. The remaining parameters are as specified in Subsection 4-3-1.

**Kalman Filter for Anomaly Detection** First, the Kalman filter is used to estimate the unknown input from the data produced by the non-linear vehicle model. Since $m = 1\,kg$, the Kalman filter defined in Equation (4-8) does not need to be redesigned. However, the process noise covariance matrix is set to

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}, \tag{6-4}$$

since the Kalman filter expects the states $e$ to behave according to the linear model.

Figures 6-9 and 6-10 show the state and $\Delta u_t$ estimation, respectively. As can be seen, the $\Delta u_t$ estimation does not follow the true $\Delta u_t$.

**Sliding Mode Observer for the Non-Linear Vehicle Model** The non-linear SMO defined in Equation (5-39) is used to reconstruct the unknown input. Since the linear part of the

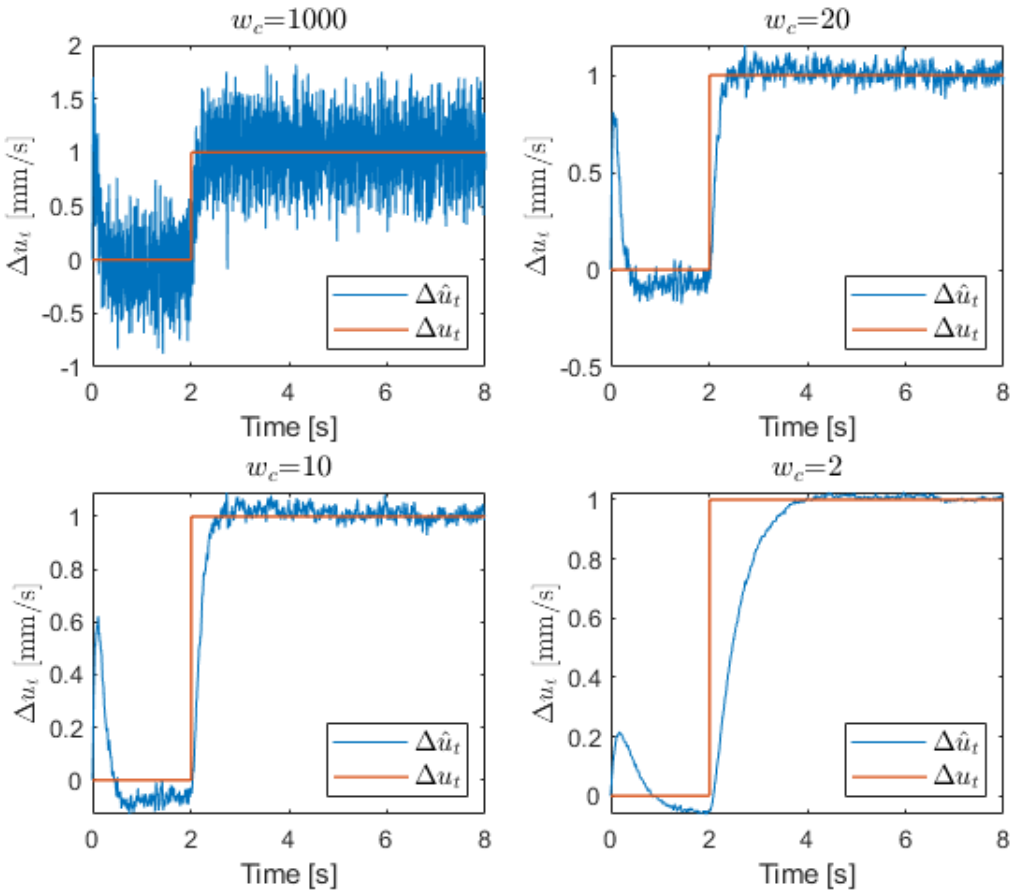



**Figure 6-8:** True (red) and estimated (blue) $\Delta u_t$ with different cut-off frequencies of the low-pass filter.

non-linear SMO has stayed the same, the design matrices $G_n$, $G_l$ and $P_0$ as defined in Section 5-2-3 can be used.

Figures 6-11 and 6-12 show the state and $\Delta u_t$ estimation, respectively, obtained from the non-linear SMO. The estimation of the states and $\Delta u_t$ are almost a perfect overlay of the true states and unknown input.

### 6-2-1 Discussion Non-Linear Sliding Mode Observer

Comparing the unknown input estimation obtained from the non-linear SMO with the Kalman filter results shows a large improvement. The linear Kalman filter is not able to correctly estimate the unknown input anymore when applied to the non-linear vehicle dynamics, while the non-linear SMO shows an almost perfect estimation as can be seen in Figure 6-12.

Even though the result is promising, an important question comes to mind. In a realistic scenario, the vehicles will not be homogeneous. With the linear vehicle model, only the parameter $\tau$ was unknown from the target vehicle, whereas with the non-linear model, the parameters $\tau$, $m$, $K_d$ and $d_m$ are unknown, which is crucial information for the non-linear SMO.

Thus, in the case of heterogeneous vehicles for both the linear as for the non-linear SMO, the parameters relevant in the vehicle model somehow need to be obtained. The European Telecommunications Standards Institute (ETSI) does provide communication protocols where this type of information is shared in the messages [52]. However, that also means that this information could be subject to data alteration. It is therefore important to further investigate more the more realistic scenario where the vehicles are not homogeneous.

## 6-3 Discussion

The contributions in this chapter were the application of the designed SMOs in simulations. The simulations show very promising results, where in both the linear and the non-linear case the unknown input is almost exactly reconstructed. However, in the presence of measurement noise, the performance quickly degrades. As a solution, the step-by-step observer was exchanged with a robust exact differentiator, and a first order low-pass filter was added to the $\Delta u_t$ estimation. The low-pass filter introduces a delay in the estimation of the unknown input. Using one of the methods suggested in the Literature (e.g., [18, 54, 121, 128]) to improve the robustness of the SMO might lead to a better trade-off between the performance of the unknown input estimation and the noise reduction. It is therefore recommended in future research to further investigate these methods.

The non-linear SMO shows very promising results in simulations. However, both the simulations for the linear and non-linear SMO do not consider network induced complications, such as communication delay, actuator delay, discretization of the control methods and package loss. Therefore, the next chapter shows the performance of the SMO in experiments to analyze the performance in a more realistic setting.

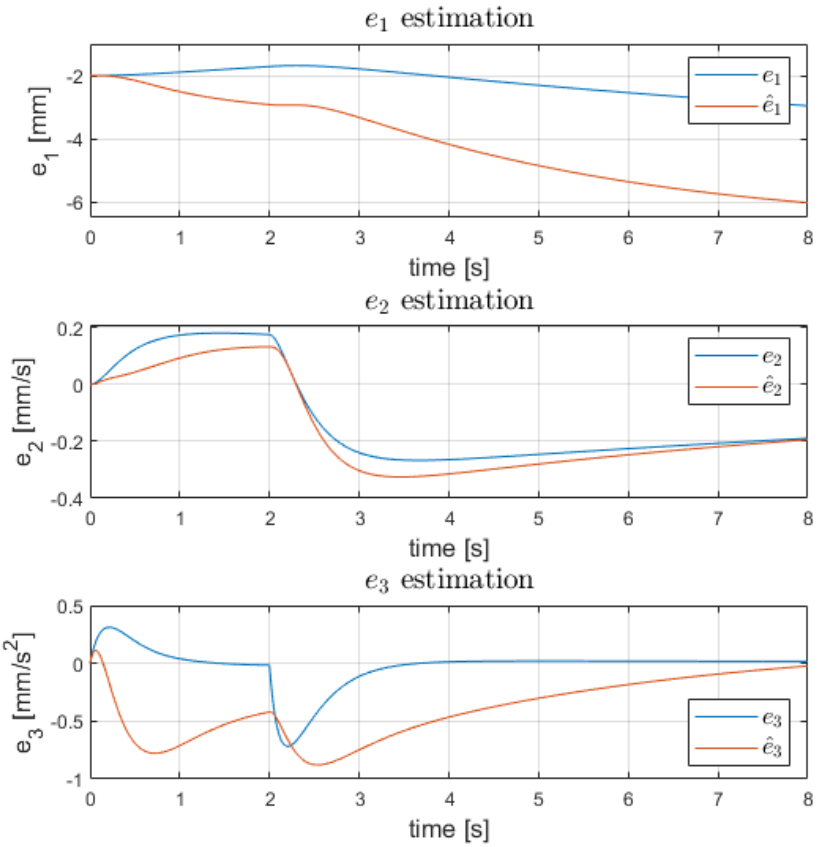



**Figure 6-9:** State estimation of data produced by a non-linear vehicle model using a linear Kalman filter.

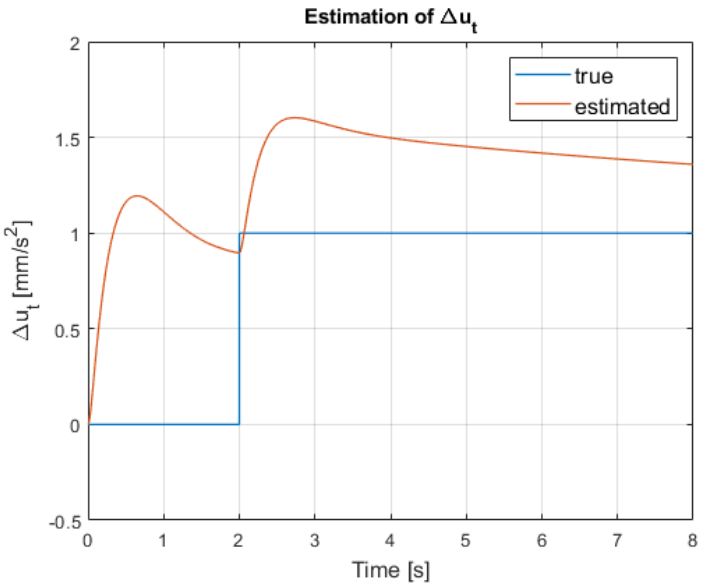



**Figure 6-10:** Unknown Input $\Delta u_t$ estimation of data produced by a non-linear vehicle model using a linear Kalman filter.

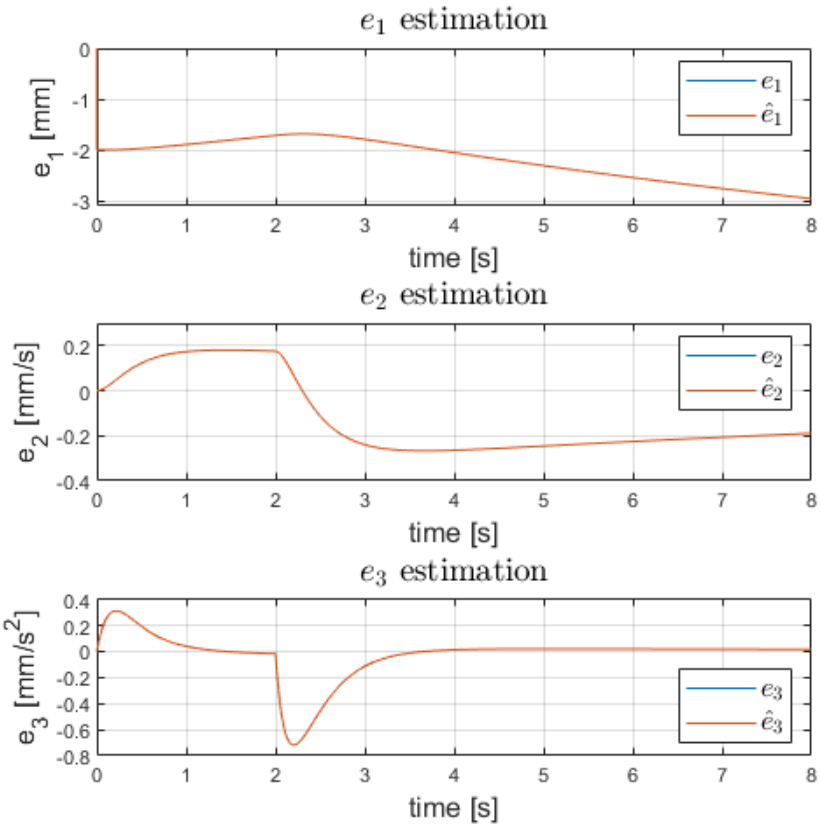



**Figure 6-11:** State estimation of data produced by a non-linear vehicle model using a non-linear SMO.

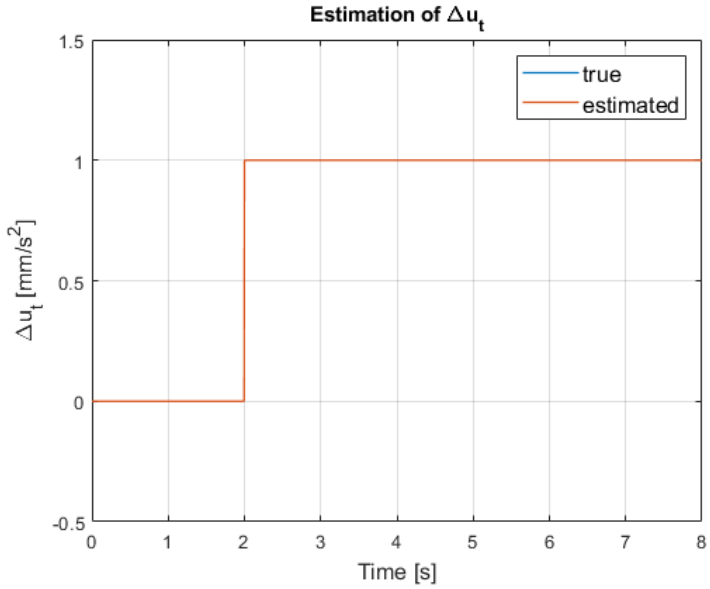



**Figure 6-12:** Unknown Input $\Delta u_t$ estimation of data produced by a non-linear vehicle model using a non-linear SMO.

# Chapter 7

# Experimental Results

The previous chapter showed promising simulation results of the unknown input reconstruction of vehicles subject to a False Data injection cyber attack while obeying an Intersection Control (IC) protocol. However, the simulations do not include the network induced complications, such as communication delay and package loss. Thus, in order to analyze the performance of the Sliding Mode Observer (SMO) in a setting closer to reality, the SMO is tested on a experimental set-up containing small electric vehicles.

The test set-up used for the experiments is a racing track with programmable racing cars, called Anki Overdrive [6]. The track is built by connecting different parts together, such as turns or intersections. An example of such a track is shown in Figure 7-1a. The vehicles have an internal path following controller, thus only the longitudinal control had to be implemented.

This chapter begins with a short explanation of the experimental set-up, whereafter the challenges of the set-up are discussed. Next, it is shown how the parameters were designed that are used in the experiments. Finally, the results for the linear and non-linear SMO are presented.

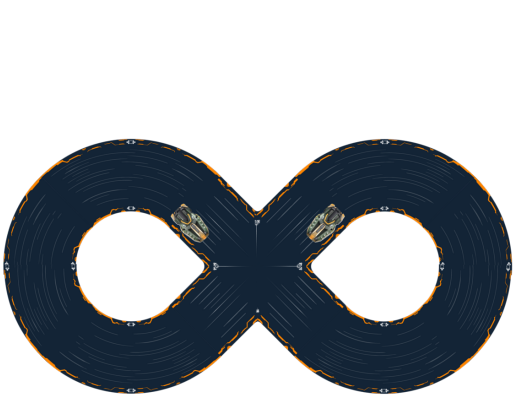
## 7-1 Experimental set-up

In the experimental set-up, the aim is to resemble the simulation set-up as described in Chapter 6 as close as possible. However, especially the communication between the vehicles is different from the simulation set-up, which will be shown in this section.

