Development of new advanced estimation algorithms for improving the resilience of the autonomous and connected vehicle

PhD Thesis

Présentée et soutenue publiquement le xx mois 202x pour l'obtention du titre de

par

Quang Huy NGUYEN

Sous la direction de

Ali ZEMOUCHE
Hugh Rafaralahy
Madjid HADDAD

Composition du jury

Angelo Alessandri — *Rapporteur*
Professeur des Universités, University of Genoa

Naïma Aït Oufroukh Mammar — *Rapporteur*
Maitre de Conférences - HDR, University of Paris Saclay

Madiha Nadri-Wolf — *Examinatrice*
Maitre de Conférences, Université de Lyon

Vicenç Puig — *Examinateur*
Professeur des Universités, Polytechnic University of Catalonia

Taous Meriem Laleg Kirati — *Examinatrice*
Directeur de Recherche, Inria

Acknowledgements

I would like to express my sincere gratitude to the many individuals who have supported me throughout my doctoral journey.

First and foremost, I extend my deepest appreciation to my supervisors, Prof. Ali Zemouche, Prof. Hugues Rafaralahy, and Dr. Madjid Haddad, for granting me the opportunity to pursue this research. I am profoundly grateful for their continuous guidance, encouragement, and trust. Their insightful advice, valuable discussions, and unwavering support both scientific and personalhave been essential to the completion of this dissertation. Without their mentorship, this work would not have reached its present form.

I am also extremely grateful to my colleagues Hasni, Shengya, Echrak, Hichem, Oussama, Sesabil, Tansuh, Siyu, Viet, Melissa, and Li Qi for their cooperation, constructive feedback, and camaraderie throughout my PhD studies. From my first day in the laboratory to my first scientific publication, they were always willing to offer help. I particularly appreciate their patience and understanding when I took on my first leadership role during a team project. Despite my limited experience, they supported me with tolerance and encouragement rather than faulting my mistakes, allowing me to grow both professionally and personally.

Similarly, during my first teaching experiences, I faced significant challenges overcoming shyness and the difficulty of teaching in French, which is not my native language. The continuous support of my colleagues played a crucial role in helping me surmount these obstacles and build my confidence.

Last but not least, I express my heartfelt gratitude to my familymy uncle and aunt, my parents, and my girlfriendfor their unconditional love, patience, and support. I am especially thankful to my uncle and aunt, who have provided unwavering financial and spiritual support throughout my entire academic journey. Their encouragement and sacrifices have been a constant source of motivation, contributing significantly to my pursuit of higher academic achievements.

Résumé

La fiabilité et la sécurité des véhicules connectés et autonomes (VCA) dépendent de manière critique de la disponibilité en temps réel d'informations précises sur l'état dynamique. Cependant, les contraintes économiques limitent souvent l'utilisation de capteurs haut de gamme. De plus, les VCA sont vulnérables aux défauts de capteurs et aux cyberattaques ciblant les communications inter-véhicules (V2V/V2I). Pour relever ces défis, cette thèse propose un cadre de « capteur logiciel » résilient basé sur la redondance analytique.

Cette recherche développe une méthodologie de modélisation hybride intégrant des équations physiques et des réseaux de neurones neuro-adaptatifs. Cette approche permet une représentation précise des paramètres complexes et variables dans le temps ainsi que des dynamiques non modélisées. S'appuyant sur ces modèles, des algorithmes d'estimation non linéaires avancés observateurs à entrées inconnues (UIO) et observateurs neuro-adaptatifs sont développés pour garantir la résilience du système.

L'efficacité des stratégies proposées est démontrée à travers trois axes principaux :

- **CACC Cyber-sécurisé :** Des architectures d'observateurs sont conçues pour détecter et atténuer une large gamme de cybermenaces, incluant l'injection de fausses données (FDI), le déni de service (DoS), ainsi que les retards et pertes de paquets. Ces observateurs permettent la reconstruction des signaux d'attaque, assurant un contrôle longitudinal robuste.

- **Peloton et Suivi de Véhicules :** Un cadre d'estimation distribué est proposé pour gérer des modèles dynamiques non linéaires. Couplé à un mécanisme de gestion de la confiance, il permet de filtrer les données non fiables et d'assurer la cohésion du peloton lors de manœuvres complexes.

- **Estimation des Forces et Dynamique du Véhicule :** L'application des observateurs hybrides neuronaux est étendue à l'estimation de forces inconnues et de dynamiques complexes, adressant des problèmes critiques tels que le risque de renversement, où la connaissance précise des forces externes est essentielle pour la sécurité.

Mots clés : Estimation distribuée, observateur neuronal, Cyberattaque, Gestion de la confiance

Abstract

The reliability and safety of Connected and Autonomous Vehicles (CAVs) critically depend on the real-time availability of precise dynamic state information. However, economic constraints often limit the use of high-end sensors. Furthermore, CAVs are vulnerable to sensor faults and cyber-attacks targeting inter-vehicle communications (V2V/V2I). To address these challenges, this thesis proposes a resilient "soft sensor" framework based on analytical redundancy.

This research develops a hybrid modeling methodology integrating physical equations with neuro-adaptive neural networks. This approach enables the accurate representation of complex, time-varying parameters and unmodeled dynamics. Building on these models, advanced nonlinear estimation algorithms: Unknown Input Observers (UIO) and neuro-adaptive observers are developed to ensure system resilience.

The effectiveness of the proposed strategies is demonstrated through three main axes:

- **Cyber-Secure CACC:** Observer architectures are designed to detect and mitigate a wide range of cyber-threats, including False Data Injection (FDI), Denial of Service (DoS), as well as delays and packet losses. These observers enable the reconstruction of attack signals, ensuring robust longitudinal control.

- **Platooning and Vehicle Tracking:** A distributed estimation framework is proposed to handle nonlinear dynamic models. Coupled with a trust management mechanism, it allows filtering out unreliable data and ensuring platoon cohesion during complex maneuvers.

- **Vehicle Dynamics and Force Estimation:** The application of hybrid neural observers is extended to the estimation of unknown forces and complex dynamics, addressing critical issues such as rollover risk, where precise knowledge of external forces is essential for safety.

Keywords : State Estimation, Neural Observer, Cyber-attack, Trust management,

Table of Contents

List of Figures

List of Tables

Chapter 1

General Introduction

Objectives

This chapter lays the foundation for the thesis by outlining the research context, challenges, and objectives in resilient state estimation for connected autonomous vehicles. It motivates the need for hybrid and cooperative observer designs, summarizes the main contributions, and presents the organization of the manuscript.

Contents

1 Historical Background of Estimation Theory and Observer Design

State observers have been a cornerstone of control theory since the 1960s. The classical Luenberger observer, introduced by Luenberger (1964/1966), uses available inputs and outputs to reconstruct the internal state of a plant and asymptotically drive the estimate toward the true state. In parallel, Kalman's work on the Kalman filter (1960) established the foundations of stochastic state estimation, widely adopted for navigation and sensor fusion.

As control applications grew in complexity (e.g., aerospace and automotive systems), observer theory expanded to cope with unknown inputs, such as disturbances or unmeasured actuator effects. Unknown Input Observers (UIOs) were developed to estimate states despite these signals, under algebraic design conditions (often expressed as rank/matching constraints), and were applied extensively to fault detection and isolation. Representative contributions include the UIO-based fault-diagnosis frameworks of Hou and Müller (1992) and Chen and Patton (1999). In the same period, sliding-mode observers emerged as a robust alternative: by injecting discontinuous corrective terms, they can enforce finite-time error convergence and, in sliding motion, reduce sensitivity to certain matched uncertainties, enabling reconstruction of faults or disturbances via the so-called equivalent input.