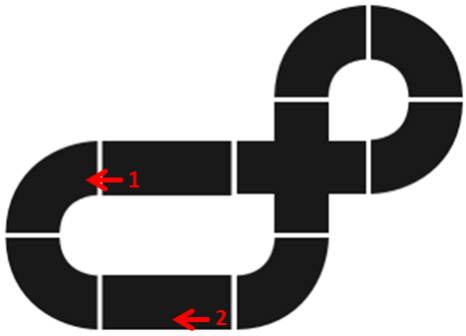
This section begins with a short description of the experiment, after which the discrete control law is elaborated. And finally, it is explained how the vehicles measure their position and velocity, and how the communication protocol is implemented.

### 7-1-1 Experiment Description

Two Anki Overdrive vehicles drive on a track as shown in Figure 7-1b, towards the intersection. In Figure 7-1b, the starting positions and directions of the target vehicle and the follower



(a) Anki Overdrive track and vehicles example from [6].




(b) Used track set-up in experiments. Starting position of the target vehicle (1), and the follower vehicle (2) are indicated in the figure.

**Figure 7-1:** Anki Overdrive tracks.

vehicle are indicated. The target vehicle —which is subject to data alteration— is driving in cruise control mode, and the follower vehicle implements the Virtual Platooning (VP) protocol.

After a few seconds, the data broadcasted by the target vehicle is subject to data alteration. Instead of receiving the true implemented control input $u_t$, the follower vehicle receives $u_{\text{rec}} = u_t + \Delta u_t$. Two different $\Delta u_t$ signals are tested. The first signal is a step signal, with final step value $100\,mm/s^2$. The second signal is a sine wave with an amplitude of $50\,mm/s^2$ and a period of $T = 2.5\,s$. The collected data from the experiments is afterwards implemented in the SMO in order to reconstruct the altered data $\Delta u_t$.

### 7-1-2 Discrete Control Law

The control methods are all implemented in discrete-time with a constant sampling time $t_k = kH$, where $H$ is the sampling size and $k \in \mathbb{Z}^+$. The discrete-time version of Cruise Control is

$$u(k+1) = -k_{cc}(v(k) - v_{\text{ref}}) + a_{\text{ref}}. \tag{7-1}$$

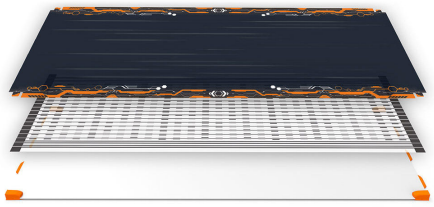
The discrete-time version of the VP control mode is obtained by using the forward Euler method

$$u(k+1) = u(k) + H\,\frac{1}{h}\left(u_t(k) - u(k) + k_p e_1(k) + k_d e_2(k)\right), \tag{7-2}$$

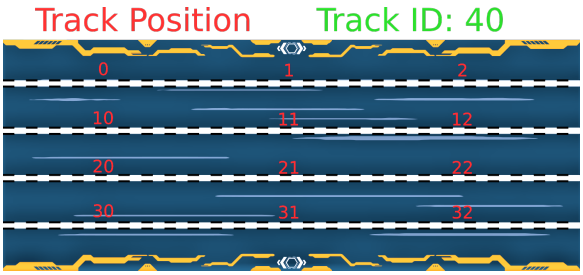
The settings of the parameters used in the experimental set-up are defined in Section 7-3.

### 7-1-3 Anki Overdrive Characteristics

This section summarizes how the state of the vehicles are measured, and how the vehicles communicate with each other.



**(a)** The layers present in an Anki track from [6].




**(b)** Example of location information that the embedded encoding patterns (red) contain on each track piece (green) from [6].

**Figure 7-2:** Structure of an Anki track.

**Measurements** The Anki vehicles measure their position and velocity by scanning an embedded encoding pattern in the tracks with an infra-red sensor. Figure 7-2a shows that the Anki tracks consist of three layers. The middle layer contains the embedded encoding patterns used to determine the location of the vehicle. The bottom and top layers are for support and protection.

The scanned patterns contain information about the ID of the track piece, and the specific location on this track. Figure 7-2b shows a straight piece with four possible lane position codes displayed in red. Each piece has 16 of such lane markers. The velocity of the vehicles is also deducted from the lane markers with an internal controller. Since the position and velocity measurements are only updated when the lane markers are passed, the measurements are event-triggered.

**Communication** The Anki vehicles cannot communicate with each other, or measure the state of the other vehicle. Therefore, a different method is implemented to imitate the Vehicular Ad-Hoc Network (VANET) protocol. Figures 7-3 and 7-4 show the communication in the experimental set-up. The vehicles send their measurements via Bluetooth to a Python API [5], which uses the Anki Drive SDK to unwrap the received messages. The Anki Drive SDK is an open source C implementation of the message protocols and the data parsing routines [87]. A program developed in Python calculates the needed control input based on the received measurements. The vehicles are represented in a separate thread in the Python program. Once a vehicle needs information from the target vehicle, the information is requested from the object representing the target vehicle. After the needed control input is calculated, the command is transmitted back to the vehicles via Bluetooth, using the Drive SDK to wrap the messages.

## 7-2 Challenges in the experimental set-up

Transitioning from simulations to an experimental set-up is known to bring about new challenges. Some of them can be neglected, while others need to be solved beforehand. Some of the challenges are:

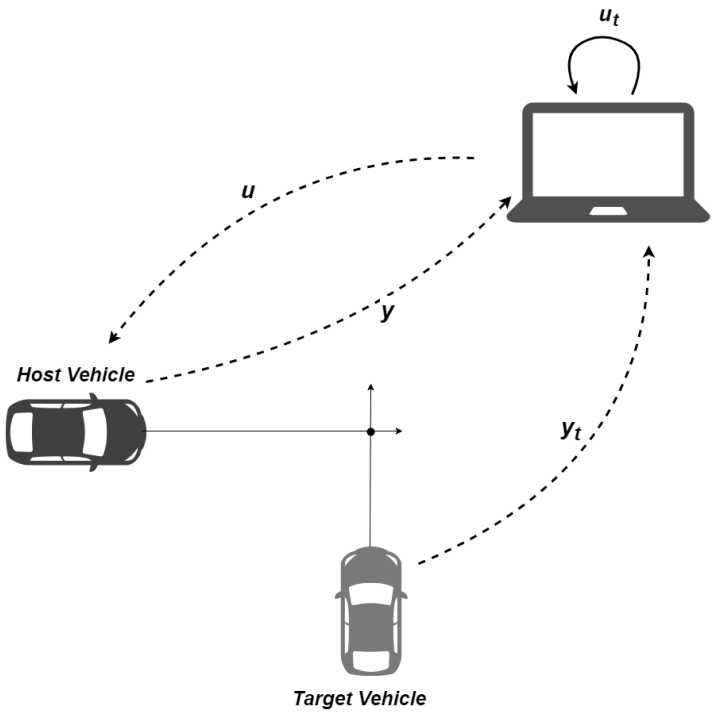



**Figure 7-3:** Communication in the experimental set-up of the host vehicle. The dashed line indicates wireless communication via Bluetooth.

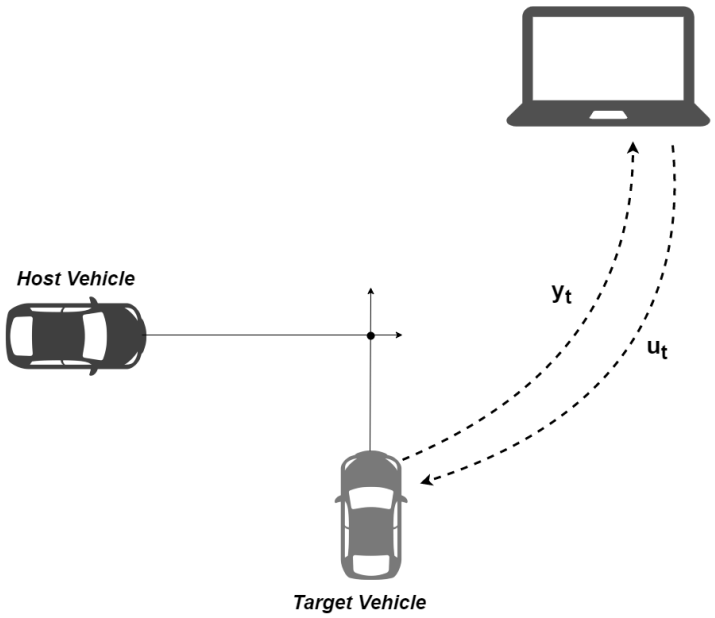



**Figure 7-4:** Communication in the experimental set-up of the target vehicle. The dashed line indicates wireless communication via Bluetooth.

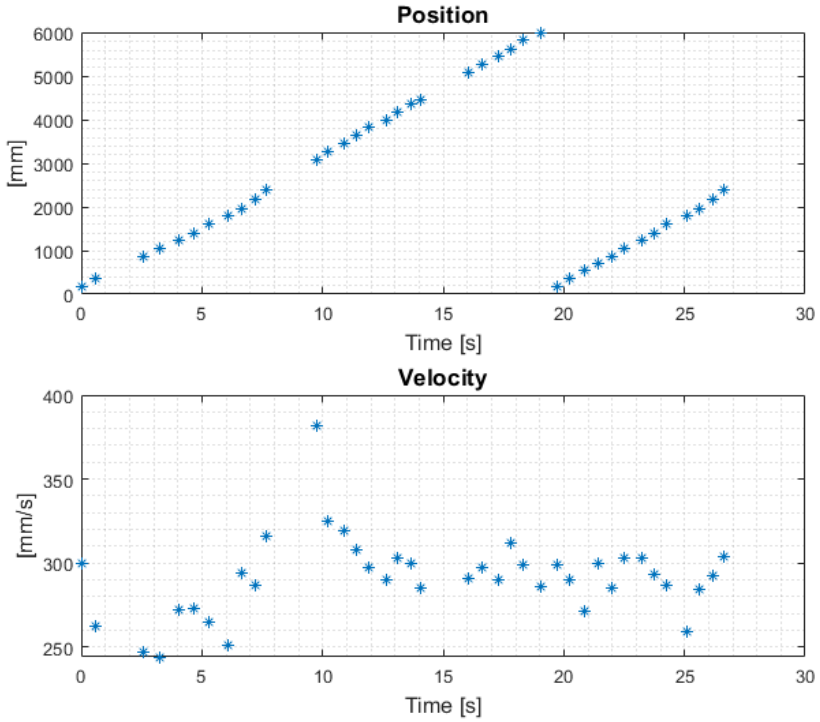



**Figure 7-5:** Position and velocity measurements of one Anki vehicle. Once the vehicle surpasses the starting position again, the location measurement jumps back to zero.

- *Communication delay*: The vehicles send and receive the measurements and control data through Bluetooth. Since no communication channel is able to transfer data instantaneously, there is a communication delay. The communication delay has a direct influence on the stability of the VP. It is therefore important that the string stability of the model (7-2) is analyzed while a communication delay is present. From the analysis of the string stability, the step size $H$ of the control law can be chosen. In Section 7-3, the choice of $H = 0.1\,s$ is substantiated, based on the string stability analysis.

- *Infrequent measurements*: The most challenging limitation of the experimental set-up is caused by the event-triggered measurements. From Figure 7-5 it can be seen that the measurements are infrequent and time-varying. The measurements are received around every 0.5 to 1 seconds, with occasional gaps of as much as 2 seconds. However, the control input needs to be implemented at least every $0.1\,s$ to maintain a stable platoon consisting of Anki vehicles. In order to produce data in between measurements based on the vehicle dynamics model (3-3), a discrete-time Kalman Filter is implemented. Section 7-3 shows the parameters used in the Kalman filter.

- *Operational velocity*: Below a certain velocity, the Anki vehicles start to move erratically. The velocity level where this starts to happen is at $v = 200\,mm/s$. An easy solution is to choose a reference velocity higher than the indicated velocity. However, it is desirable that the vehicles participate in the VP as long as possible for the Unknown Input analysis afterwards. Thus a reference velocity of $v = 300\,mm/s$ is chosen, such that it is not much higher than the indicated level but still allows room for deceleration.

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $k_p$ | 0.6 | | |
| $k_d$ | 1 | $Q_k$ | $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ |
| $k_{cc}$ | 1 | | |
| $H$ | 0.1 | | |
| $H_{obs}$ | 0.01 | $R_k$ | $\begin{bmatrix} 2000 & 0 \\ 0 & 3000 \end{bmatrix}$ |
| $\tau$ | 0.002 | | |

**Table 7-1:** Parameters used in the experimental set-up.

- *Internal controller*: The vehicles used in the experiments have an internal controller that regulates the control input prior to executing it. Ideally, a negative acceleration profile will lead to a deceleration of the vehicle's velocity. However, in order to decelerate the Anki Overdrive vehicles, it is necessary to supply the desired velocity alongside the acceleration profile to reach this velocity. Thus, if the desired velocity is lower than the current velocity, the vehicle will decelerate with the provided control input. In order to conform to this regulation, the control input is implemented as

$$\begin{cases} v = 1000\ mm/s, & u = u, & \text{if } u > 0, \\ v = 0\ mm/s, & u = |u|, & \text{if } u < 0, \\ v = v\,, & u = 0 & \text{otherwise.} \end{cases} \tag{7-3}$$

  The reason for choosing $v = 1000\,mm/s$ is that the reference velocity is much lower. This will prevent the internal controller of the Anki vehicles from interrupting the implementation of the control input.

  Another disadvantage that rises from the internal controller is that there is no knowledge on how the internal controller exactly operates. It is not known if there is another regulation before the control input is implemented. This could lead to unknown non-linearities in the vehicles' behaviours.

- *Saturation on control input*: Since the control input is equal to the desired acceleration, the control input is bounded by the physical limits. If the vehicle is not able to execute the provided control input, it will not behave according to the vehicle model which has an adverse effect on the unknown input estimation. A solution is to implement a saturation to the control input prior to the unknown input estimation. The saturation on the control input is chosen as $u \in \begin{bmatrix} -500 & 500 \end{bmatrix}$.

## 7-3 Parameter Design

All the parameters used in the experiment are defined in Table 7-1. The reason behind the choices of the parameters $H$, $\tau$, $Q_k$ and $R_k$ are elaborated below.

### 7-3-1 Choice of Engine Time Lag $\tau$

To estimate the engine time lag $\tau$, an experiment is performed on the Anki vehicles. A vehicle drives on the Anki track while maintaining a constant velocity of $v = 300\,mm/s$. Next, the vehicle switches to $v = 500\,mm/s$ with maximum acceleration. The length of the duration before the vehicle reaches the new velocity $v = 500\,mm/s$ can be used to estimate $\tau$.

From `Matlab`'s System Identification Toolbox [75], it is derived that the motor dynamics constant is equal to $\tau = 0.002$.

### 7-3-2 Choice of Control Sample Size $H$

As previously elaborated in Chapter 2, it is important that the vehicles participating in a Virtual Platoon satisfy the string stability requirement. String stability implies that a disturbance in the velocity of one vehicle will not be amplified throughout the platoon. Network induced effects, such as the sampling time and the communication delay, can cause the VP to become unstable. Therefore, it is important that the string stability is analyzed while the typical network effects are present, such that an appropriate sample size $H$ can be chosen.

To analyze the string stability, the method from Öncü et al. [86] is adjusted such that it fits the dynamics of the vehicles from the experimental set-up. The complete analysis is provided in Appendix C.

**Definition 7.1**
A VP is called string stable when the following equation holds [86]

$$\left| \frac{\Delta_i(e^{jw})}{\Delta_{i-1}(e^{jw})} \right| \leq 1 \ \forall w,\ i = 1, ..., n, \tag{7-4}$$

where $\Delta_i$ is the $\mathcal{Z}$-transform of the velocity of vehicle $i$, and $n$ is the amount of vehicles present in the VP. □

From Figures 7-3 and 7-4, it can be seen that the vehicles send their velocity to a laptop, which calculates and broadcasts the needed control input. Therefore, both vehicles experience a communication delay, even when only the cruise control mode is implemented. The communication delay caused by the remote calculations is modelled as a constant actuator delay $\theta_a$. When needing information from the target vehicle, the delay is modelled as the communication delay $\theta$.