Throughout the 1990s and 2000s, a broad family of robust and adaptive designs—including H_∞, high-gain, and adaptive observers—were proposed to maintain estimation performance under modeling errors, noise, and component faults. In automotive systems, these observers and filters became core building blocks, from early ECUs using Kalman filtering to multi-sensor schemes estimating lateral dynamics (e.g., slip angle or roll angle), wheel slip for anti-lock braking, and difficult-to-measure quantities such as road bank angle or tire forces modeled as disturbances. This historical evolution, from classical linear observers to robust and unknown-input extensions, sets the stage for today's resilient estimation frameworks.

2 Recent Advances in Hybrid and Resilient Estimation

Contemporary research in estimation builds on this legacy, addressing new challenges posed by connected and autonomous vehicles. One major trend is the development of hybrid observer architectures that combine model-based and data-driven elements. Purely data-driven estimators (e.g. deep neural network filters) can achieve high accuracy by learning complex patterns, but they lack transparency and are difficult to certify for safety-critical use. Conversely, purely model-based observers provide analytical guarantees of stability/robustness but may become inaccurate if the model is incomplete or if classical existence conditions (like observability rank or matching conditions) are violated. To overcome these trade-offs, hybrid approaches integrate physics-informed observer design with learning mechanisms. For example, a nominal observer (such as a Luenberger or Kalman filter or an Unknown Input Observer) is used to ensure basic robustness, while a learning module (e.g. a neural network or adaptive element) concurrently compensates for modeling errors or time-varying parameters. Recent studies have shown the promise of this approach. Jeon et al. (2024) develop a neuro-adaptive state observer where a neural network estimates unmodeled nonlinear dynamics; the design guarantees exponential stability of both the state estimation error and the neural parameter error, with even an H_∞ performance bound when the learned

approximation is imperfect. They further propose a switching hybrid estimator that alternates between learning (system identification) mode and a conventional observer mode, enabling online parameter updates only when sufficient excitation is present. This illustrates how blending learning with traditional observers can yield resilient estimators that adapt to unknown dynamics while preserving theoretical guarantees.

Another area of rapid progress is resilient state estimation under cyber-attacks and faults. As vehicles become connected, the estimation algorithms must contend with adversarial inputs (malicious sensor data or spoofed communication) in addition to natural disturbances. Classical robust observers treated unknown disturbances as benign inputs, but modern resilient estimation explicitly models intelligent attacks that may seek to deceive the estimator. A common strategy is to treat attacks as additional unknown inputs or to design observers that are robust to outliers. Unknown Input Observers have been revisited in this context: for instance, Jeon et al. (2020) demonstrated simultaneous cyber-attack detection and sensor health monitoring in a connected vehicle platoon using a bank of observers. In that work, deviations in an observer's residual signaled either a malicious data injection or a sensor fault, allowing the system to detect and isolate compromised signals. Sliding mode techniques have also been adapted for resilience – their intrinsic robustness to matched uncertainties is naturally suited to detecting attacks that enter through actuators or certain sensors. Recent studies integrate high-order sliding mode differentiators to estimate and reconstruct attack signals in real time. For example, Sadki et al. (2025) use an adaptive higher-order sliding differentiator as part of a cyberattack estimation scheme on a connected vehicle, achieving finite-time convergence of the attack estimate despite measurement noise. Meanwhile, researchers have relaxed the stringent conditions required by earlier unknown-input methods. The classical observer matching condition (which demands that each unknown input affect only certain state subspaces) severely limited applicability, since many practical systems – especially networked vehicles – do not satisfy this constraint. Advanced designs now employ geometric and optimization-based approaches to bypass these limitations. Floquet et al. (2007) and Kalsi et al. (2010) introduced strategies to augment the system outputs (via auxiliary high-gain filters or differentiators) such that a sliding-mode UIO can be constructed even when the original rank condition does not hold. More recently, Zhao et al. (2025) developed a distributed unknown input observer that leverages a joint condition across multiple vehicles in a platoon, instead of per-vehicle conditions, thereby allowing cooperative estimation where each vehicle's observer overcomes local unobservability by fusing information from neighbors. Such cooperative estimation approaches are especially relevant to connected automated driving, as they enable an entire platoon or fleet of vehicles to collectively estimate critical states and disturbance/attack inputs that no single vehicle could estimate in isolation.

In tandem with observer innovations, the community has explored data-driven estimation enhancements. Learning-based observers encompass methods like neural network observers, Gaussian process filters, and physics-informed neural networks (PINNs) embedded into observers. These techniques aim to improve estimation accuracy for highly nonlinear or partially known systems by learning from data. Crucially, they are often designed in a closed-loop manner with the system model: rather than replacing the model entirely, they correct or augment it. For example, neural network function approximators have been used to estimate vehicle tire-road friction or unknown aerodynamic forces, augmenting a nominal vehicle model observer. By training these networks online (adaptive) or offline, the observer can grad-

ually "learn" the unmodeled dynamics while the model-based portion ensures stability bounds. This hybrid learning paradigm is visible in many recent works on autonomous vehicles, often under names like neural observers, adaptive high-gain observers, or observer-based learning. Across the board, the thrust of current research is clear: marry the reliability of control-theoretic observers with the flexibility of machine learning. The result is estimation algorithms that are more resilient to both unexpected physical disturbances and adversarial attacks, which is vital as vehicles operate in open and uncertain environments.

2.1 Industrial relevance and deployment challenges

Bridging these advanced techniques from research to industry is a non-trivial task. Automotive systems are a prime example of cyber-physical systems where estimation algorithms must satisfy real-world constraints beyond theoretical performance. One major consideration is functional safety: industry standards (e.g. ISO 26262) demand that any component affecting vehicle control be robustly verified and fail-safe. This means an estimator must not only perform well nominally, but also handle worst-case scenarios (sensor failures, extreme environmental conditions, communication dropouts) without causing unsafe behavior. Model-based observers have an advantage here – their behavior under faults can often be bounded or analyzed using control theory. For instance, a well-designed H_∞ observer or interval observer can guarantee that estimation errors remain within set bounds for any disturbance within a specified energy or interval limit. In contrast, purely data-driven or black-box learning approaches raise concerns because proving their correctness or stability in all cases is difficult. Industry engineers thus tend to favor algorithms that come with analytic guarantees or at least are interpretable. This conservatism has motivated the hybrid approaches discussed earlier, which allow incorporation of machine learning without sacrificing the rigor of classical observers.

Another challenge is real-time computational constraints. Automotive electronic control units (ECUs) and networks have limited processing power and strict timing deadlines (often on the order of 1–10 ms for low-level control loops). Estimation algorithms with heavy computation (e.g. large neural networks or complex optimization-based observers) might be infeasible to run on current in-vehicle hardware. This drives the need for efficient implementations or hardware acceleration. It is not uncommon for advanced estimation algorithms to be prototyped in MATLAB/Simulink on a PC and then have to be simplified or optimized (e.g. using fixed-point arithmetic or reducing neural network size) for an actual vehicle ECU. The gap between academic prototypes and automotive-qualified software can be significant. Additionally, any algorithm intended for market deployment faces extensive testing: billions of miles of simulations and road tests are needed to validate autonomous vehicle software under diverse conditions. Estimation errors that are negligible on average could prove catastrophic in rare edge cases, so engineers must identify and mitigate these through scenario testing.