The string stability analysis is implemented in `Matlab` and varied over the parameters $H$, $\theta$ and $\theta_a$. From an experiment on the Anki vehicles, an estimation of the actuator delay is obtained. The experiment is performed by sending the command to switch on the lights along with the desired control input. By timing the duration of calculating the control input, and switching on the lights, it appears that the actuator delay $\theta_a$ is at most $\theta_a = 0.3\,s$. Therefore, the string stability is analyzed for $0 < \theta_a \leq 0.4$ and $0 < \theta \leq 0.9$. With a step size of $H = 0.1$, the VP is string stable for the whole range of $\theta_a$ and $\theta \leq 0.37\,s$. Thus, with a step size of $H = 0.1$, the VP is string stable within the estimated actuator delay of $\theta_a \leq 0.3\,s$ and the communication delay $\theta \leq 0.37\,s$. From Figure 7-3 it can be seen that the communication delay $\theta$ will be approximately equal to the actuator delay $\theta_a$. Therefore, it is concluded that the VP is string stable within the estimated communication delay, with a sample size of $H = 0.1\,s$.

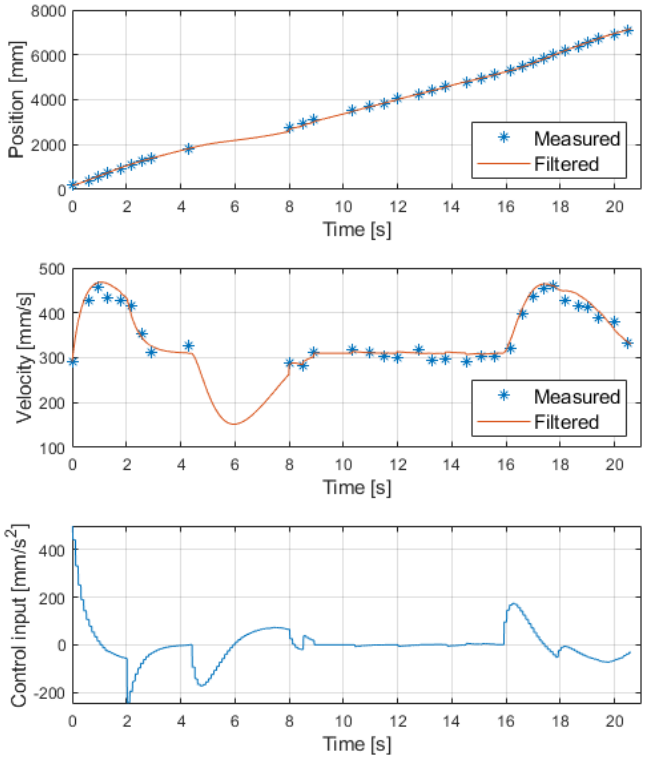



**Figure 7-6:** Kalman Filter outputs (red) and measurements (blue) of an Anki vehicle. Top: position over time. Middle: velocity over time. Bottom: implemented control input.

### 7-3-3 Kalman Filter Parameters

In order to produce state information within the measurements, a discrete-time Kalman filter is implemented with sample size $H_{obs} = 0.01\,s$. The Kalman filter implements the discretized linear vehicle model from Equation (3-3). Prior to implementing the Kalman filter, the noise covariance matrices, $Q_k$ and $R_k$, need to be determined. However, the Anki Overdrive characteristics do not include any accuracy information on the measurements, or even how the velocity is derived from the coding patterns. The initial guess for $R_k$ is that the location measurements are more accurate than the velocity measurement. After some tuning the matrices are chosen as

$$Q_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_k = \begin{bmatrix} 2000 & 0 \\ 0 & 3000 \end{bmatrix}. \tag{7-5}$$

Figure 7-6 shows the Kalman Filter outputs of the target and host vehicle, respectively. Instead of using the absolute position of the vehicles on the track, the total driven distance is used. The vehicles drive on a closed track, thus the absolute position jumps back to zero at the start of a new round, which disturbs the state estimation of the Kalman filter.

## 7-4 Experimental Result Linear Sliding Mode Observer

The control inputs and measurements are collected from the experiments, such that the SMOs designed in Chapter 5 can be tested afterwards.

**Design Matrices Sliding Mode Observers** The linear SMO as designed in Chapter 5 is used, where the design matrices $G_l$ and $G_n$ from Equation (5-7) containing the parameters from Table 7-1, are equal to

$$G_l = \begin{bmatrix} 0 & 0 & 0 \\ 0.2307 & -9.18e04 & 4.47e04 \\ -4.4072e04 & -20.26 & -0.2307 \\ -20.258 & -1.4762e07 & 9.814e04 \end{bmatrix},$$
$$G_n = \begin{bmatrix} -0.0012 & 490.69 & -223.6 \\ 2.3068e-04 & -0.0203 & -2.3068e-04 \\ -0.0203 & -1.476e04 & 98.138 \end{bmatrix}, \tag{7-6}$$

with the remaining design matrix equal to $(P_0\mathcal{D}_2)^T = \begin{bmatrix} -0.0016 & 3.4372 & 7.5428 \end{bmatrix}$.

**Reconstruction Unknown Input Step Signal** Since the SMO is designed in continuous-time, the control input used in the experiments is transformed from discrete to continuous-time by applying a zero-order hold. Since the SMO needs a continuous stream of measurement data, the data derived from the Kalman filter is used as an input to the observer.

Figures 7-7 and 7-8 show the state estimation and unknown input estimation, respectively, where $\Delta u_t$ is a step signal. The peaking structure in both the state and $\Delta u_t$ estimation are caused by the discrete control input $u$. The control input $u$ is calculated by the discrete control law in Equation (7-2), while the SMO expects the control input to behave according to the continuous-time model (3-1). Counting the peaks in the $\Delta u_t$ estimation in Figure 7-8, reveals that there are ten peaks in one second, each around $0.1\,s$ apart, which coincides with the control step size $H$. Therefore, it can be concluded that the discretization of the control law causes these spikes.

The large peaking structure in $e_3$ are also caused by the discrete control input. The Kalman filter calculates the position and velocity on the basis of the control input, thus at the next control input $u(k+1)$, the velocity changes accordingly. The derivative of the sudden change of $e_2$ leads to the large peaking structure in $e_3$. The larger peaks in the $\Delta u_t$ estimation is caused by the measurement update in the Kalman filter. Especially in the state $e_1$, the effect of the measurement update is visible. At exactly the points that $e_1$ has a jump, a larger peak can be perceived in the $\Delta u_t$ estimate. The proper solution for a more accurate $\Delta u_t$ estimation would be to discretize the SMO. Furthermore, using a discrete-time version of the SMO enables the possibility to implement the SMO in real-time.

**Reconstruction Unknown Input Sine Wave** Figures 7-9 and 7-10 show the state estimation and unknown input estimation, respectively, where $\Delta u_t$ is a sine wave. The same phenomena as previously described are visible in the $\Delta u_t$ estimation. Even though the estimation is noisy, the unknown input is estimated accurately.

### 7-4-1 Discussion Linear Sliding Mode Observer Result

A few important notes must be made. Even though the unknown input estimation is close to the true unknown input, an important assumption is not tested in the current experimental set-up. The unknown input reconstruction relies on the prospect that the state $e_2$ is measurable. Since the Anki Overdrive set-up is not suitable to test this important assumption, a next step would be to test the whole VANET working principle in a larger experiment. Nevertheless, the executed experiments still provide an important insight on the possibility to reconstruct the unknown input, while network induced effects are present.

Secondly, the data used for the unknown input reconstruction is obtained from the Kalman filter. The Kalman filter implements a simplified linear vehicle model. Thus, the data used to reconstruct the unknown input is already filtered according to the linear vehicle model. Therefore, every time a measurement update is provided, the unknown input estimation becomes noisy.

## 7-5 Experimental Result Non-Linear Sliding Mode Observer

The important advantage of the SMO is the simplicity with which the linear model can be exchanged for a non-linear model. Therefore, it is the goal to compare the result of the non-linear SMO with the linear SMO obtained from the experiments. Unfortunately, the collected data from the Anki Overdrive experiments is inadequate to test the non-linear vehicle model on. This section elaborates on which methods were tried, and the reasons for why the collected data is not suitable for the desired tests.

### 7-5-1 Measurement Data for the Non-Linear Vehicle Model

As is shown in Figure 7-5, the position and velocity measurements are received around every $0.5 - 1$ seconds. However, the intended control input from the target vehicle $u_t$, is received every $0.1\,s$. Thus, to apply a non-linear SMO to the measured and received data, it is necessary that there is state information in between the measurements. During the experiments, a linear Kalman filter was implemented to produce the needed data. However, as will be explained below, the data obtained from the Kalman filter is inadequate for the non-linear SMO. Therefore, an alternative method is implemented.

**Measurements Obtained from the Kalman Filter** As shown in Section 7-3-3, a linear Kalman filter was used to produce data in between the measurements. However, since the measurements are so infrequent, the Kalman filter runs in open loop for a considerate amount of time between two measurements. Thus, at every measurement update, the estimated state is not matching the measurement which results in a jump in the state estimates. Figure 7-11 shows

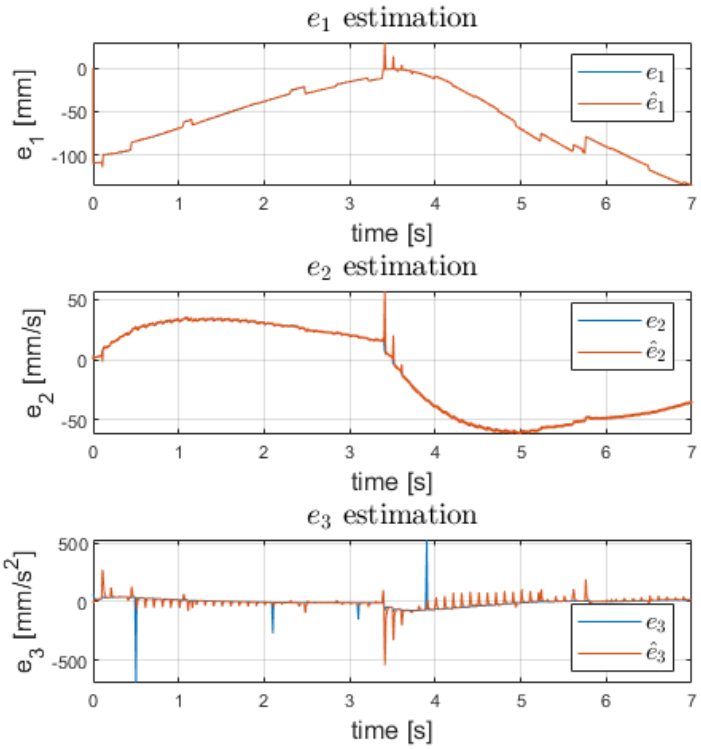



**Figure 7-7:** State estimation of the VP from the perspective of the host vehicle. The target vehicle is subject to data alteration. The data alteration is represented by a step signal.

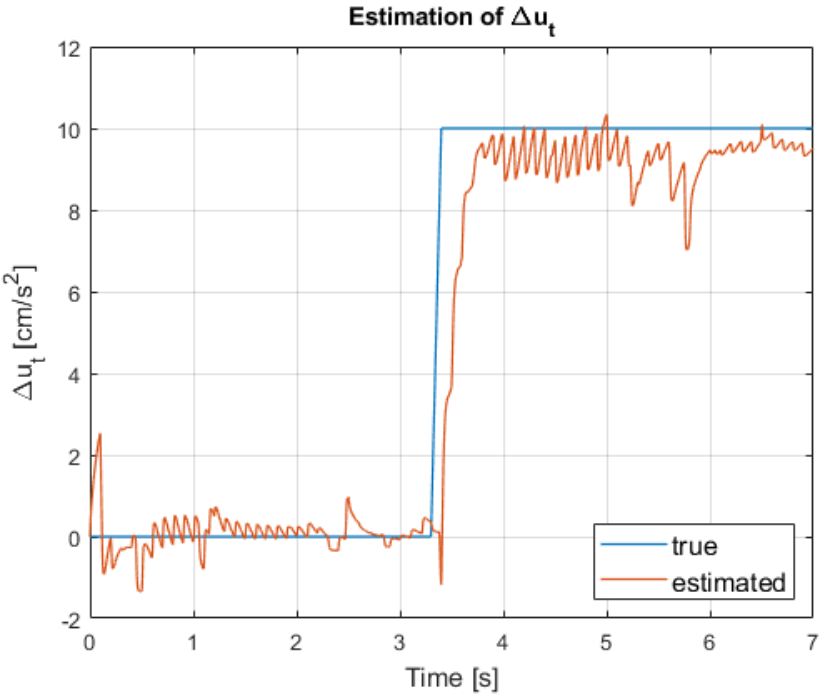



**Figure 7-8:** Unknown input $\Delta u_t$ estimation using a SMO.

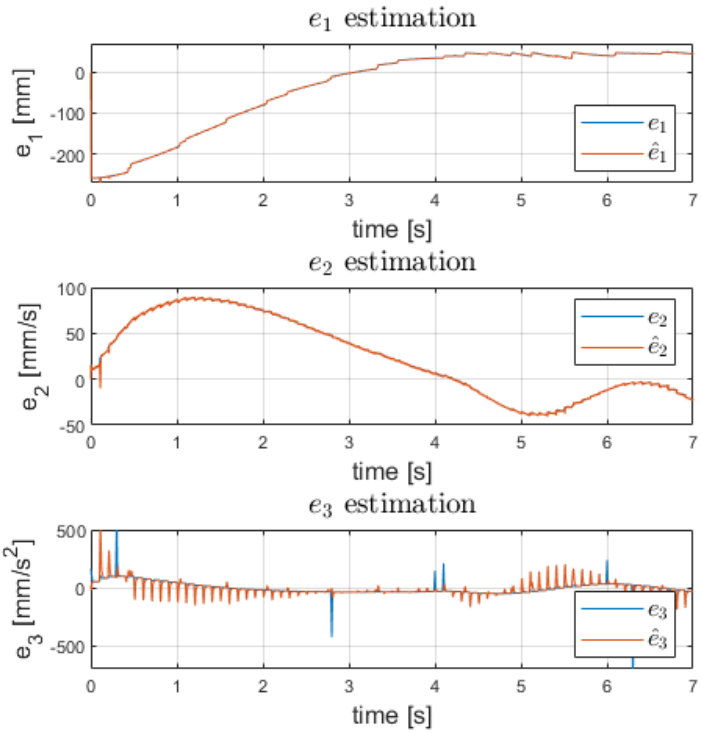



**Figure 7-9:** State estimation of the VP from the perspective of the host vehicle. The target vehicle is subject to data alteration. The data alteration is represented by a sine wave.

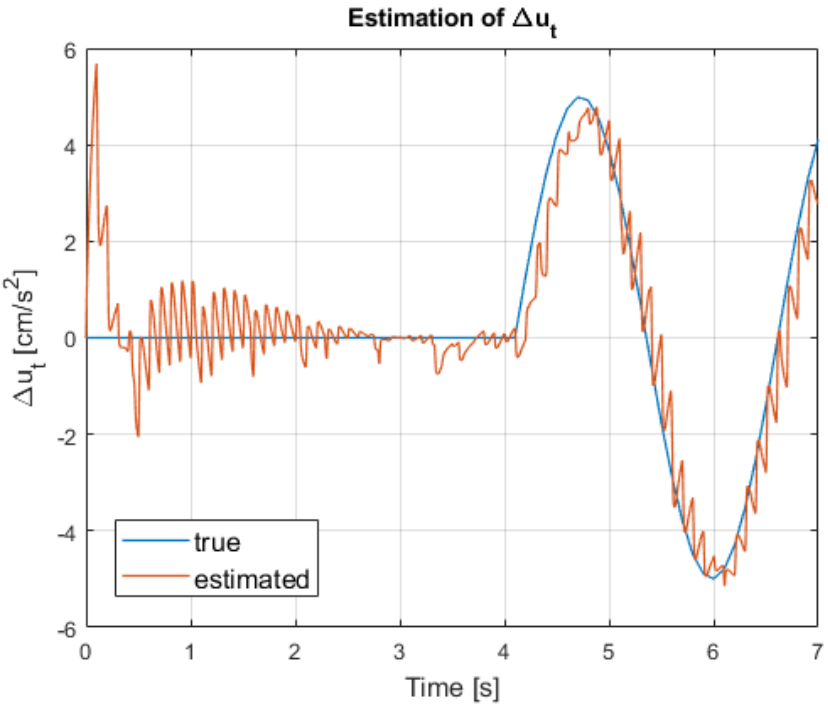



**Figure 7-10:** Unknown input $\Delta u_t$ estimation using a SMO.

a close-up of this phenomenon. It can be seen that the infrequent measurements lead to a 'staircase-wise' behaviour in the state estimation. Thus, the non-linear vehicle dynamics are neglected at the time instants where no measurement data is available. Therefore, the data obtained by the Kalman filter is not suitable to fit a non-linear vehicle model on.