Environmental uncertainty presents further difficulties. Autonomous and connected vehicles operate in dynamic, unstructured environments where sensor data can degrade (heavy rain or fog affecting cameras and LiDAR, for example) and where the vehicle's dynamics can change (tire friction changes with road surface, load variations, etc.). Estimation algorithms must be robust to these variations or adaptive enough to track them. This is closely tied to resilience: an estimator should ideally detect when sensors are providing poor data (whether due to faults, attacks, or harsh conditions) and adjust weightings or

switch to fallback sensors. Industrial systems often include redundancy (e.g. multiple sensors measuring the same quantity) so that observer-based fault detection can exclude a bad sensor reading and use an alternative. For connected vehicles, redundancy can even be virtual – if a vehicle loses its own sensor, it might use a neighbor's information via V2V communication. However, this introduces the vulnerability of connectivity: as mentioned earlier, V2V signals may arrive delayed, be missing, or be maliciously corrupted. The estimator and the overall control system must be designed to maintain safety despite these communication issues. For example, cooperative Adaptive Cruise Control (CACC) algorithms typically degrade gracefully to normal ACC if V2V data is lost or untrustworthy, relying only on local radar sensing. This graceful degradation requires observers to quickly detect anomalies in received data (e.g. a sudden improbable drop in a neighbor's reported speed might indicate a spoofed message) and isolate or reject that data.

From an industrial standpoint, resilience against cyber-attacks is increasingly recognized as a necessary feature of autonomous vehicle software. High-profile demonstrations of car hacking have led to efforts in securing communications and ensuring observer-based monitors can promptly detect intrusions. Estimation algorithms that can reconstruct an attack signal (like the false acceleration command in a platoon) in real time are valuable for triggering mitigation strategies (such as switching to a safe mode). In this thesis, for instance, one objective is to estimate false data injection attacks on V2V links as unknown inputs and remove their effect. This kind of capability is likely to be incorporated in future vehicle intrusion detection systems (IDS) and resilient controllers.

Finally, it is important to mention the role of high-fidelity simulation and real-world testing platforms in bridging theory to practice. Researchers and engineers increasingly use platforms like CARLA (an open-source vehicle simulator) and industry tools like dSPACE/Simulink for validation of estimation algorithms in realistic scenarios. For example, in Chapter 2 of this thesis, a resilient observer for a CACC platoon under attack was tested in the CARLA simulator. The scenario involved a lead vehicle broadcasting false acceleration data to its followers; using CARLA allowed the inclusion of realistic vehicle dynamics and sensor noise. Results showed that the proposed observer could still detect and estimate the attack, though with a convergence delay of a few seconds and some noise-induced estimation error. Such studies underscore practical issues (estimator convergence time, noise sensitivity, model discrepancies) that might not be evident in pure theory. By incorporating realistic benchmarks and case studies, the development of observers can be guided to address the gaps between an idealized design and a deployable solution. In summary, while advanced estimation algorithms hold great promise for improving autonomy and safety, their industrial adoption hinges on meeting strict requirements for real-time performance, reliability, and certifiability in the unpredictable conditions of the real world.

3 Literature Review Summary

Before proceeding to the detailed contributions in Chapters 2–4, we summarize here the key themes from the literature that underpin this thesis. A comprehensive review was conducted on state-of-the-art observers and estimation methods relevant to resilient autonomous vehicles, spanning both classical techniques and emerging hybrid approaches.

Unknown Input and Robust Observers: Given the thesis focus on estimating states under disturbances

and attacks (modeled as unknown inputs), Unknown Input Observers (UIOs) are a central tool. Classical UIO designs for linear systems require that the direct feedthrough of unknown inputs satisfies certain algebraic conditions (commonly, an input-to-output rank condition). This ensures the observer's error dynamics can be decoupled from unknown inputs. Early works by Hou, Müller, Chen, Patton, and others formalized these conditions and developed UIOs for fault detection in the 1990s. However, these conditions are often restrictive in practice – many systems do not naturally satisfy them. To address this, researchers proposed several extensions. One approach is the use of functional observers or reduced-order observers that estimate only a function of the state (e.g. the fault itself) rather than the full state, relaxing design constraints. Another powerful approach is the incorporation of sliding-mode or high-gain techniques: Kalsi et al. (2010) showed that by adding a high-gain differentiator to generate auxiliary outputs, one can construct a sliding-mode observer even when the standard matching condition fails. Similarly, Floquet & Barbot (2006) employed a second-order sliding mode observer to recover unknown inputs in systems not meeting UIO requirements. These methods effectively extend the class of systems for which unknown inputs can be estimated, by either transforming the system or using robust numerical differentiation to counteract unknown influences. In the realm of nonlinear systems, robust UIO theory has also advanced. For example, Zemouche & Boutayeb (2009) developed a nonlinear observer with an H_∞ performance level that can recover unknown inputs in a synchronization context. Hassan et al. (2013) designed UIOs for nonlinear time-delay systems, ensuring robustness despite the presence of delayed unknown inputs. These works and others form a rich foundation for designing observers that maintain performance under model uncertainties, disturbances, and faults – a foundation upon which our proposed methods build.

Sliding Mode and High-Gain Observers: Sliding mode observers merit special attention in the literature due to the popularity in safety-critical applications. Edwards and Spurgeon's seminal work (1998) laid out a framework for sliding-mode observers that can exactly decouple matched disturbances. The appeal is that, once in sliding motion, the estimation error is governed by a reduced-order dynamics independent of the unknown input, yielding inherent robustness. Numerous studies have applied sliding-mode observers to vehicle systems, leveraging their finite-time convergence and robustness to parameter variations. High-order sliding modes (e.g. using super-twisting algorithms) were later introduced to improve estimation accuracy and chattering issues. As noted, Floquet et al. (2007) combined high-order sliding mode differentiators with UIO design to handle unmatched unknown inputs. In an automotive context, sliding-mode observers and differentiators have been used for estimating variables like tire forces and detecting faults/attacks. Recent literature includes applications of adaptive sliding observers to detect sensor attacks in CACC platoons, where the observer provides a real-time estimate of the attack signal. High-gain observers, while not employing discontinuous control, similarly provide strong convergence guarantees by trading off sensitivity to high-frequency noise. Meng et al. (2025) proposed a distributed high-gain observer for a network of autonomous vehicles, demonstrating how each vehicle can observe not only its own state but also a neighbor's state with the help of fast error dynamics and V2V communication. This method falls under the broader category of distributed observers.

Distributed and Cooperative Estimation: The cooperative estimation paradigm is increasingly prominent in the literature due to the rise of connected vehicles and multi-agent robotics. Instead of designing an observer for a single system in isolation, the idea is to have a network of observers (one per vehicle/agent)

that share information to improve overall estimation quality. A key benefit is tackling unobservable modes or unknown inputs that no single agent could estimate alone. Recent theoretical contributions, such as the distributed UIO by Zhao et al. (2025), show that collaborative unknown input estimation is possible in vehicle platoons: by relaxing per-vehicle observer conditions and enforcing a joint condition across the platoon, the unknown inputs (e.g. a driving disturbance or an attack affecting one vehicle) become observable to the network as a whole. Similarly, consensus-based observers and diffusion filters have been studied for sensor networks and CACC systems, where each vehicle fuses its local sensor data with neighbors' data. Meng et al. (2025) demonstrate that a distributed high-gain observer can enhance string stability by quickly propagating state estimates along a platoon. Cooperative estimation also appears in the context of fault tolerance: if one vehicle's sensor fails, neighboring vehicles can help estimate that vehicle's state via inter-vehicle communication (assuming the network remains secure). The literature highlights both the potential and challenges of distributed observers – while they can significantly improve estimation under connectivity, they must be designed to handle communication delays, packet losses, and potential spoofing. Techniques like event-triggered update laws, resilient consensus (ignoring extreme outlier nodes), and decentralized attack detection are being developed to bolster cooperative observers.

Learning-Based and Adaptive Observers: In recent years, there is a surge of interest in incorporating machine learning into observers, as noted in various survey articles and emerging research. These learning-based observers encompass approaches where either part of the system model is learned from data, or the observer gains are adjusted via learning algorithms. One stream of work involves neural network observers, where a neural net is trained to emulate an inverse model or to directly output state estimates from raw sensor data. However, more relevant to control literature are neuro-adaptive observers that maintain a clear separation between a physics-based model and a learned uncertainty model. An example (already cited) is the adaptive neural observer by Jeon et al. (2024), which treats the unknown nonlinear dynamics as a function approximator tuned online with Lyapunov stability guarantees. Importantly, their design yields convergence of both the state error and the neural network weights, illustrating that learning can be done in a stable closed-loop manner. Other adaptive observer techniques include extremum-seeking observers (which adjust parameters to minimize output error), or dual-estimation approaches where a parameter estimator (which could be learning-based) runs alongside the state estimator. In the context of autonomous vehicles, such adaptive schemes have been applied to things like battery state-of-charge estimation (where neural nets learn aging effects) and vehicle dynamics estimation (learning tire force models). The hybrid model+data trend is evident: rather than rely on pure black-box models, the literature is converging on methods that treat learning as an augmentation to robust observers – aligning with the philosophy of this thesis.

Applications in Resilient Control and Attack Mitigation: The ultimate motivation behind these estimation advances is to enable resilient control strategies. A robust observer is often one half of a resilient control loop – it provides the feedback state (and possibly disturbance estimates) that a controller needs to adjust and maintain performance under adversity. For example, in a resilient vehicle platooning scenario, an observer might estimate a sudden drop in lead vehicle speed due to an attack, and the following vehicles' controllers can then temporarily revert to a safe gap until trust is restored. Many research efforts combine observers with fault-tolerant control laws: e.g., using the estimated fault from a UIO to reconfigure control allocation (as in active fault-tolerant control systems). In the cyber-security

realm, observer-based attack detection and isolation is a key enabler – by comparing output estimates with actual sensor readings (residual analysis), one can detect inconsistencies caused by attacks. Fawzi et al. (2014) and Pajic et al. (2017) studied fundamental limits on state estimation under sparse sensor attacks, showing that with enough redundancy, observers can exactly recover the true state even if some sensors are malicious. In practical terms, works like Jeon et al. (2020) on connected vehicles exemplify applying such principles: the observer not only estimates the vehicle state but also flags which sensor/communication channels may be under attack, enabling a higher-level system to exclude or downweight those signals. Cooperative Adaptive Cruise Control (CACC) testbeds often implement observers for estimating the lead vehicle's intentions or for filtering V2V data – these ensure that a glitch or spoof in the communication does not directly corrupt the vehicle following behavior. Some literature reports using software-in-the-loop and hardware-in-the-loop experiments (with tools like MATLAB/Simulink, CARLA, or PreScan) to validate these resilient estimation and control setups before on-road trials. As a result, there is a growing body of benchmarks and case studies demonstrating that advanced observers can indeed increase the resilience of autonomous vehicles. For instance, the CARLA simulation in Chapter 2 (involving a platoon under a V2V attack) is in line with other studies that use high-fidelity simulators to test observer performance in realistic traffic scenarios. The positive outcomes in those simulations – e.g., successfully detecting an attack or maintaining safe inter-vehicle distances despite sensor faults – reinforce the practical value of the surveyed estimation techniques.

In summary, the literature shows a clear evolution toward resilient, hybrid estimation methods that are applicable to connected and autonomous vehicles. Unknown Input Observers and sliding mode observers provide a backbone for robustness against disturbances and attacks, while adaptive and learning elements address modeling mismatches and complex nonlinearities. Cooperative estimation extends these concepts to networked vehicle systems, ensuring that the benefits of connectivity (improved awareness and redundancy) can be realized without succumbing to its vulnerabilities. The subsequent chapters (Chapters 2–4) build upon these key ideas – each chapter delves into a specific advancement (from resilient UIO design to learning-enhanced observers and distributed estimation in platoons), contributing novel algorithms aligned with the directions identified in this review. Through both theoretical development and simulation validation, we aim to advance the state of the art in resilient state estimation for autonomous and connected vehicles, informed by the rich body of existing knowledge summarized above.

4 Positioning and challenges

Three methodological challenges motivate the developments in this manuscript.

- **Unknown input reconstruction under realistic assumptions:** estimating attacks/disturbances is not only a detection problem; reconstruction is needed for mitigation and control. Yet, classical unknown input observer (UIO) designs rely on rank/matching conditions that may fail for practical vehicle models.

- **Nonlinear and time-varying uncertainty:** vehicle dynamics include effects that are difficult to capture with fixed parametric models (e.g., aggregated tire/road effects, drag, actuator dynamics variations). Learning can approximate these uncertainties, but must be embedded into a stable

observer structure.

- **Distributed estimation with adversarial information:** in platoons, estimation depends on exchanged data. Robustness requires mechanisms to reduce the influence of compromised or unreliable neighbors without destroying convergence.

The remainder of the thesis proposes observer designs addressing these points in a unified resilient estimation viewpoint.

5 Approach overview

The thesis follows a consistent workflow: (i) formulate vehicle/communication models with disturbances and attacks as unknown inputs, (ii) design observers with explicit robustness properties, (iii) enrich the estimators with learning or trust layers when needed, and (iv) validate by simulation on representative scenarios.