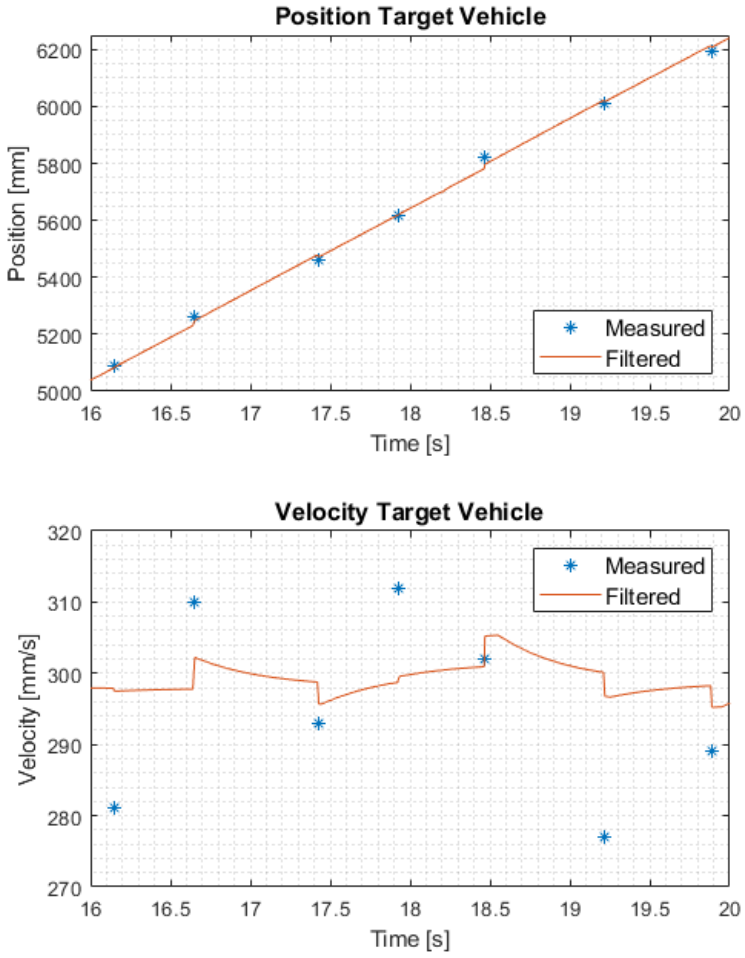



**Figure 7-11:** Close-up of the measurements of the target vehicle using a Kalman filter.

**Alternative Collection of Measurement Data** Since the data from the Kalman filter cannot be used to fit a non-linear vehicle model, an alternative needs to be implemented. As indicated before, it is not possible to solely use the measured data points as an input for the unknown input estimation, due to the fact that the control input from the target vehicle is received more frequently than the measurements. As an alternative, a concept is considered where a non-linear model is fitted as close as possible to the measurements for every data set. The data generated by the measurement models is used as the measurements to test the SMO.

Next, a general vehicle model is fitted to all available data sets. Around 40 data sets are collected from the Anki Overdrive experiments, which are split into a training set, validation

set and a testing set. The training set contains around 60% of the data, the validation set 30% and the test set contains the final 10%. A general non-linear vehicle model is obtained by fitting a model to the training data, use to predict the data in the validation set of which the result is used to alter the non-linear model, and finally, the test set is used for a concluding unbiased evaluation of the model.

The general non-linear model is used for to design an SMO. Next, the data produced by the measurement models is used to test the performance of the reconstruction of the unknown input by the SMO.

### 7-5-2 Non-Linear Vehicle Models

The first step to test the non-linear SMO is to find a non-linear model that fits the measured data from the experiments. Different non-linear models are tried, such that the model closest to the measurements can be used for the non-linear SMO design.

**Theoretical Non-Linear Vehicle Model** The first non-linear model that is tried to fit the experimental measurements, is the non-linear vehicle model described in Equation 6-3. The coefficients that need to be identified from the Anki data are $K_d$, $d_m$, $m$ and $\tau$.

**Polynomial Non-Linear Model** The second non-linear model is the non-linear polynomial, as stated below

$$\dot{a} = k_0 a + k_1 u + k_2 v + k_3 v^2 + k_4 v a + k_5, \tag{7-7}$$

where $k_i$ are the design parameters. Other models that are tried have minor variations on the polynomial given in Equation 7-7. For example, a variation of the model (7-7) is to make the parameter $k_0$ or $k_1$ dependent of the velocity.

### 7-5-3 System Identification Methods

In order to obtain a non-linear vehicle model to the Anki vehicles, the parameters from the models need to be identified. Three different system identification techniques are tried.

**Fmincon and Lsqnonlin** The first two solvers that are tried are `fmincon` and `lsqnonlin`. `Fmincon` finds the minimum of the function $f(x)$ for $x$, where an optional bound on $x$ can be provided. The objective function $f(x)$ is given by

$$f(x) = \sum_N \frac{(\hat{y}(k) - y(k))^2}{N}, \tag{7-8}$$

where $x$ is a vector containing the coefficients that needs to be identified, $N$ is the length of the data set and $y(k)$ are the outputs measured at $k$. The estimation $\hat{y}(k)$ is obtained using `Matlab`'s differential equation solver `ODE45` on the model (6-3), and interpolating $\hat{y}(t)$

at the desired data points $k$. The lower and upper bounds of $x = \begin{bmatrix}\tau & m & K_d & d_m\end{bmatrix}$ are $lb = \begin{bmatrix}0.00001 & 0.00001 & 0.00001 & 0\end{bmatrix}$ and $ub = \begin{bmatrix}1 & 10 & 1 & 10000\end{bmatrix}$.

`Lsqnonlin` has a similar working principle, where it finds the minimum for $||f(x)||_2$ by varying $x$, where again a lower and upper bound can be provided for $x$. The objective function is equal to $\hat{y}(k) - y(k)$.

However, it turns out that both objective functions have multiple local minima. For example, Figure 7-12 shows the value of the objective function $f(x)$ over $\tau$ obtained from the solver `fmincon`, while the remaining identification variables are kept constant. Solvers such as `fmincon` and `lsqnonlin` use the initial point of $x$ to find the minimum of the objective. Thus, the optimizer quickly gets trapped at a local minima that is close to the initial point. Therefore, a different optimizer is used.

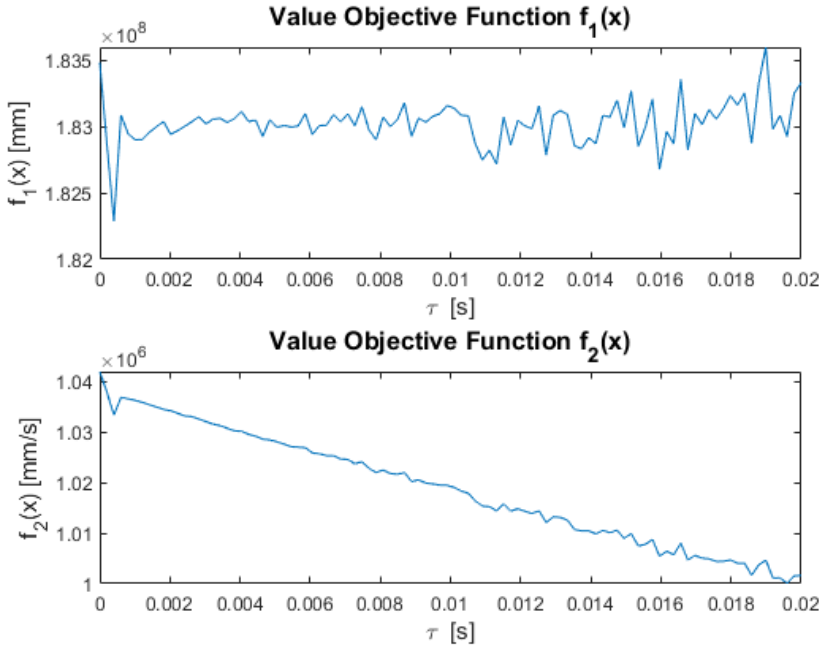



**Figure 7-12:** Value of objective function varying over $\tau$, while other identification variables are kept constant. Objective functions $f_1(x)$ and $f_2(x)$ are the quadratic errors between the estimated and measured position and velocity, respectively.

**Genetic Algorithm** An optimizer that is known to work well with functions containing local minima, is a Genetic Algorithm (GA). A GA is a global search technique that is inspired by Darwin's evolution theory [64]. The algorithm starts with a population of candidates representing the parameters, whose fitness are evaluated. Based on the fitness of the candidates, mutations and crossovers are used to produce a new generation. Due to the parallel evaluation of multiple candidates, the algorithm is able to escape local minima. Figure 7-13 shows the velocity estimation of eight different data sets taken from the training data, using the non-linear vehicle model given in Equation (5-34) together with the coefficients obtained from the GA. The model coefficients obtained from the GA are $x = \begin{bmatrix}\tau & m & K_d & d_m\end{bmatrix} = \begin{bmatrix}0.0582 & 1.2124 & 9.4133 \times 10^{-6} & 0.2652\end{bmatrix}$.

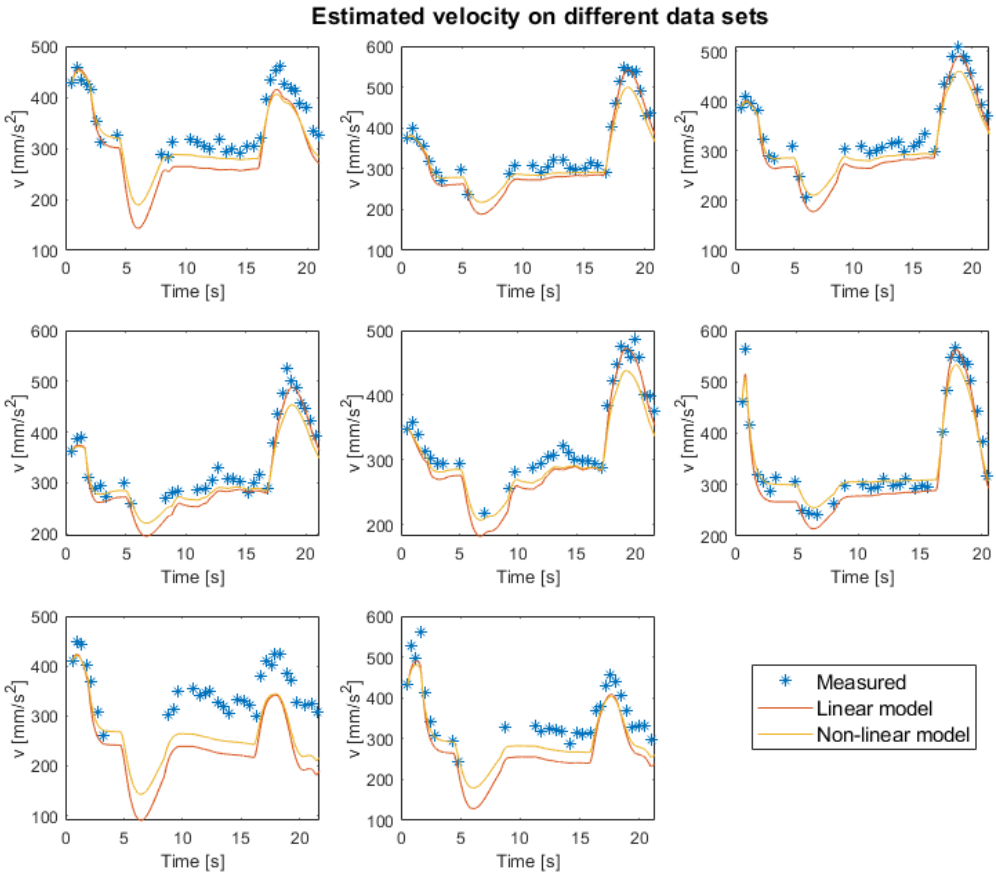



**Figure 7-13:** Estimated velocity using the non-linear vehicle model fitted on the training data sets, of which eight are shown.

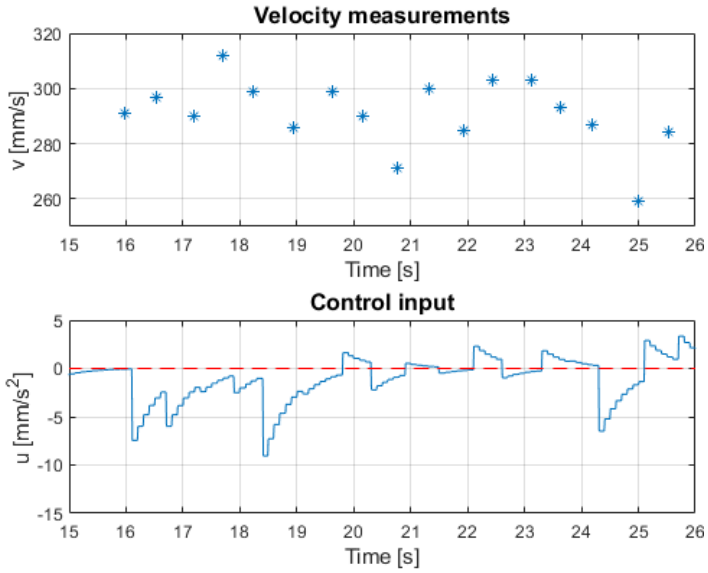



**Figure 7-14:** Close-up of velocity measurements (top), together with the intended control input (bottom).

### 7-5-4 Analysis of the Identified Parameters

In Figure 7-13 it can be seen that the non-linear vehicle model does not perform significantly better than the linear model. Using the non-linear model together with the found coefficients, the value of the objective function is equal to $f(x) = 9.228 \times 10^4$. The value of the objective function is five times better compared to the value of the objective function using the linear model $f(x) = 5.39 \times 10^5$. However, from Figure 7-13 it appears that the non-linear model does not completely explain the behaviour of the vehicles.

Although different non-linear vehicle models are tried, the value of the objective function did not significantly improve. This implies that the relation between the control input and the measurements may not be consistent throughout the data-sets.

Upon closer inspection of the data, a remarkable phenomenon is found. Figure 7-14 shows a close-up of the measured velocity of the vehicle, together with the used control input. It is noticeable that even though the control input is below zero from $t = 15\,s$ until $t = 19.8\,s$, the vehicle accelerates at $t = 16\,s$, $t = 17.5\,s$ and $t = 19.7\,s$. Furthermore, the control input is above zero at $t = 22.1\,s$, where the vehicle indeed accelerates. However, at $t = 23.3\,s$ the vehicle decelerates while approximately the same control input is given.

These inconsistencies are present throughout the whole data set, which greatly hinders the fitting of a non-linear model to the measurements. The inconsistencies between the control input and the velocity are especially noticeable when the control input is around zero, which could indicate that the internal controller of the Anki vehicles interferes when implementing a low control input. However, it could also mean that the measurement noise is much higher than expected, implying that the phenomenon is also present at higher control inputs, but is less visible.

Despite the cause of the inconsistencies in the data, the conclusion is that the collected data is unsuitable for identifying the non-linear dynamics. Unfortunately, this means that the experimental data cannot be used to test the non-linear SMO.