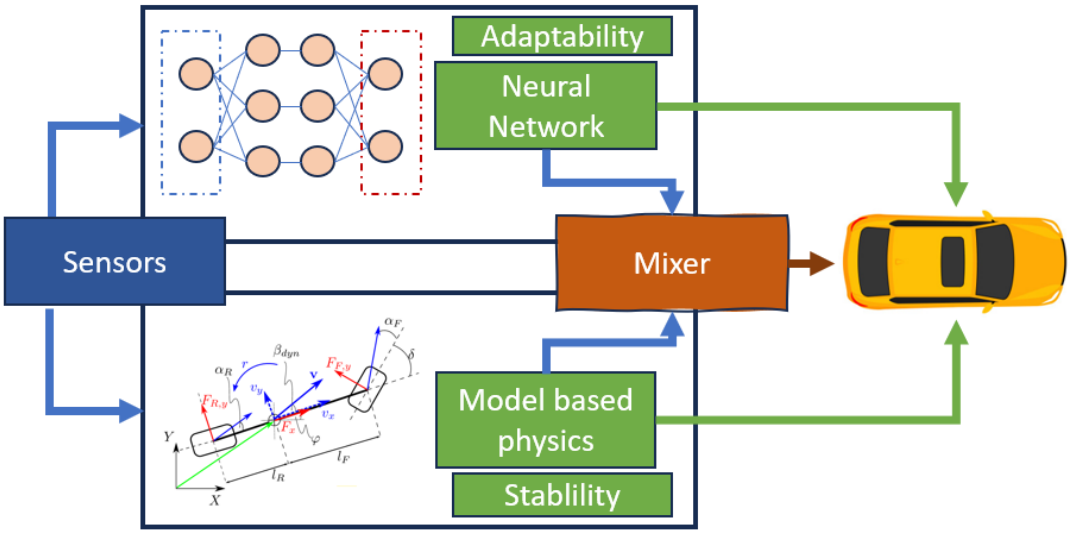


Figure 1.1: Hybrid resilient estimation framework for connected vehicles

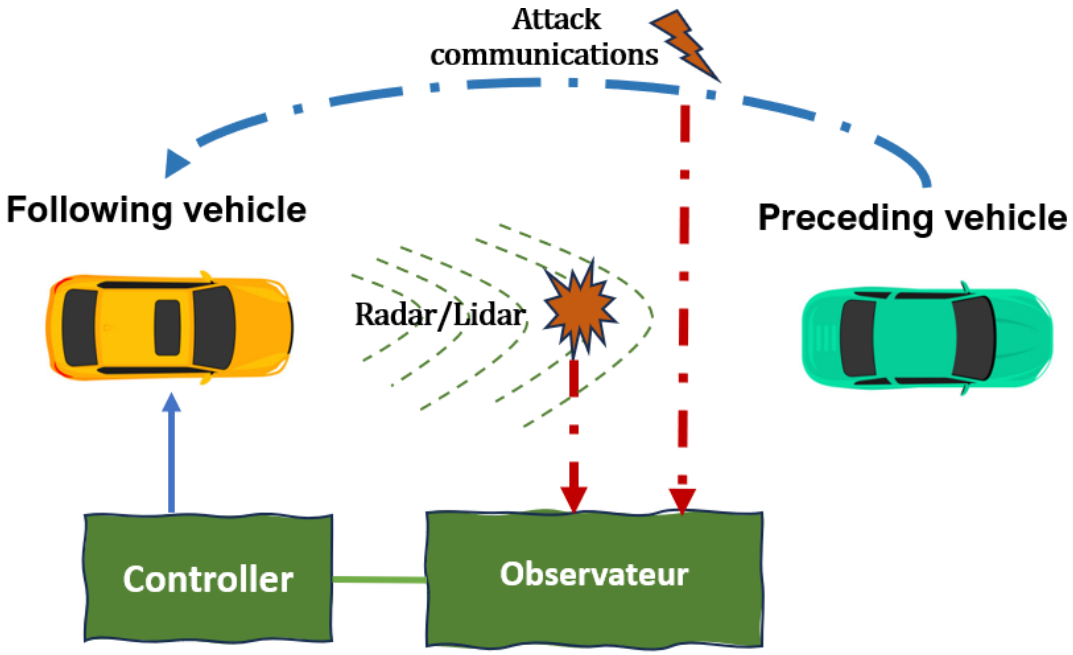


Figure 1.2: CACC under cyber-attack scenario

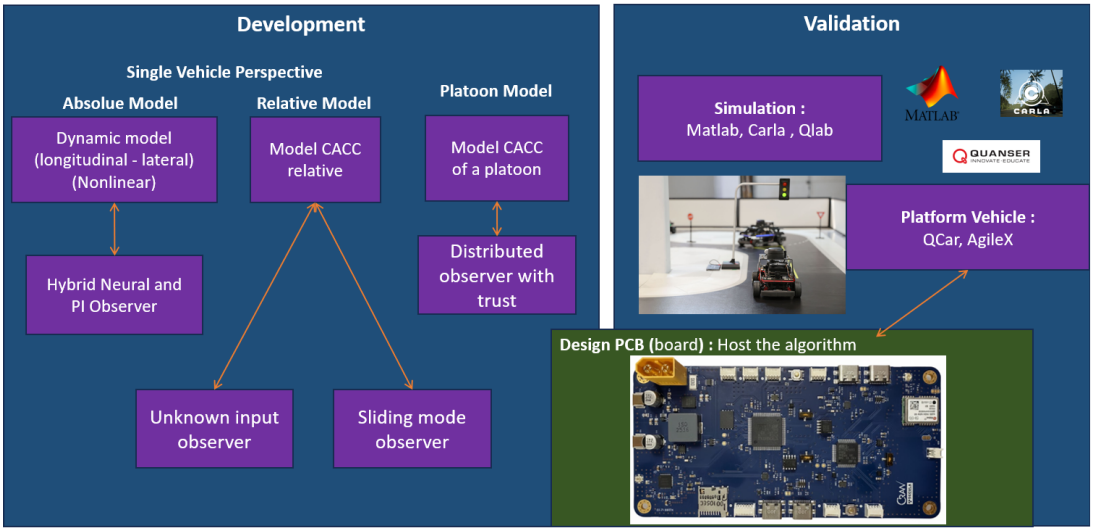


Figure 1.3: Organization of the thesis

6 Main contributions

The main contributions of this thesis can be summarized as follows:

- **Resilient estimation and attack reconstruction for CACC:** an observer-based framework that estimates vehicle states and reconstructs corrupted communicated signals modeled as unknown inputs.
- **Learning with generalized UIOs for nonlinear estimation:** a hybrid teacher–student architecture combining a generalized UIO (robust bounded estimation) with a neural adaptive observer (learning structured uncertainty) under LMI-based stability conditions.
- **Trust-aware distributed state estimation for platoons:** a distributed observer architecture that adapts neighbor influence using a trust/divergence metric to maintain estimation performance under cyber/communication attacks.

Overall, the thesis contributes observer designs that combine robustness, adaptivity, and resilience in connected vehicle applications.

7 Organization of the manuscript

The manuscript is organized as follows.

- Chapter 2 studies resilient state estimation and cyber-attack reconstruction for connected autonomous vehicles in a CACC setting.
- Chapter 3 introduces a learning-based unknown input observer framework for robust nonlinear estimation, combining a generalized UIO with a neural adaptive observer.
- Chapter 4 proposes a resilient trust-aware distributed observer design for connected vehicle platoons under adversarial conditions.

- The conclusion summarizes the main results and outlines perspectives for future work.

Chapter 2

Resilient State Estimation and Cyber-Attack Reconstruction for Connected Autonomous Vehicles

Objectives

The primary objective of this chapter is to enhance the resilience of Cooperative Adaptive Cruise Control (CACC) systems against cyberattacks. Specifically, this work aims to:

- Design a discrete-time Unknown Input Observer (UIO) capable of simultaneously estimating relative vehicle states and reconstructing false data injection signals.
- Address the structural rank condition failures caused by standard discretization through advanced modeling techniques (output delay and exact discretization).
- Develop an observer-based control strategy that compensates for attacks in real-time, ensuring the safety and stability of the vehicle platoon.

Contents

1 Introduction

Connected autonomous vehicles (CAVs) have revolutionized conventional transportation systems through the facilitation of wireless communication and interaction among vehicles [Bu, 2012]. A significant stride in this domain is the emergence of Cooperative Adaptive Cruise Control (CACC), which builds upon the capabilities of Adaptive Cruise Control (ACC). While ACC autonomously adjusts vehicle speeds based on preceding traffic, CACC elevates this functionality by integrating inter-vehicle communication. As wireless communication and data exchange become more prevalent in CAVs, there is a growing concern about cyberattacks. It is essential to detect these attacks to maintain the safety, reliability, and integrity of vehicular communication systems. These attacks can take many forms, such as false data injection, Denial of Service (DoS), eavesdropping, or interference, targeting communication channels or sensor systems. Hence, detecting and mitigating these attacks is crucial for ensuring that CAVs function correctly and road safety is maintained [Khalil, 2020; Huang, 2022; Mousavinejad, 2019; Cabelin, 2021].

In prior work, a proportional integral observer was introduced in [Pan, 2023b] for state estimation and spoofing attack detection, and in [Pan, 2023a] for addressing the DoS attack issue by estimating data transmission delays. Nevertheless, these results are limited to scenarios where the attack evolves slowly. Some other work such as [Yamamoto, 2021] and [Cheng, 2023] utilize the concept of unknown input observer (UIO) residual. This residual is used to detect attacks and is enhanced by setting an adaptive threshold that considers the impact of disturbances, thereby improving the accuracy of the attack detector. This UIO-based technique, as utilized also in [Jeon, 2020] within the ACC framework, is specifically designed for constant cyberattack signals.

In this paper, we address the challenge of cyberattack detection by utilizing an UIO to estimate attacks and create an observer-based control to mitigate their adverse effects on platoons with CACC. This algorithm can also be used to monitor cyberattack status with different types of control and modifications. A significant difference in our method is that we transmit data using the acceleration of the preceding vehicle rather than the control input. The advantage of this approach is that in a platoon, we cannot have the same control input, and using acceleration data is more suitable. In contrast to the methodology presented in [Pan, 2023b; Pan, 2023a; Jeon, 2020], our proposed approach demonstrates the capability to effectively handle time-varying cyberattack. Unlike the framework outlined in [Pan, 2023b; Pan, 2023a; Jeon, 2020], which offers adaptability to dynamic cyberattack scenarios.

On each vehicle, the UIO-based estimator computes the relative state between the preceding and following vehicles while examining any potentially erroneous communication signal data. In the event of a cyberattack, a resilience-controlled system is activated, leveraging the state estimated from UIO. Simulation experiments have demonstrated that the proposed mechanism accurately estimates the system state with a 1-step delay and detects attacks with a 2-step delay. This mechanism not only enhances the system stability but also mitigates safety losses induced by cyberattacks.

2 Problem Formulation

2.1 General description of the CACC system

The CACC scenario working on a platoon can be illustrated as in Figure 2.1. In Figure 2.1, a_{i-1} represents

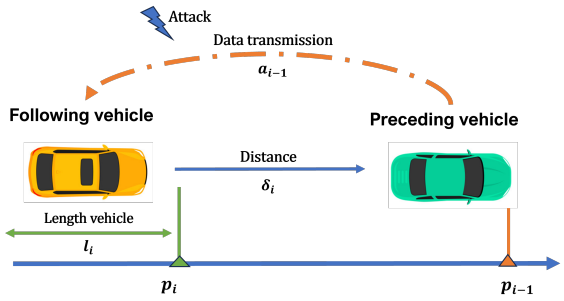


Figure 2.1: Illustration of CACC working on a vehicle platoon.

the acceleration of the preceding vehicle transmitted through a wireless communication channel to the following vehicle; δ_i denotes the distance between the rear of the preceding vehicle and the bumper of the following vehicle; l_i is the length of the following vehicle; p_i and p_{i-1} correspond to the positions of the following and preceding vehicles, respectively.

The CACC control system of a vehicle comprises two controllers: an upper-level controller and a lower-level controller. The lower-level controller is responsible for utilizing throttle and/or brake control inputs to achieve the desired acceleration tracking. On the other hand, the upper-level controller is tasked with calculating the desired acceleration to maintain the desired spacing with respect to a preceding vehicle. In line with the approach discussed in [Rajamani, 2002], this paper focuses exclusively on the investigation of the upper-level controller, operating under the assumption that a suitable lower-level controller is already in place.

2.2 CACC modeling subject to cyberattack

Basically, the dynamics of i^{th} vehicle are a Position-Velocity-Acceleration third-order linear model given by the following set of equations:

$$\begin{cases} \dot{p}_i(t) = v_i(t) \\ \dot{v}_i(t) = a_i(t) \\ \dot{a}_i(t) = \frac{1}{\tau_i} u_i(t) - \frac{1}{\tau_i} a_i(t) \end{cases} \tag{2.1}$$

where

- $p_i(t), v_i(t)$ and $a_i(t)$ denote the position, velocity and acceleration of the i^{th} vehicle, respectively;
- $u_i(t)$ is the i^{th} vehicle control input representing the desired driving/braking force or acceleration;
- τ is the engine's constant time lag.

The actual spacing between vehicles are $\varepsilon_{i,act} = p_{i-1} - p_i$ and the desired spacing between vehicles in the constant time-gap spacing policy is given by $\varepsilon_{i,des} = L + hv_i$. It follows that the spacing error can be written as:

$$\varepsilon_{i,des} - \varepsilon_{i,act} \triangleq p_i - p_{i-1} + L + hv_i = \varepsilon_i + hv_i, \tag{2.2}$$

where L is the minimum safety distance and h is the time-gap. Then, the controller u_i is given by [Rajamani, 2002]:

$$u_{\mathrm{i}} = -k_1 a_{i-1} + (k_1 + hk_1k_2)a_i - \frac{1}{h}(1 - hk_1k_2)\dot{\varepsilon}_i - \frac{k_2}{h}\varepsilon_i - k_2 v_i \tag{2.3}$$

where a_{i-1} is the acceleration of the preceding vehicle obtained by using inter-vehicle communication. The scalar gains k_1 and k_2 are the controller design parameters to be designed according to the detailed procedure introduced in [Rajamani, 2002].

The controller u_{i} defined in (2.3) is convenient in perfect situation where the acceleration of the preceding vehicle, a_{i-1}, provided through V2V communication channel is not corrupted by intruders. Unfortunately, in CACC platoons, the V2V communication may provide unreliable acceleration due to potential tampering by attackers. To develop a resilient controller to cyberattacks, we have to consider the attack signal in the design of the controller. To this purpose, we assume that the following vehicle receives the signal $\mu(t)$ corrupted by injected false data in the wireless communication channel:

$$\mu(t) = a_{i-1}(t) + f^c(t) \tag{2.4}$$

where $a_{i-1}(t)$ is the true acceleration and $f^c(t)$ is the additive cyberattack signal. This cyberattack scheme under additive false signals can cover several cyberattack architectures. These may include message falsification attacks, spoofing attacks, and denial of service (DoS) attacks. For instance, in the DoS case, by using the differential mean value theorem, there exists $\theta_t \in [t - \tau(t),\ t]$ such that the delayed signal $a_{i-1}(t - \tau(t))$ can be represented as follows:

$$\begin{aligned}\mu(t) = a_{i-1}(t - \tau(t)) &= a_{i-1}(t) \underbrace{- \frac{\mathrm{d}a_{i-1}}{\mathrm{d}t}(\theta_t)\tau(t)}_{f^c} \\ &= a_{i-1}(t) + f^c.\end{aligned} \tag{2.5}$$

Under a cyberattack, the following vehicle receives μ and will utilize it in its controller. Then, instead of (2.3), the controller u_{i} will be implemented as follows:

$$u_{\mathrm{i}} = -k_1 \mu(t) + (k_1 + hk_1k_2)a_i - \frac{1}{h}(1 - hk_1k_2)\dot{\varepsilon}_i - \frac{k_2}{h}\varepsilon_i - k_2 v_i. \tag{2.6}$$

As shown in Figure 2.1, the radar-measured rear-to-bumper distance is defined as follows:

$$\delta_i = p_{i-1} - p_i - l_{i-1}, \tag{2.7}$$

Then, taking the time derivative of δ_i results in:

$$\dot{\delta}_i = v_{i-1} - v_i, \ddot{\delta}_i = a_{i-1} - a_i, \dddot{\delta}_i = \dot{a}_{i-1} - \dot{a}_i \tag{2.8}$$

where $\dot{\delta}_i$, $\ddot{\delta}_i$, and $\dddot{\delta}_i$ are the relative speed, acceleration, and jerk (rate of acceleration changes) between

two adjacency vehicle. By using (2.7) and (2.8), the controller (2.6) becomes

$$\begin{aligned} u_{\mathrm{i}} = (hk_1k_2)\,\mu - (k_1 + hk_1k_2)\,f^c - k_2 v_i - \frac{k_2}{h}\,(L - l_{i-1}) \\ - (k_1 + hk_1k_2)\,\ddot{\delta}_i + \frac{1}{h}\,(1 - hk_1k_2)\,\dot{\delta}_i + \frac{k_2}{h}\,\delta_i. \end{aligned} \tag{2.9}$$

We obtain acceleration information of the preceding vehicle directly through wireless communication. Consequently, we introduce the following assumption regarding the jerk of the preceding vehicle.