## 7-6 Discussion

In this chapter, the linear SMO is tested on the data collected in experiments. Although the estimation of the unknown input is noisy, the linear SMO is able to reconstruct the unknown input. Before this research, no experiment has been conducted to detect the presence of false data in vehicles performing an IC concept.

Unfortunately, it is concluded that the data obtained form the Anki Overdrive is unsuitable to fit a non-linear vehicle model on. Therefore, the non-linear SMO could not be tested on the experimental data. Nevertheless, the previous chapter shows promising simulation results. Thus, the non-linear SMO as designed in Chapter 6 makes a quick implementation possible on a more accurate experimental set-up. It is therefore recommended for future research to test the non-linear SMO on an experimental set-up that has more frequent measurements, and with more information on the noise characteristics.

Furthermore, since the vehicles implement a discrete control law, it is recommended to discretize the SMO as well. Using the discrete-time SMO, it is possible to implement the observer in real time in order to analyze the data per received package.

# Chapter 8

# Conclusions and Discussion

Providing autonomous vehicles with the means to communicate with each other enables the possibility to automate intersections. However, the communication channel presents a dangerous vulnerability to different cyber attacks. In this thesis, it was shown that a false data injection attack with a period of 1.3 seconds can cause a collision on the intersection. To this end, a Sliding Mode Observer (SMO) was designed in this thesis to detect and reconstruct the injected false data. The designed observer was tested in different simulation scenarios and on an experimental set-up.

This chapter presents concluding remarks together with a discussion reflecting on the study as a whole. From the discussion, recommendations for future research are presented.

## 8-1 Conclusions

The main research questions will be answered in this section, based on the conclusions found per chapter. The main research questions as presented in Chapter 1 are

1. *If connected autonomous vehicles are participating in an Intersection Control protocol, to what extent should the broadcasted data from the surrounding vehicles be altered in order to cause a collision?*
2. *Is it possible to detect the presence of anomalous data and reconstruct it, in the case that the transmitted messages from connected autonomous vehicles participating in an Intersection Control concept are subject to data alteration?*

### 8-1-1 First Research Question

The type of transmitted data depends on the Intersection Control (IC) technique, thus to investigate to what extend the transmitted data needs to be altered to cause a collision at the

intersection, an IC technique needs to be chosen. In Chapter 2, the chosen IC technique was a Virtual Platooning (VP) protocol. The VP protocol guarantees a collision free intersection by implementing a safe virtual inter-vehicle distance. The safe inter-vehicle distance is kept by using both the measured data, as well as the data received through wireless communication, which is the intended acceleration of the target vehicle. In Chapter 3, the transmitted data $u_t$ was altered with $\Delta u_t$ in simulations. From the simulations, several conclusions can be drawn.

**Timing of Data Alteration** In order to maintain a safe virtual distance to the target vehicle, the vehicle uses both data obtained from on-board sensors, as well as the received data from the target vehicle in the control law. Thus, to cause a collision on the intersection, the timing of the altered data needs to be correct. In the case that the data alteration of $u_t$ is temporarily and is timed too early, the vehicles will recover to a safe inter-vehicle distance before the intersection is reached.

**Data Alteration** The received data is split into two parts: $u_{t,\text{rec}} = u_t + \Delta u_t$. Since the received data is the intended acceleration of the target vehicle, $\Delta u_t$ needs to be positive to cause a collision at the intersection. The follower vehicle will need to accelerate to reach the altered $u_{t,\text{rec}}$. In the case that $\Delta u_t$ is negative, the altered data will not cause a collision. However, it will have a negative impact on the vehicle throughput.

The data alteration researched in Chapter 3 were within the physical limits, such that a simple check cannot detect the altered data. While the vehicles have a reference velocity of $3m/s$, it was shown that a temporarily data alteration with the value $\Delta u_t = 4\,m/s^2$ can cause a collision at the intersection within 1.3 seconds.

Furthermore, it was shown that the size of the data alteration has a large influence on the required duration of the false data injection. A step signal with a final value of $\Delta u_t = 4\,m/s^2$ causes a collision within 1.3 seconds, whereas a step signal with a final value of $\Delta u_t = 1\,m/s^2$ needs a duration of 6 seconds.

Another $\Delta u_t$ signal that was researched in simulations is a block wave with a randomized frequency. As expected, as long as the injection of the false data is timed correctly, and the average of the altered signal is higher than $0\,m/s^2$, a collision at the intersection will be the consequence.

### 8-1-2 Second Research Question

The received data was split into two parts, $u_{t,\text{rec}} = u_t + \Delta u_t$, where $\Delta u_t$ is referred to as the unknown input to the system. In this thesis, it was chosen to design a Sliding Mode Observer to reconstruct the unknown input $\Delta u_t$. To be able to design the observer, the important assumption was made that the following vehicle can measure the relative velocity from its target vehicle.

Conclusions on the second research question are provided at three levels, the theory, the simulations and the experiments.

**Theoretical Results** In chapter 4 it was shown how a Kalman filter can be designed to estimate the unknown input of the system. The Kalman filter design was tested in simulations. While the dynamics of the vehicles are produced with a linear vehicle model, the Kalman filter is able to accurately reconstruct the unknown input, even in the presence of measurement noise. However, in a realistic setting, the vehicles will not behave according to the simple linear vehicle model. To this end, an SMO was designed, which is not limited to a linear model when reconstructing the unknown input of the system.

Chapter 5 provides the theoretical design of a linear SMO, and a simple extension to a non-linear SMO. In order to use the SMO for the reconstruction of the unknown input, two requirements need to be met. However, the VP state-space model does not meet one of the requirements. Therefore, a technique to circumvent this requirement was implemented, which involves the estimation of the state where the unknown input enters. The extra state is estimated using a so-called step-by-step observer.

From chapter 5, it is immediately visible that the design of the SMO for anomaly reconstruction is more complex than the Kalman filter. On the other hand, the extension from a linear SMO to a non-linear SMO is obtained by simply extending the observer with the non-linear dynamics.

**Simulation Results** To investigate the performance of both the linear and the non-linear SMO, Chapter 6 provides the simulation results of the observers. Without the presence of measurement noise, the linear SMO can almost perfectly reconstruct the unknown input. However, the step-by-step observer used to estimates the extra state, and the SMO are both high gain observers. High gain observers are known to amplify measurement noise in the state estimates. Therefore, the step-by-step observer was replaced with a robust exact differentiator. Furthermore, a low-pass filter was added to the $\Delta u_t$ estimation to remove the high frequency noise. Using this technique, similar results were obtained as with the Kalman filter under measurement noise. However, the linear Kalman filter does still outperform the SMO designed for a linear vehicle model in the presence of measurement noise.

To test the performance of the non-linear SMO, the dynamics of the vehicles were produced by a non-linear vehicle model. Again, the $\Delta u_t$ estimation is almost a perfect overlay of the true $\Delta u_t$, whereas the Kalman filter is not able to reconstruct the unknown input using the non-linear vehicle dynamics as an input. It must be noted that these simulations are performed without measurement noise. Nonetheless, these simulation results are very promising. The possibility to simply extend the linear SMO with the non-linear dynamics provides a high degree of freedom in the design of the SMO.

**Experimental Results** The simulations showed promising results on both the linear and the non-linear SMO. To test the observers in more realistic circumstances, experiments were performed of vehicles following a VP protocol. The vehicles in the experimental set-up experience realistic complications, such as measurement noise, communication delay, package drops and discretization of the control law. Unfortunately, the experimental set-up introduces an extra complication. The vehicles measure their position and velocity by scanning an encoding patter in the track, resulting in infrequent and uneven spaced measurements. The measurement only took place every $0.5 - 1$ seconds, thus the data could not directly be

applied in the SMO. Therefore, a Kalman filter was implemented to produce data within the measurements.

The collected data was applied to the linear continuous-time SMO to retrieve an estimate of the unknown input $\Delta u_t$. The SMO expects the control input to behave according to the model $\dot{u}$. Thus the estimation shows clear effects of the discretized control law and the effect of the measurement updates in the Kalman filter. Nevertheless, the linear SMO is still able to approximate the true implemented $\Delta u_t$.

Unfortunately, as explained in detail in Chapter 7, the collected data was unsuitable to fit a non-linear vehicle model, thus could not be used to test the non-linear SMO.

**Final Conclusion** From the simulation and the experimental results, the conclusion can be drawn that it is indeed possible to reconstruct the anomalous data received through wireless communication. However, it is important to note that this conclusion is drawn based on the assumption that the vehicles can measure their relative velocity. Future research is necessary to investigate to what extend this assumption holds.

## 8-2 Discussion and Recommendations

This section discusses and reflects on the study as a whole. Furthermore, this thesis is concluded with the recommendations for future research.

### 8-2-1 Discussion

The discussion of the thesis is divided into three parts. First, the choice of IC is discussed after which an elaboration on the SMO is given, and finally, the experimental set-up is reviewed.

**Choice of Intersection Controller** The automation of intersections is a project that is still years away. Thus, which IC technique is eventually going to be implemented is not known. However, the choice of IC technique influences the type of data that is transmitted via wireless communication.

In VP enabled IC protocol, vehicles use the intended acceleration of the target vehicle to avoid collisions on the intersection. Thus, it is a natural choice to implement an observer for anomaly detection, since the observer can directly use the VP control law. In other IC protocols, this choice may be less applicable. For example, in optimization based IC an often proposed anti-collision constraint is to use the time of arrival. In this case, to verify the time of arrival, the observer design is less straight forward.

**Discussion on the Sliding Mode Observer** In order to use the SMO to reconstruct the unknown input, it is important that two requirements are met. As shown in Chapter 5, the so-called ranking condition is not met for the VP state-space. Therefore, a technique from [34] was applied to fit the ranking condition, which implies to estimate the state where the

unknown input enters in. The disadvantage with using this technique, is that the performance of the unknown input estimation depends on the accuracy of the extra state estimation.

Furthermore, in the case that measurement noise is present, the Kalman filter outperforms the linear SMO. This is mainly caused by the estimation of the third state component, using a robust exact differentiator. The robust exact differentiator introduces a delay in the estimation of the third state component. The SMO tries to compensate for this delay, which is visible in the unknown input Reconstruction in Figure 6-8.

**Discussion on the Experiments** To test the SMO in a more realistic setting, the VP protocol was applied in an experimental set-up. Due to the infrequent measurements of the vehicles, a Kalman filter was added to produce data in between the measurements. The Kalman filter uses the simple linear vehicle model to estimate the states of the system. However, since the measurements are so infrequent, the Kalman filter runs in open loop for a considerate amount of time between two measurements. Thus, at every measurement update, the estimated state is not matching the measurements which results in a jump in the state estimates.

Furthermore, to estimate the unknown input, the SMO was implemented in continuous-time. Thus, the SMO expects the collected data to behave like the continuous-time VP model. The VP model implements the continuous-time control law $\dot{u}$. However, since the control law is discretized, the control input implemented in the experiments does not behave like the model $\dot{u}$.

Both the measurement updates as well as the discrete control law cause a peaking structure in the reconstruction of the unknown input, as can be seen in Figure 7-8.

As also stated in the conclusion, the SMO depends on the important assumption that the vehicles can measure their relative velocity. However, the vehicles in the experimental set-up can only measure their own position and velocity. Thus, this important assumption could not be tested on the experimental set-up.

And finally, the communication protocol used in the experimental set-up is different from the actual Vehicular Ad-Hoc Network (VANET). In the protocol used in the experiments, the communication delays are present at different steps in the VP method than when using the VANET.

### 8-2-2 Recommendations

Considering the presented conclusions and the discussed items, several recommendations can be made for future research.

1. *Test the non-linear SMO on a more precise experimental set-up.* The largest limitation posed by the experimental set-up were the infrequent measurements. The infrequent measurements prevented the possibility to test the non-linear SMO on the experiments. Therefore, it is recommended to test the performance of the non-linear SMO on a more accurate experimental set-up. Using such a set-up simplifies the estimation of the non-linear vehicle model. Furthermore, a data set with more frequent measurements also omits the need for the Kalman filter to produce data withing the measurement points.

This allows for a better substantiated performance analysis of the SMO for anomalous data reconstruction.

2. *Investigate to what extend the measurement assumption holds.* The design of the SMO relies on the assumption that the vehicles can measure their relative velocity. Thus, it is recommended to investigate what the precision and range is on the measured relative velocity. Furthermore, the vehicles approach the intersection from distinct trajectories, thus it is likely that the line-of-sight might be blocked (e.g., by another vehicle). If the line-of-sight is blocked, the vehicles are not able to measure the relative velocity to each other. Thus, it is recommended to investigate alternatives in such scenarios.

3. *Discretize the SMO such that it can be implemented in real-time.* Transforming the SMO from continuous-time to discrete-time will improve the unknown input reconstruction in the case that the used control law is discretized.

4. *Implement the SMO in real-time* The accuracy of the unknown input estimation was analyzed by applying the SMO to the collected data from the experiments. Eventually, it is the aim to implement the SMO in real-time, thus should be tested in experiments as well.

5. *Further investigate alternative methods to improve the robustness of the SMO.* In Chapter 6, a robust exact differentiator was used to estimate the third state. The robust differentiator introduces a delay in the state estimation, which has a negative impact on the unknown input estimation. It is therefore recommended to explore methods to either replace the robust exact differentiator, or to make the SMO more robust.

6. *Investigate the effect of model uncertainties in the SMO.* The performance of the SMO depends on how accurate the model of both the target and host vehicle is. It is recommended to look further in the influence of parameter uncertainty in the reconstruction of the unknown input.

# Appendix A

# Simulink Model of the Sliding Mode Observer

This appendix shows the Simulink model of the linear Sliding Mode Observer (SMO), without the alteration needed for measurement noise.