Assumption 1 (Bounded Jerk). *The jerk of the preceding vehicle is assumed to be negligible over one sampling period, i.e.,*

$$\dot{a}_{i-1} \approx 0. \tag{2.10}$$

This assumption is standard in upper-level CACC design and is justified by the limited bandwidth of vehicle actuators.

Substituting equation (2.9) into (2.1) and using (2.10) and the notation $x = \begin{bmatrix} \delta_i & \dot{\delta}_i & \ddot{\delta}_i \end{bmatrix}^\top$, we obtain the model expressed under the following condensed matrix form:

$$\begin{cases} \dot{x} = Ax + Bv_i + F\mu + \Delta - Wf^c \\ y = Cx \end{cases} \tag{2.11}$$

with

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{-k_2}{\tau h} & \frac{-k_3}{\tau h} & \frac{(k_1 - k_3)}{\tau} \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ \frac{k_2}{\tau} \end{bmatrix}, F = \begin{bmatrix} 0 \\ 0 \\ \frac{k_3}{\tau} \end{bmatrix},$$

$$\Delta = \begin{bmatrix} 0 \\ 0 \\ \frac{k_2(L - l_{i-1})}{\tau h} \end{bmatrix}, W = \begin{bmatrix} 0 \\ 0 \\ \frac{k_3 - k_1}{\tau} \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

where $k_3 \triangleq 1 - hk_1k_2$.

To avoid cumbersome equations, we omit the subscripts i and $i-1$ when representing the interaction between two cars in the CACC scenario. We remove these subscripts from the matrices A, B, F, Δ, W, and C, as well as from the vectors x, y, and f_c. We retain only v_i to avoid confusion with previous equations.

2.3 Problem formulation and objectives

The primary objective in the CACC control problem is to estimate the false data, f^c, and compensate for it within the controller u_{i}. A natural solution involves using the Unknown Input Observer (UIO) technique to simultaneously estimate the state x and the false data f^c.

However, standard UIO design requires a specific rank condition, namely $\mathrm{rank}(CW) = \mathrm{rank}(W)$ [Chaouche, 2022; Trinh, 2011]. For the CACC system defined in (2.11), we observe that $\mathrm{rank}(CW) = 0$ while $\mathrm{rank}(W) = 1$. Consequently, the standard rank constraint is not satisfied, preventing direct application of the method.

To overcome this limitation, we require additional information, specifically the relative acceleration. One approach is to use a real-time differentiator to estimate the derivative of the relative velocity online. Since $CB = CF = C\Delta = CW = 0$, we can define the derivative of y as a pseudo-measurement:

$$\bar{y} \triangleq \dot{y} = CAx = \bar{C}x. \tag{2.12}$$

With this new measurement, we verify that $\text{rank}(\bar{C}W) = \text{rank}(W) = 1$, which allows us to construct a UIO to estimate both x and f^c, provided that additional stability conditions are met.

There are numerical differentiators in the literature allowing the online calculation of the derivative of a given signal, such as the famous sliding mode observer [Zhu, 2012], and the high-gain observer [Dabroom, 1997]. However, these numerical differentiators fail in some situations, namely in the presence of sensor noises. An alternative solution to avoid real-time estimation of the derivative of y consists in investigating the problem by using the discrete-time version of the model, as shown in Figure 2.2 which considers additive sensor noise ω.

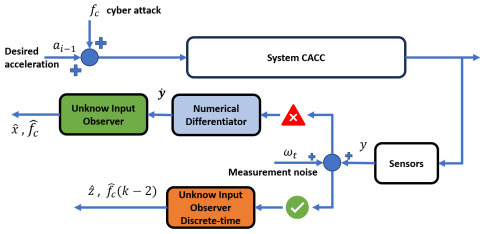


Figure 2.2: Discrete-time model-based UIO.

Indeed, in discrete time, there is no differentiation since the measured signal is a sequence of numbers. The discrete-time version of equation (2.11) is given as follows:

$$\begin{cases} x_{k+1} = \mathcal{A}x_k + \mathcal{B}v_{i,k} + \mathcal{F}\mu_k + \bar{\Delta} - \mathcal{W}f_k^c \\ y_k = Cx_k \end{cases} \tag{2.13}$$

where

$$\mathcal{A} = (\mathbb{I}_3 + TA),\ \mathcal{B} = TB,$$

$$\mathcal{F} = TF,\ \mathcal{W} = TW,\ \bar{\Delta} = T\Delta,$$

and T being the sampling time. In this case, we propose shifting the output equation backward using the discrete-time state equation (2.13). This allows us to express the output y_k as a function of the delayed state $x_{k-\tau}$, where τ is the smallest positive integer that satisfies the necessary rank condition.

In the following section, we adopt this strategy to develop a discrete-time model-based UIO for the CACC system, enabling the estimation of the cyberattack signal f^c.

3 Discrete-Time Model-Based UIO design for CACC system

This section is devoted to the development of a new unknown input observer based on the discrete-time model (2.13). By introducing the variables

$$z_t \triangleq x_{k-1} \text{ and } \bar{y}_t \triangleq y_{k-1} \tag{2.14}$$

with $t = k - 1$ and considering additive vector noises, ω_t, in both the output measurements and in the process equation, we obtain the following new system

$$\begin{cases} z_{t+1} = \mathcal{A}z_t + \mathcal{B}v_{i,t} + \mathcal{F}\mu_t \\ \qquad\quad +\bar{\Delta} - \mathcal{W}f_t^c + E_\omega\omega_t \\ \bar{y}_t = \mathcal{C}z_t + D_\omega\omega_t \end{cases} \tag{2.15}$$

for which the rank condition is satisfied, i.e.:

$$\text{rank}(\mathcal{C}\mathcal{W}) = \text{rank}(\mathcal{W}) = 1. \tag{2.16}$$

3.1 UIO structure for the shifted system

By using an UIO corresponding to the shifted system (2.15), we will estimate simultaneously z_t and f_{t-1}^c. By using the notation

$$\xi_t \triangleq \begin{bmatrix} z_t \\ f_{t-1}^c \end{bmatrix},$$

$$\mathbb{E} = \begin{bmatrix} \mathbb{I}_3 & \mathcal{W} \end{bmatrix},\ A_\xi = \begin{bmatrix} \mathcal{A} & 0 \end{bmatrix},\ C_\xi = \begin{bmatrix} \mathcal{C} & 0 \end{bmatrix}$$

the system (2.15) is written under the following descriptor form:

$$\begin{cases} \mathbb{E}\xi_{t+1} = A_\xi\xi_t + \mathcal{B}v_t + \mathcal{F}\mu_t + \Delta + E_\omega\omega_t \\ \bar{y}_t = C_\xi\xi_t + D_\omega\omega_t. \end{cases} \tag{2.17}$$

From (2.16), we have

$$\text{rank}\left(\begin{bmatrix} \mathbb{E} \\ C_\xi \end{bmatrix}\right) = n + \text{rank}(\mathcal{C}\mathcal{W}) = n + 1.$$

Now, let P_z and Q_z be two matrices of appropriate dimensions such that

$$\begin{bmatrix} P_z & Q_z \end{bmatrix} = \left(\begin{bmatrix} \mathbb{E} \\ C_\xi \end{bmatrix}^\top \begin{bmatrix} \mathbb{E} \\ C_\xi \end{bmatrix}\right)^{-1} \begin{bmatrix} \mathbb{E} \\ C_\xi \end{bmatrix}^\top. \tag{2.18}$$

It follows that

$$P_z\mathbb{E} + Q_zC_\xi = \mathbb{I}_p \tag{2.19}$$

where p is the number of output measurements.