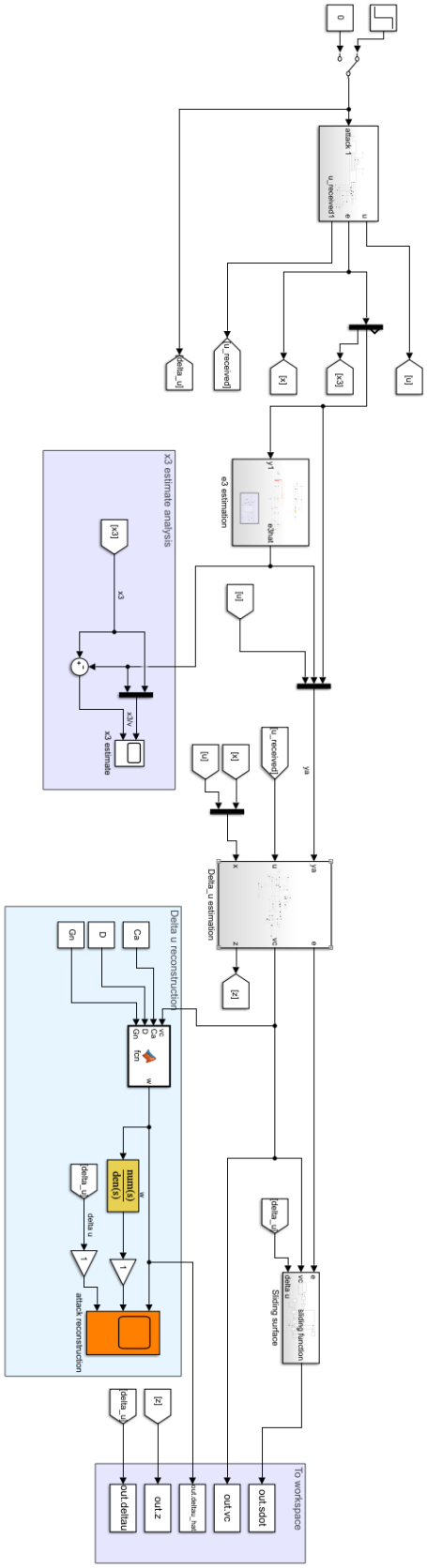



**Figure A-1:** Simulink model of Sliding Mode Observer

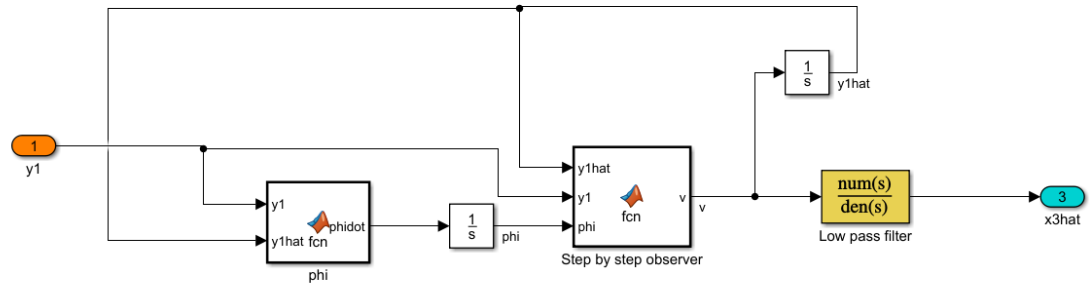



**Figure A-2:** Simulink model of $e_3$ estimation

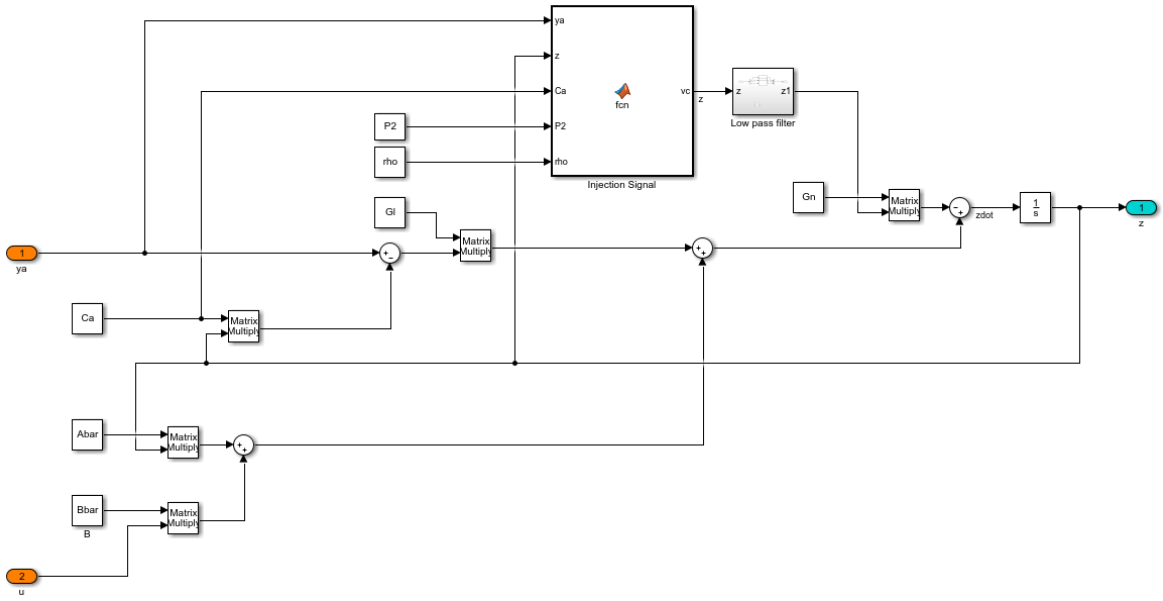



**Figure A-3:** Simulink model of the SMO

# Appendix B

# Stability of the Sliding Mode Observer

In this appendix, the stability proof of the linear Sliding Mode Observer (SMO) is provided. Combining the proofs given in Ref. [34] and Ref. [124] it can be guaranteed that the error dynamics of the SMO (5-7) is quadratically stable. The error dynamics are given by

$$\dot{\bar{\epsilon}} = (\bar{A} - G_l \bar{C}_a)\bar{\epsilon} + \bar{D}\Delta u_t - G_n v_c(\bar{C}_a \bar{\epsilon}). \tag{B-1}$$

The Lyapunov candidate function that guarantees quadratic stability of Equation (B-1) is $\mathcal{V}(\bar{\epsilon}) = \bar{\epsilon}\bar{P}\bar{\epsilon}$, where $\bar{P} > 0$. The matrix $\bar{P}$ has the following structure

$$\bar{P} = \begin{bmatrix} \bar{P}_1 & \bar{P}_1\bar{L} \\ \bar{L}^T\bar{P}_1 & \bar{P}_2 + \bar{L}^T\bar{P}_1\bar{L} \end{bmatrix} \geq 0, \tag{B-2}$$

where $\bar{P}_1 \in R^{(n-p)\times(n-p)}$ and $\bar{P}_2 \in R^{p\times p}$. Implementing Equation (B-1), the derivative of the Lyapunov function is

$$\dot{V} = \bar{\epsilon}^T(\bar{A}_0^T\bar{P} + \bar{P}\bar{A}_0)\bar{\epsilon} + 2\bar{\epsilon}^T\bar{P}G_n v - 2\bar{\epsilon}^T\bar{P}\bar{D}\Delta u_t, \tag{B-3}$$

where $\bar{A}_0 = \bar{A} - G_l\bar{C}_a$. The non-linear feedback matrix $G_n$ is defined as

$$G_n = \begin{bmatrix} -\bar{L}T^T \\ T^T \end{bmatrix} P_0^{-1}. \tag{B-4}$$

The matrix $T$ is part of the $\bar{C}_a$ matrix after the transformation (5-22). Designing $\bar{L}$ as $\bar{L} = \begin{pmatrix} L & 0 \end{pmatrix}$ with the dimensions $L \in R^{(n-p)\times(p-q)}$, a convenient structure for $\bar{P}G_n$ is obtained. The matrix $P_0$ is defined as

$$P_0 := T\bar{P}_2T^T. \tag{B-5}$$

Now using the convenient structure of $\bar{L}$ and the definitions for $\bar{P}$ (B-2), $G_n$ (B-4) and $\bar{C}_a$ (5-22), the following relation is obtained

$$\bar{P}\bar{G}_n = \begin{bmatrix} 0 \\ \bar{P}_2 T^T \end{bmatrix} P_0^{-1} = \bar{C}_a^T. \tag{B-6}$$

Furthermore, multiplying $\bar{P}$ with the unknown input distribution matrix $D$ again leads to a convenient form due to the structure of $\bar{L}$

$$\bar{P}\bar{D} = \begin{bmatrix} 0 \\ \bar{P}_2 D_2 \end{bmatrix} = \bar{C}_a^T P_0 \mathcal{D}_2, \tag{B-7}$$

where the matrix $\mathcal{D}_2$ is defined as

$$\mathcal{D}_2 := T D_2. \tag{B-8}$$

Next, implementing definitions (B-6), (B-7) and (B-8) in the derivative of the Lyapunov candidate (B-3) gives

$$\dot{V} = \bar{e}^T (\bar{A}_0^T \bar{P} + \bar{P}\bar{A}_0)\bar{e} + 2e_y^T v - 2e_y^T P_0 \mathcal{D}_2 \Delta u_t. \tag{B-9}$$

Implementing the definition of the injection signal (5-9) and the bound on the unknown input (5-10) leads to

$$\dot{V} \leq \bar{e}^T (\bar{A}_0^T \bar{P} + \bar{P}\bar{A}_0)\bar{e} - 2\rho \|P_0 \mathcal{D}_2\| \|e_y\| + 2\gamma \|P_0 \mathcal{D}_2\| \|e_y\|. \tag{B-10}$$

Finally, implementing the definition $\rho \geq \gamma + \mu$, we find

$$\dot{V} \leq \bar{e}^T (\bar{A}_0^T \bar{P} + \bar{P}\bar{A}_0)\bar{e} - 2\mu \|P_0 \mathcal{D}_2\| \|e_y\|. \tag{B-11}$$

By appropriately designing $G_l$ and $\bar{P}$, the result of $(\bar{A}_0^T \bar{P} + \bar{P}\bar{A}_0)$ can be made negative definite. Thus $\dot{V} \leq 0$, which completes the stability proof.

# Appendix C

# String stability of the Experimental Set-Up

It is important that the vehicles participating in a (Virtual) Platoon satisfy the string stability requirement. String stability implies that a disturbance in the velocity of one vehicle will not be amplified throughout the platoon. Network induced effects, such as the sampling time and the communication delay can cause the Virtual Platooning (VP) to become unstable. Therefore, the string stability of the experimental set-up is analyzed is this section, while the typical network effects are present. In order to analyze the string stability, the method from Öncü et al. [86] is slightly adjusted to fit the dynamics of the vehicles in the experimental set-up. Furthermore, to simplify the analysis, it is assumed that the delays in the system are constant. A VP is called string stable when the following equation holds

$$\left| \frac{\Delta_i(e^{jw})}{\Delta_{i-1}(e^{jw})} \right| \leq 1 \, \forall w, \, i = 1, ..., n, \tag{C-1}$$

where $\Delta_i$ is the $\mathcal{Z}$-transform of the velocity of vehicle $i$, and $n$ is the amount of vehicles present in the VP. This means that it is necessary that the velocity of the participating vehicles is present as a state in the system dynamics. Therefore, instead of using the state vector $x$ as defined in Equation 2-11, the state-space is rewritten such that the velocity $v$ is included as a state. The new state vector is equal to $x_i^T = \begin{bmatrix} e_i & v_i & a_i & u_{ff} \end{bmatrix}$. While implementing the Cruise Control law, the dynamics of the target vehicle, indicated by $x_1$, is given by

$$\begin{aligned} \dot{x}_1 &= A_{1,1}x_1 + A_{1,0}x_0 + B_{s,1}u_1, \\ A_{1,1} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{\tau} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \; B_{s,1} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\tau} \\ 0 \end{bmatrix}, \; A_{1,0} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \end{aligned} \tag{C-2}$$

where $u_r$ is the reference acceleration and $x_0$ is the state of a phantom vehicle, which implements the reference velocity and control input. The Cruise Control law is given by $u_1 = K_{1,0}x_0 + K_{1,1}x_1 + K_r u_r$, where $K_{1,0} = \begin{bmatrix}0 & 0 & 0 & 0\end{bmatrix}$, $K_{1,1} = \begin{bmatrix}-k_{cc} & 0 & 0 & 0\end{bmatrix}$ and $K_{1,r} = 1$. Next, the dynamics of the following vehicle $x_2$, also referred to as the host vehicle, is as follows

$$\dot{x}_2 = A_{2,2}x_2 + A_{2,1}x_0 + B_{s,2}u_2 + B_{c,2}u_{1,c}$$

$$A_{2,2} = \begin{bmatrix}0 & -1 & -h & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & -\frac{1}{\tau} & 0\\ 0 & 0 & 0 & -\frac{1}{h}\end{bmatrix},\ B_{s,2} = \begin{bmatrix}0\\ 0\\ \frac{1}{\tau}\\ 0\end{bmatrix},\ A_{2,1} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix},\ B_{c,2} = \begin{bmatrix}0\\ 0\\ 0\\ -\frac{1}{h}\end{bmatrix}, \tag{C-3}$$

where $u_{1,c}$ is the control input of the target vehicle, received through wireless communication. The control law given in Equation (3-1) is rewritten to fit the model (C-3). Thus $u_2 = K_{2,1}x_1 + K_{2,2}x_2 + K_r u_r$, where $K_{2,1} = \begin{bmatrix}0 & k_d & 0 & 0\end{bmatrix}$, $K_{1,1} = \begin{bmatrix}k_p & -k_d & -k_d h & 1\end{bmatrix}$ and $K_{2,r} = 0$.

Next, it is necessary to include the networked induced effects, such as communication delays, in the dynamics. In Figures 7-3 and 7-4 it is shown that the calculation of the needed control input is done remotely, thus both vehicles experience communication delay. The communication delay caused by the remote calculations is modelled as a constant actuator delay, captured by the following equation

$$\tilde{u}(t) = u(t - \theta_a). \tag{C-4}$$

The actuator delay $\theta_a$ is defined in the Laplace domain as $e^{-\theta_a s}$. A common technique to approximate this pure delay is by using a rational Padé approximation. The state-space that includes the actuator delay is equal to

$$\begin{aligned}\dot{\tilde{p}} &= A_p\tilde{p} + B_p u,\\ \tilde{u} &= C_p\tilde{p} + D_p u,\end{aligned} \tag{C-5}$$

where $\tilde{p} = \begin{bmatrix}p_1 & p_2 & \dots & p_\kappa\end{bmatrix}$ and $\kappa$ defines the order of the Padé approximation. Replacing $u_i$ in Equations (C-2) and (C-3) with the Padé approximation $\tilde{u}_i$, directly applies the actuator delay in the system dynamics. The extended state vector that contains the rational approximation of the actuator delay is given by $\tilde{x}^T = \begin{bmatrix}x^T & \tilde{p}_i^T\end{bmatrix}$. The extended state vector is further used for string stability analysis.

The next step in analyzing the string stability is to merge the dynamics of the participating vehicles, such that the complete VP dynamics are described. The merged dynamics of the VP is represented by

$$\dot{\bar{x}}_n = \underbrace{(\bar{A}_n + \bar{B}_{s,n}\bar{K}_n)}_{A^{ACC}}\bar{x}_n + \bar{B}_{c,n}\bar{u}_{n,w} + B_{s,n}\bar{K}_r u_r, \tag{C-6}$$

where $\bar{x}_n^T = \begin{bmatrix}\tilde{x}_0 & \tilde{x}_1 & \tilde{x}_2\end{bmatrix}$ and the system matrices are defined as

$$\bar{A}_n = \begin{bmatrix} \tilde{A}_0 & 0 & 0 \\ \tilde{A}_{1,0} & \tilde{A}_{1,1} & 0 \\ 0 & \tilde{A}_{2,1} & \tilde{A}_{2,2} \end{bmatrix}, \ \bar{B}_{s,n} = \begin{bmatrix} \tilde{B}_{s,0} & 0 & 0 \\ 0 & \tilde{B}_{s,1} & 0 \\ 0 & 0 & \tilde{B}_{s,2} \end{bmatrix}, \ \bar{K}_r = \begin{bmatrix} K_{0,r} \\ K_{1,r} \\ K_{2,r} \end{bmatrix}, \ \bar{B}_{c,n} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \tilde{B}_{c,2} & 0 \end{bmatrix}. \tag{C-7}$$

The total feedback matrix $\bar{K}_n$ that takes the new augmented state $\tilde{x}_n$ into account is

$$\bar{K}_n = \begin{bmatrix} K_{1,0} & 0 & K_{1,1} & 0 & 0 & 0 \\ 0 & 0 & K_{2,1} & 0 & K_{2,2} & 0 \end{bmatrix}. \tag{C-8}$$

Now that the complete VP dynamics are defined, the remaining network induced effects are implemented. The continuous-time model (C-6) is discretized with sampling time $t_k = kH$, where $H$ is the sampling size. Furthermore, the delay in receiving the control input from the target vehicle $u_{t,c}$ is modeled as $\theta = \theta^* + (l-1)H$, where $l \in \{1, 2, ...\}$ and $\theta^* \in [0, h]$. In the case that $l > 0$, the delay on receiving $u_{1,c}$ are larger than the sampling time $H$, and are referred to as large delays. Considering large delays, the state vector is extended with the prior control inputs $\xi^T = \begin{bmatrix} \bar{x}_n^T & \bar{u}_{n,k-1}^T & ... & \bar{u}_{n,k-l}^T \end{bmatrix}$. The discrete-time model containing the augmented state vector $\xi$ is obtained by exact differentiating [86], and is given by

$$\xi_{k+1} = \underbrace{\left(A_\xi(\theta, H) + B_\xi(\theta, H) K_\xi\right)}_{\bar{A}_\xi} \xi_k + \Gamma(H) u_{r,k}, \tag{C-9}$$

where $\bar{K}_n$ and $\bar{K}_r$ are extended with zeros to form $K_\xi$ and $K_{r,\xi}$. The state matrices are defined as

$$A_\xi(\theta, H) = \begin{bmatrix} e^{A^{ACC} H} & M_{l-1} & M_{l-2} & \dots & M_0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & I & 0 & \dots & 0 \\ \vdots & & \ddots & \dots & \\ 0 & \dots & 0 & I & 0 \end{bmatrix}, \ B_\xi(\theta, H) = \begin{bmatrix} M_l \\ I \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$\Gamma_r(H) = \begin{bmatrix} \int_0^H e^{A^{ACC} s} ds B_{s,n} \bar{K}_r \\ I \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \ M_j = (\theta, H) = \begin{cases} \int_{h-t_{j+t}}^{h-t_j} e^{A^{ACC} s} ds \bar{B}_{c,n}, & \text{if } 0 \leq j \leq 1, \\ 0, & \text{if } 1 < j \leq l, \end{cases} \tag{C-10}$$

where $t_0 = 0$, $t_1 = \theta^*$ and $t_2 = H$. Finally, using the state matrices defined in Equation (C-10) an analysis of the string stability can be performed. The authors from [86] note that the string stability Equation (C-1) can be rewritten into

$$\left| \frac{\Delta_i(e^{jw})}{\Delta_{i-1}(e^{jw})} \right| = \left| \Psi_{\Delta_i}(e^{jw}) \left( \Psi_{\Delta_{i-1}}(e^{jw}) \right)^{-1} \right|, \tag{C-11}$$

where the function $\Psi_{\Delta_i}(z)$ is the discrete-time transfer function from the acceleration profile $u_r$ to the velocity of the $i$-th vehicle (C-10),

$$\Psi_{\Delta_i}(z) = C_{\Delta_i} \left( zI - \bar{A}_\xi(\theta, H) \right)^{-1} \Gamma_r(H). \tag{C-12}$$

The string stability analysis was implemented in `Matlab` and varied over the parameters $H$, $\theta$ and $\theta_a$. From a small experiment on the Anki vehicles, shown in Chapter 7, it appears that the actuator delay $\theta_a$ is at most $\theta_a = 0.3\,s$. Therefore, the string stability was analyzed for $0 < \theta_a \leq 0.4$ and $0 < \theta \leq 0.9$. With a step size of $H = 0.1$, the VP is string stable for the whole range of $\theta_a$ and $\theta \leq 0.37\,s$. Thus, with a step size of $H = 0.1$, the VP is string stable within the estimated actuator delay of $\theta_a \leq 0.3\,s$ and the communication delay $\theta \leq 0.37\,s$. From Figure 7-3 it can be seen that the communication delay $\theta$ will be approximately equal to the actuator delay $\theta_a$. It is concluded that with $\theta \leq 0.37\,s$, the VP is string stable, provided that the actuator delay is indeed not higher than $0.3\,s$.

# Bibliography


[1] Z Abdollahi Biron, S Dey, and P Pisu. "Real-Time Detection and Estimation of Denial of Service Attack in Connected Vehicle Systems". In: *IEEE Trans. Intell. Transp. Syst.* 19.12 (Dec. 2018), pp. 3893–3902.

[2] 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)*. ieeexplore.ieee.org, Jan. 2017, pp. 157–162.

[3] I M Almomani, N Y Alkhalil, E M Ahmad, and R M Jodeh. "Ubiquitous GPS vehicle tracking and management system". In: *2011 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*. ieeexplore.ieee.org, Dec. 2011, pp. 1–6.

[4] 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". In: *IEEE Commun. Mag.* 53.6 (June 2015), pp. 126–132.

[5] *Anki Drive SDK Python wrapper*. Version 1.0. 2017.

[6] *Anki OVERDRIVE - Intelligent Racing Robot System: Anki*.

[7] Christopher Attard, Shane Elwart, Jeff Allen Greenberg, Rajit Johri, John P Joyce, Devinder Singh Kochhar, Douglas Scott Rhode, John Shutko, and Hongtei Eric Tseng. "Fault handling in an autonomous vehicle". 9406177. Aug. 2016.

[8] Rachid Attia, Rodolfo Orjuela, and Michel Basset. "Combined longitudinal and lateral control for automated vehicle guidance". In: *Veh. Syst. Dyn.* 52.2 (Feb. 2014), pp. 261–279.

[9] Reza Azimi, Gaurav Bhatia, Raj Rajkumar, and Priyantha Mudalige. *Intersection management using vehicular networks*. Tech. rep. SAE Technical Paper, 2012.

[10] Seyed Reza Azimi, Gaurav Bhatia, Ragunathan Raj Rajkumar, and Priyantha Mudalige. "Vehicular networks for collision avoidance at intersections". In: *SAE International Journal of Passenger Cars-Mechanical Systems* 4.2011-01-0573 (2011), pp. 406–416.

[11] G Bartolini, A Ferrara, and E Usai. "Chattering avoidance by second-order sliding mode control". In: *IEEE Trans. Automat. Contr.* 43.2 (Feb. 1998), pp. 241–246.

[12] G Bartolini, A Ferrara, E Usai, and V I Utkin. "On multi-input chattering-free second-order sliding mode control". In: *IEEE Trans. Automat. Contr.* 45.9 (Sept. 2000), pp. 1711–1717.

[13] Zoleikha Abdollahi Biron. "A resilient control approach to secure cyber physical systems (CPS) with an application on connected vehicles". PhD thesis. Clemson University, 2017.

[14] Zoleikha Abdollahi Biron, Satadru Dey, and Pierluigi Pisu. "Sensor fault diagnosis of connected vehicles under imperfect communication network". In: *ASME, Dynamic Systems and Control Conference*. Vol. 1. researchgate.net, 2016, V001T16A003.

[15] N Bißmeyer, S Mauthofer, K M Bayarou, and F Kargl. "Assessment of node trustworthiness in VANETs using data plausibility checks with particle filters". In: *2012 IEEE Vehicular Networking Conference (VNC)*. ieeexplore.ieee.org, Nov. 2012, pp. 78–85.

[16] Levente Buttyán, Tamás Holczer, and István Vajda. "On the Effectiveness of Changing Pseudonyms to Provide Location Privacy in VANETs". In: *Security and Privacy in Ad-hoc and Sensor Networks*. Springer Berlin Heidelberg, 2007, pp. 129–141.

[17] Luca Maria Castiglione, Paolo Falcone, Alberto Petrillo, Simon Pietro Romano, and Stefania Santini. "Cooperative Intersection Crossing over 5G". In: *arXiv preprint arXiv:1907.07643* (2019).

[18] Changsheng Li and M Elbuluk. "A robust sliding mode observer for permanent magnet synchronous motor drives". In: *IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02*. Vol. 2. ieeexplore.ieee.org, Nov. 2002, 1014–1019 vol.2.

[19] Jie Chen, Ron J Patton, and Hong-Yue Zhang. "Design of unknown input observers and robust fault detection filters". In: *Int. J. Control* 63.1 (Jan. 1996), pp. 85–105.

[20] A Colombo and D Del Vecchio. "Least Restrictive Supervisors for Intersection Collision Avoidance: A Scheduling Approach". In: *IEEE Trans. Automat. Contr.* 60.6 (June 2015), pp. 1515–1527.

[21] European Commission. "Traffic Safety Basic Facts on Junctions". In: *Directorate General for Transport* (June 2018).

[22] Sae On-Road Automated Vehicle Standards Committee et al. "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems". In: *SAE Standard J* 3016 (2014), pp. 1–16.

[23] Soodeh Dadras, Ryan M Gerdes, and Rajnikant Sharma. "Vehicular Platooning in an Adversarial Environment". In: *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*. ASIA CCS '15. Singapore, Republic of Singapore: ACM, 2015, pp. 167–178.

[24] S Dietzel, J Petit, G Heijenk, and F Kargl. "Graph-Based Metrics for Insider Attack Detection in VANET Multihop Data Dissemination Protocols". In: *IEEE Trans. Veh. Technol.* 62.4 (May 2013), pp. 1505–1518.

[25] John R Douceur. "The Sybil Attack". In: *Peer-to-Peer Systems*. Springer Berlin Heidelberg, 2002, pp. 251–260.

[26] K Dresner and P Stone. “Multiagent traffic management: An improved intersection control mechanism”. In: *Proceedings of the fourth international joint* (2005).

[27] Kurt Dresner and Peter Stone. “Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism”. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*. AAMAS ’04. New York, New York: IEEE Computer Society, 2004, pp. 530–537.

[28] William B Dunbar and Derek S Caveney. “Distributed receding horizon control of vehicle platoons: Stability and string stability”. In: *IEEE Transactions on Automatic Control* 57.3 (2011), pp. 620–633.

[29] Christopher Edwards and Sarah K Spurgeon. “On the development of discontinuous observers”. In: *International Journal of control* 59.5 (1994), pp. 1211–1229.

[30] Richard Gilles Engoulou, Martine Bellaıche, Samuel Pierre, and Alejandro Quintero. “VANET security surveys”. In: *Comput. Commun.* 44 (May 2014), pp. 1–13.

[31] Yong Feng, Fengling Han, and Xinghuo Yu. “Chattering free full-order sliding-mode control”. In: *Automatica* 50.4 (2014), pp. 1310–1314.

[32] R M G Ferrari and A M H Teixeira. “Detection and isolation of routing attacks through sensor watermarking”. In: *2017 American Control Conference (ACC)*. ieeexplore.ieee.org, May 2017, pp. 5436–5442.

[33] Riccardo M G Ferrari and André M H Teixeira. “Detection and isolation of replay attacks through sensor watermarking”. In: *IFAC-PapersOnLine* 50.1 (2017), pp. 7363–7368.

[34] T Floquet, C Edwards, and S K Spurgeon. “On sliding mode observers for systems with unknown inputs”. In: *Int. J. Adapt Control Signal Process.* Lecture Notes 21.8-9 (Oct. 2007), pp. 638–656.

[35] Leonid Fridman, Yuri Shtessel, Christopher Edwards, and Xing-Gang Yan. “Higher-order sliding-mode observer for state estimation and input reconstruction in nonlinear systems”. In: *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal* 18.4-5 (2008), pp. 399–412.

[36] M Gerlach and F Guttler. “Privacy in VANETs using Changing Pseudonyms - Ideal and Real”. In: *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*. ieeexplore.ieee.org, Apr. 2007, pp. 2521–2525.

[37] M Ghosh, A Varghese, A A Kherani, and A Gupta. “Distributed Misbehavior Detection in VANETs”. In: *2009 IEEE Wireless Communications and Networking Conference*. ieeexplore.ieee.org, Apr. 2009, pp. 1–6.

[38] Robert Bosch GmbH. “Mid-range radar sensor (MRR) for front and rear applications”. In: *Bosch Mobility Solutions* (2017).

[39] Philippe Golle, Dan Greene, and Jessica Staddon. “Detecting and Correcting Malicious Data in VANETs”. In: *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*. VANET ’04. Philadelphia, PA, USA: ACM, 2004, pp. 29–37.

[40] Mehmet Ali Guney and Ioannis A Raptis. *Scheduling-driven Motion Coordination of Autonomous Vehicles at a Multi-Lane Traffic Intersection*. 2018.

[41] M R Hafner, D Cunningham, L Caminiti, and D Del Vecchio. "Cooperative Collision Avoidance at Intersections: Algorithms and Experiments". In: *IEEE Trans. Intell. Transp. Syst.* 14.3 (Sept. 2013), pp. 1162–1175.

[42] Hamssa Hasrouny, Abed Ellatif Samhat, Carole Bassil, and Anis Laouiti. "VANet security challenges and solutions: A survey". In: *Vehicular Communications* 7 (Jan. 2017), pp. 7–20.

[43] Ammar Haydari and Yasin Yilmaz. *Real-Time Detection and Mitigation of DDoS Attacks in Intelligent Transportation Systems*. 2018.

[44] L He and W T Zhu. "Mitigating DoS attacks against signature-based authentication in VANETs". In: *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*. Vol. 3. ieeexplore.ieee.org, May 2012, pp. 261–265.

[45] R W van der Heijden, S Dietzel, T Leinmüller, and F Kargl. "Survey on Misbehavior Detection in Cooperative Intelligent Transportation Systems". In: *IEEE Communications Surveys Tutorials* 21.1 (2019), pp. 779–811.

[46] R van der Heijden, T Lukaseder, and F Kargl. "Analyzing attacks on cooperative adaptive cruise control (CACC)". In: *2017 IEEE Vehicular Networking Conference (VNC)*. ieeexplore.ieee.org, Nov. 2017, pp. 45–52.

[47] A Hesham, A Abdel-Hamid, and M A El-Nasr. "A dynamic key distribution protocol for PKI-based VANETs". In: *2011 IFIP Wireless Days (WD)*. ieeexplore.ieee.org, Oct. 2011, pp. 1–3.

[48] H C Hsiao, A Studer, C Chen, A Perrig, F Bai, et al. "Flooding-resilient broadcast authentication for vanets". In: *Proceedings of the 17th* (2011).

[49] R Hult, G R Campos, P Falcone, and H Wymeersch. "An approximate solution to the optimal coordination problem for autonomous vehicles at intersections". In: *2015 American Control Conference (ACC)*. ieeexplore.ieee.org, July 2015, pp. 763–768.

[50] R Hult, M Zanon, S Gros, and P Falcone. "Primal decomposition of the optimal coordination of vehicles at traffic intersections". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. ieeexplore.ieee.org, Dec. 2016, pp. 2567–2573.

[51] Robert Hult, Mario Zanon, Sebastien Gros, and Paolo Falcone. *Optimal Coordination of Automated Vehicles at Intersections: Theory and Experiments*. 2018.

[52] European Telecommunications Standards Institute. "Intelligent Transport Systems (ITS); ITS-G5 Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band". In: *ETSI EN 302 663 V1.3.0* (May 2019).

[53] Niloofar Jahanshahi and Riccardo M G Ferrari. "Attack detection and estimation in cooperative vehicles platoons: A sliding mode observer approach". In: *IFAC-PapersOnLine* 51.23 (2018), pp. 212–217.

[54] B Jiang, M Staroswiecki, and V Cocquempot. "Fault estimation in nonlinear uncertain systems using robust/sliding-mode observers". In: *IEE Proceedings - Control Theory and Applications* 151.1 (Jan. 2004), pp. 29–37.

[55] Q Jin, G Wu, K Boriboonsomsin, and M Barth. "Platoon-based multi-agent intersection management for connected vehicle". In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. ieeexplore.ieee.org, Oct. 2013, pp. 1462–1467.

[56] Boyoon Jung and Gaurav S Sukhatme. “Detecting moving objects using a single camera on a mobile robot in an outdoor environment”. In: *International conference on intelligent autonomous systems*. robotics.usc.edu, 2004, pp. 980–987.

[57] M A S Kamal, J Imura, T Hayakawa, A Ohata, and K Aihara. “A Vehicle-Intersection Coordination Scheme for Smooth Flows of Traffic Without Using Traffic Lights”. In: *IEEE Trans. Intell. Transp. Syst.* 16.3 (June 2015), pp. 1136–1147.

[58] M A S Kamal, J Imura, A Ohata, T Hayakawa, and K Aihara. “Coordination of automated vehicles at a traffic-lightless intersection”. In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. ieeexplore.ieee.org, Oct. 2013, pp. 922–927.

[59] A Katriniok, S Kojchev, E Lefeber, and H Nijmeijer. “Distributed Scenario Model Predictive Control for Driver Aided Intersection Crossing”. In: *2018 European Control Conference (ECC)*. June 2018, pp. 1746–1752.

[60] Alexander Katriniok, Peter Kleibaum, and Martina Joševski. “Distributed Model Predictive Control for Intersection Automation Using a Parallelized Optimization Approach”. In: *IFAC-PapersOnLine* 50.1 (July 2017), pp. 5940–5946.

[61] Twan Keijzer and Riccardo M G Ferrari. “A Sliding Mode Observer Approach for Attack Detection and Estimation in Autonomous Vehicle Platoons using Event Triggered Communication”. In: (Nov. 2019).

[62] Tiffany Hyun-Jin Kim, Ahren Studer, Rituik Dubey, Xin Zhang, Adrian Perrig, Fan Bai, Bhargav Bellur, and Aravind Iyer. “VANET Alert Endorsement Using Multi-source Filters”. In: *Proceedings of the Seventh ACM International Workshop on VehiculAr InterNETworking*. VANET ’10. Chicago, Illinois, USA: ACM, 2010, pp. 51–60.

[63] M Kneissl, A Molin, H Esen, and S Hirche. “A Feasible MPC-Based Negotiation Algorithm for Automated Intersection Crossing*”. In: *2018 European Control Conference (ECC)*. June 2018, pp. 1282–1288.

[64] K Kristinsson and G A Dumont. “System identification and control using genetic algorithms”. In: *IEEE Trans. Syst. Man Cybern.* 22.5 (Sept. 1992), pp. 1033–1046.

[65] A de La Fortelle. “Analysis of reservation algorithms for cooperative planning at intersections”. In: *13th International IEEE Conference on Intelligent Transportation Systems*. ieeexplore.ieee.org, Sept. 2010, pp. 445–449.

[66] V Lebastard, Y Aoustin, and F Plestan. “Step-by-step sliding mode observer for control of a walking biped robot by using only actuated variables measurement”. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. ieeexplore.ieee.org, Aug. 2005, pp. 559–564.

[67] Hoon Lee and Vadim I Utkin. “Chattering suppression methods in sliding mode control systems”. In: *Annu. Rev. Control* 31.2 (Jan. 2007), pp. 179–188.

[68] J Lee and B Park. “Development and Evaluation of a Cooperative Vehicle Intersection Control Algorithm Under the Connected Vehicles Environment”. In: *IEEE Trans. Intell. Transp. Syst.* 13.1 (Mar. 2012), pp. 81–90.

[69] Arie Levant. “Robust exact differentiation via sliding mode technique”. In: *Automatica* 34.3 (1998), pp. 379–384.

[70] M W Levin, H Fritz, and S D Boyles. “On Optimizing Reservation-Based Intersection Controls”. In: *IEEE Trans. Intell. Transp. Syst.* 18.3 (Mar. 2017), pp. 505–515.

[71] Michael W Levin, Stephen D Boyles, and Rahul Patel. “Paradoxes of reservation-based intersection controls in traffic networks”. In: *Transp. Res. Part A: Policy Pract.* 90 (Aug. 2016), pp. 14–25.

[72] F Li and P Xiong. “Practical Secure Communication for Integrating Wireless Sensor Networks Into the Internet of Things”. In: *IEEE Sens. J.* 13.10 (Oct. 2013), pp. 3677–3684.

[73] L Li and F Wang. “Cooperative Driving at Blind Crossings Using Intervehicle Communication”. In: *IEEE Trans. Veh. Technol.* 55.6 (Nov. 2006), pp. 1712–1724.

[74] B Liu, Y Zhong, and S Zhang. “Probabilistic Isolation of Malicious Vehicles in Pseudonym Changing VANETs”. In: *7th IEEE International Conference on Computer and Information Technology (CIT 2007)*. ieeexplore.ieee.org, Oct. 2007, pp. 967–972.

[75] Lennart Ljung. “System identification toolbox”. In: *The Matlab user’s guide* (1988).

[76] N Lo and H Tsai. “Illusion Attack on VANET Applications - A Message Plausibility Problem”. In: *2007 IEEE Globecom Workshops*. ieeexplore.ieee.org, Nov. 2007, pp. 1–8.

[77] *Long-range radar sensor (LRR), Surround sensor for radar-based driver assistance systems*. Robert Bosch GmbH. 2019.

[78] R Lu, X Lin, T H Luan, X Liang, and X Shen. “Pseudonym Changing at Social Spots: An Effective Strategy for Location Privacy in VANETs”. In: *IEEE Trans. Veh. Technol.* 61.1 (Jan. 2012), pp. 86–96.

[79] Mi Lv, Wenwu Yu, Yuezu Lv, Jinde Cao, and Wei Huang. “An integral sliding mode observer for CPS cyber security attack detection”. en. In: *Chaos* 29.4 (Apr. 2019), p. 043120.

[80] Alejandro Ivan Morales Medina, Nathan Van de Wouw, and Henk Nijmeijer. “Automation of a T-intersection using virtual platoons of cooperative autonomous vehicles”. In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE. 2015, pp. 1696–1701.

[81] N Meskin and K Khorasani. “Actuator Fault Detection and Isolation for a Network of Unmanned Vehicles”. In: *IEEE Trans. Automat. Contr.* 54.4 (Apr. 2009), pp. 835–840.

[82] *Mid-range radar sensor (MRR), Surround sensor for radar-based driver assistance systems*. Robert Bosch GmbH. 2019.

[83] Vicente Milanés, Steven E Shladover, John Spring, Christopher Nowakowski, Hiroshi Kawazoe, and Masahide Nakamura. “Cooperative adaptive cruise control in real traffic situations”. In: *IEEE Transactions on Intelligent Transportation Systems* 15.1 (2013), pp. 296–305.

[84] Yilin Mo and Bruno Sinopoli. “False data injection attacks in control systems”. In: *Preprints of the 1st workshop on Secure Control Systems*. pdfs.semanticscholar.org, 2010, pp. 1–6.

[85] A I Morales Medina, N van de Wouw, and H Nijmeijer. “Cooperative Intersection Control Based on Virtual Platooning”. In: *IEEE Trans. Intell. Transp. Syst.* 19.6 (June 2018), pp. 1727–1740.

[86] S Öncü, J Ploeg, N van de Wouw, and H Nijmeijer. "Cooperative Adaptive Cruise Control: Network-Aware Analysis of String Stability". In: *IEEE Trans. Intell. Transp. Syst.* 15.4 (Aug. 2014), pp. 1527–1537.

[87] Anki Overdrive. *Anki Drive SDK*. Version 0.1.0. Mar. 2016.

[88] Soyoung Park, Baber Aslam, Damla Turgut, and Cliff C Zou. "Defense against sybil attack in vehicular ad hoc network based on roadside unit support". In: *MILCOM 2009-2009 IEEE Military Communications Conference*. researchgate.net, 2009, pp. 1–7.

[89] S Parkinson, P Ward, K Wilson, and J Miller. "Cyber Threats Facing Autonomous and Connected Vehicles: Future Challenges". In: *IEEE Trans. Intell. Transp. Syst.* 18.11 (Nov. 2017), pp. 2898–2915.

[90] F Pasqualetti, F Dörfler, and F Bullo. "Attack Detection and Identification in Cyber-Physical Systems". In: *IEEE Trans. Automat. Contr.* 58.11 (Nov. 2013), pp. 2715–2729.

[91] J Petit and S E Shladover. "Potential Cyberattacks on Automated Vehicles". In: *IEEE Trans. Intell. Transp. Syst.* 16.2 (Apr. 2015), pp. 546–556.

[92] J Ploeg, B T M Scheepers, E van Nunen, N van de Wouw, and H Nijmeijer. "Design and experimental evaluation of cooperative adaptive cruise control". In: *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. ieeexplore.ieee.org, Oct. 2011, pp. 260–265.

[93] J Ploeg, N van de Wouw, and H Nijmeijer. "Lp String Stability of Cascaded Systems: Application to Vehicle Platooning". In: *IEEE Trans. Control Syst. Technol.* 22.2 (Mar. 2014), pp. 786–793.

[94] X Qian, J Gregoire, A de La Fortelle, and F Moutarde. "Decentralized model predictive control for smooth coordination of automated vehicles at intersection". In: *2015 European Control Conference (ECC)*. ieeexplore.ieee.org, July 2015, pp. 3452–3458.

[95] Yue Quan, Wen Chen, Zhihai Wu, and Li Peng. "Distributed fault detection and isolation for leader–follower multi-agent systems with disturbances using observer techniques". In: *Nonlinear Dyn.* 93.2 (July 2018), pp. 863–871.

[96] Maxim Raya and Jean-Pierre Hubaux. "Securing vehicular ad hoc networks". In: *Int. J. Inf. Comput. Secur.* 15.1 (2007), pp. 39–68.

[97] M Realpe, B X Vintimilla, and L Vlacic. "A Fault Tolerant Perception system for autonomous vehicles". In: *2016 35th Chinese Control Conference (CCC)*. ieeexplore.ieee.org, July 2016, pp. 6531–6536.

[98] S RoselinMary, M Maheshwari, and M Thamaraiselvan. "Early detection of DOS attacks in VANET using Attacked Packet Detection Algorithm (APDA)". In: *2013 International Conference on Information Communication and Embedded Systems (ICICES)*. ieeexplore.ieee.org, Feb. 2013, pp. 237–240.

[99] S Ruj, M A Cavenaghi, Z Huang, A Nayak, and I Stojmenovic. "On Data-Centric Misbehavior Detection in VANETs". In: *2011 IEEE Vehicular Technology Conference (VTC Fall)*. ieeexplore.ieee.org, Sept. 2011, pp. 1–5.

[100] Brent Schwarz. "LIDAR: Mapping the world in 3D". In: *Nat. Photonics* 4.7 (2010), p. 429.

[101] S Sheikholeslam and C A Desoer. "Longitudinal control of a platoon of vehicles with no communication of lead vehicle information: a system level study". In: *IEEE Trans. Veh. Technol.* 42.4 (Nov. 1993), pp. 546–554.

[102] Yuri Shtessel, Christopher Edwards, Leonid Fridman, and Arie Levant. *Sliding mode control and observation*. Springer, 2014.

[103] A. Singh. *An intro to Kalman Filters for Autonomous Vehicles*. 2018. URL: https://towardsdatascience.com/an-intro-to-kalman-filters-for-autonomous-vehicles-f43dd2e2004b.

[104] S So, P Sharma, and J Petit. "Integrating Plausibility Checks and Machine Learning for Misbehavior Detection in VANET". In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. ieeexplore.ieee.org, Dec. 2018, pp. 564–571.

[105] Center for Strategic and International Studies (CSIS). *Significant Cyber Incidents Since 2006*. May 2020.

[106] H Stübing, J Firl, and S A Huss. "A two-stage verification process for Car-to-X mobility data based on path prediction and probabilistic maneuver recognition". In: *2011 IEEE Vehicular Networking Conference (VNC)*. ieeexplore.ieee.org, Nov. 2011, pp. 17–24.

[107] I Studnia, V Nicomette, E Alata, Y Deswarte, M Kaâniche, and Y Laarouchi. "Survey on security threats and protection mechanisms in embedded automotive networks". In: *2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W)*. ieeexplore.ieee.org, June 2013, pp. 1–12.

[108] Saif Al-Sultan, Moath M Al-Doori, Ali H Al-Bayatti, and Hussien Zedan. "A comprehensive survey on vehicular Ad Hoc network". In: *Journal of Network and Computer Applications* 37 (Jan. 2014), pp. 380–392.

[109] M Sun, M Li, and R Gerdes. "A data trust framework for VANETs enabling false data detection and secure vehicle tracking". In: *2017 IEEE Conference on Communications and Network Security (CNS)*. ieeexplore.ieee.org, Oct. 2017, pp. 1–9.

[110] S Tangade, S S Manvi, and P Lorenz. "Decentralized and Scalable Privacy-Preserving Authentication Scheme in VANETs". In: *IEEE Trans. Veh. Technol.* 67.9 (Sept. 2018), pp. 8647–8655.

[111] A M H Teixeira and R M G Ferrari. "Detection of Sensor Data Injection Attacks with Multiplicative Watermarking". In: *2018 European Control Conference (ECC)*. ieeexplore.ieee.org, June 2018, pp. 338–343.

[112] *The difference between Ultrasonic and Radar level sensors*. https://www.apgsensors.com/about-us/blog/radar-and-ultrasonic-sensors. Accessed: july 2019. Automation Products Group, Inc, 2014.

[113] O Tonguz*, N Wisitpongphan*, F Bait, P Mudaliget, and V Sadekart. "Broadcasting in VANET". In: *2007 Mobile Networking for Vehicular Environments*. ieeexplore.ieee.org, May 2007, pp. 7–12.

[114] Mark VanMiddlesworth, Kurt Dresner, and Peter Stone. "Replacing the Stop Sign: Unmanaged Intersection Control for Autonomous Vehicles". In: *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3*. AAMAS '08. Estoril, Portugal: International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 1413–1416.

[115] Michel Verhaegen and Vincent Verdult. *Filtering and system identification: a least squares approach*. Cambridge university press, 2007.

[116] L Wang, G Liu, L Sun, and Y Lin. "An Effective Pseudonym Generating Scheme for Privacy and Anonymity in VANETs". In: *2016 International Conference on Information System and Artificial Intelligence (ISAI)*. ieeexplore.ieee.org, June 2016, pp. 267–270.

[117] A Wasef, R Lu, X Lin, and X Shen. "Complementing public key infrastructure to secure vehicular ad hoc networks [Security and Privacy in Emerging Wireless Networks]". In: *IEEE Wirel. Commun.* 17.5 (Oct. 2010), pp. 22–28.

[118] S Weerakkody, X Liu, S H Son, and B Sinopoli. "A graph theoretic characterization of perfect attackability and detection in Distributed Control Systems". In: *2016 American Control Conference (ACC)*. ieeexplore.ieee.org, July 2016, pp. 1171–1178.

[119] A D Wood and J A Stankovic. "Denial of service in sensor networks". In: *Computer* 35.10 (Oct. 2002), pp. 54–62.

[120] Lingyun Xiao and Feng Gao. "Practical string stability of platoon of adaptive cruise control vehicles". In: *IEEE Transactions on intelligent transportation systems* 12.4 (2011), pp. 1184–1194.

[121] Xing-Gang Yan and C Edwards. "Robust sliding mode observer-based actuator fault detection and isolation for a class of nonlinear systems". In: *Proceedings of the 44th IEEE Conference on Decision and Control*. ieeexplore.ieee.org, Dec. 2005, pp. 987–992.

[122] Chen Yan, Wenyuan Xu, and Jianhao Liu. "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle". In: *DEF CON* 24 (2016).

[123] Gongjun Yan, Stephan Olariu, and Michele C Weigle. "Providing VANET security through active position detection". In: *Comput. Commun.* 31.12 (July 2008), pp. 2883–2897.

[124] Xinghuo Yu and Jian-Xin Xu. *Variable structure systems: towards the 21st century*. Vol. 274. Springer Science & Business Media, 2002.

[125] M Zanon, R Hult, S Gros, and P Falcone. "Experimental Validation of Distributed Optimal Vehicle Coordination". In: *2018 European Control Conference (ECC)*. ieeexplore.ieee.org, June 2018, pp. 1511–1516.

[126] Mario Zanon, Sébastien Gros, Henk Wymeersch, and Paolo Falcone. "An asynchronous algorithm for optimal vehicle coordination at traffic intersections". In: *IFAC-PapersOnLine* 50.1 (2017), pp. 12008–12014.

[127] C Zhang, X Lin, R Lu, P Ho, and X Shen. "An Efficient Message Authentication Scheme for Vehicular Communications". In: *IEEE Trans. Veh. Technol.* 57.6 (Nov. 2008), pp. 3357–3368.

[128] J R Zhang, S J Xu, and A Rachid. "Robust sliding mode observer for automatic steering of vehicles". In: *ITSC2000. 2000 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.00TH8493)*. ieeexplore.ieee.org, Oct. 2000, pp. 89–94.

[129] K Zhao and L Ge. "A Survey on the Internet of Things Security". In: *2013 Ninth International Conference on Computational Intelligence and Security*. ieeexplore.ieee.org, Dec. 2013, pp. 663–667.

[130] Feng Zhu and Satish V Ukkusuri. "A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment". In: *Transp. Res. Part C: Emerg. Technol.* 55 (June 2015), pp. 363–378.

[131] I H Zohdy, R K Kamalanathsharma, and H Rakha. "Intersection management for autonomous vehicles using iCACC". In: *2012 15th International IEEE Conference on Intelligent Transportation Systems*. ieeexplore.ieee.org, Sept. 2012, pp. 1109–1114.

[132] Ismail H Zohdy and Hesham A Rakha. *Intersection Management via Vehicle Connectivity: The Intersection Cooperative Adaptive Cruise Control System Concept*. 2016.