By considering the following two-stage observer

$$\begin{cases} \kappa_{t+1} = \Big(P_z A_\xi - KC_\xi\Big)\kappa_t + P_z \mathcal{B} v_{i,t} \\ \qquad\quad + \Big[\Big(P_z A_\xi - KC_\xi\Big)Q_z + K\Big]\,\bar{y}_t \\ \qquad\quad + P_z \mathcal{F}\mu_t + P_z \Delta \\ \hat{\xi}_t = \kappa_t + Q_z \bar{y}_t \end{cases} \tag{2.20}$$

the estimation error $\tilde{\xi}_t = \hat{\xi}_t - \xi_t$ satisfies the equation:

$$\begin{aligned} \tilde{\xi}_{t+1} = &\Big(P_z A_\xi - KC_\xi\Big)\tilde{\xi}_t \\ &+ \Big[\Big(KD_\omega - P_z E_\omega\Big) \quad Q_z D_\omega\Big]\,\bar{\omega}_t, \end{aligned} \tag{2.21}$$

where $\bar{\omega}_t$ is defined as follows:

$$\bar{\omega}_t \triangleq \begin{bmatrix} \omega_t^\top & \omega_{t+1}^\top \end{bmatrix}^\top. \tag{2.22}$$

Remark 1. *The central focus of this paper does not lie in the details of designing controller parameters k_1 and k_2. Rather, our attention is directed towards the development of a novel cyberattack detection algorithm. It is crucial to note that in the proposed methodology, we make an assumption that the control design phase, specifically ensuring the string stability of the controller (2.3), is considered as a distinct and independent process from the estimation design procedure outlined in this paper. As demonstrated in [Rajamani, 2002], the controller (2.3) achieves string stability under specific conditions. This stability depends on the transfer function $H(s) = \frac{\bar{\varepsilon}_i}{\bar{\varepsilon}_{i-1}}$, where $\bar{\varepsilon}_i$ is defined in (2.2), satisfying the inequality $|H(j\omega)| \leq 1$. Detailed analysis in [Rajamani, 2002] reveals that such a condition is met when two critical parameters adhere to certain criteria. Specifically, the condition holds when $-k_1 h > \tau$ and $k_2 > 0$, as meticulously explained in the calculations provided in [Rajamani, 2002, Eqs.(32)-(34)].*

3.2 LMI-Baed ISS design method

The objective consists in determining the observer gain K such that the estimation error $\tilde{\xi}_t$ is exponentially robust with respect to the bounded disturbance ω_t. The next theorem provides sufficient conditions expressed in terms of LMIs ensuring the robust exponential stability of $\tilde{\xi}_t$. For the sake of brevity, we put $\mathcal{A}_z \triangleq P_z A_\xi$, $\mathcal{D}_z \triangleq Q_z D_\omega$, and $\mathcal{E}_z \triangleq P_z E_\omega$.

Theorem 1. *Assume there exist two symmetric and positive definite matrices $\mathcal{P}$ and $\mathcal{S}$, a matrix $\mathcal{Z}$ of appropriate dimension, and a scalar α, with $\alpha \in]0,1[$, such that the following LMI condition holds:*

$$\begin{bmatrix} -\alpha\mathcal{P} & 0 & \mathcal{A}_z^\top \mathcal{P} - C_\xi^\top \mathcal{Z} \\ (\star) & -\mathcal{S} & \begin{bmatrix} D_\omega^\top \mathcal{Z} - \mathcal{E}_z^\top \mathcal{P} \\ \mathcal{D}_z^\top \mathcal{P} \end{bmatrix} \\ (\star) & (\star) & -\mathcal{P} \end{bmatrix} < 0 \tag{2.23}$$

Then the estimation error $\tilde{\xi}_k$, with $K = \mathcal{P}^{-1}\mathcal{Z}^\top$, satisfies the following exponential Input-to-State Stable (ISS) bound:

$$\boxed{\left\|\tilde{\xi}_t\right\| \leq \sqrt{\frac{\lambda_{\max}(\mathcal{P})}{\lambda_{\min}(\mathcal{P})}}\left\|\tilde{\xi}_0\right\| \alpha^{\frac{t}{2}} + \sqrt{\frac{\lambda_{\max}(\mathcal{S})}{(1-\alpha)\lambda_{\min}(\mathcal{P})}} \max_{0\leq k\leq t} \|\bar{\omega}_k\|.} \tag{2.24}$$

Proof. The proof is based on the Lyapunov function $\vartheta_t \triangleq \tilde{\xi}_t^\top \mathcal{P}\tilde{\xi}_t$. By expanding the expression of ϑ_{t+1} along the trajectories of (2.21), we can deduce easily that under the LMI condition (2.23), we have

$$\vartheta_{t+1} - \alpha\vartheta_t \leq \bar{\omega}_t^\top \mathcal{S}\bar{\omega}_t.$$

Hence, by induction technique and backward substitution, we deduce that

$$\vartheta_t \leq \vartheta_0\alpha^t + \frac{\lambda_{\max}(\mathcal{S})}{1-\alpha} \max_{0\leq k\leq t} \|\bar{\omega}_k\|^2.$$

Consequently, we can conclude by using the double inequality

$$\lambda_{\min}(\mathcal{P})\left\|\tilde{\xi}_t\right\|^2 \leq \vartheta_t \leq \lambda_{\max}(\mathcal{P})\left\|\tilde{\xi}_t\right\|^2.$$

□

3.3 Defense strategy

The key idea consists of developing a Resilient, Robust, and Reliable CACC ($3\mathbb{R}-$CACC) controller able to cope with cyberattacks, external disturbances, and data loss. The $3\mathbb{R}-$CACC defense mechanism is as depicted in Figure 2.3. As soon as an attack or a significant external disturbance is detected, the CACC controller (2.3) will switch to the ACC controller to forget the false data in communication data.

$$u_i = -\frac{1}{h}(v_i - v_{i-1} + \lambda\bar{\epsilon}_i) \tag{2.25}$$

with the condition $h > 2\tau$ and $\lambda > 0$ in order to ensure the string stability. After implementing the controller (2.25), the system no longer remains the same as (2.13). Therefore, we need to transform the controller (2.25) in terms of x and rewrite a new system similar to (2.13). This enables the system to detect cyberattacks, which will be treated as unknown inputs.

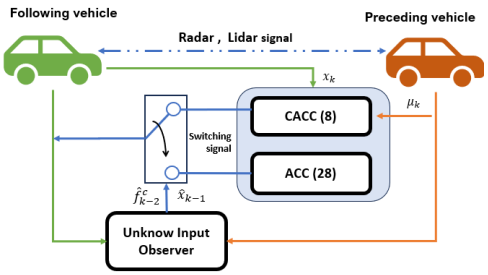


Figure 2.3: Resilient control for CACC system.

Remark 2. *The LMI conditions (2.23) correspond to the system without uncertainties, which guarantee certain robustness with respect to the parameter uncertainties in the sense of the exponential ISS criterion (2.24). Indeed, with the proposed method, if we have uncertainties, i.e.:* $A + \delta A$ *and* $C + \delta C$ *instead of* A *and* C*, for instance, we can add* $\delta Ax(t)$ *and* $\delta Cx(t)$ *in the disturbance vector* ω *and then the LMIs ensure the exponential ISS bound with respect to the new vector of disturbances,* ω*, containing the parameter uncertainties. However, such a technique does not ensure exponential convergence of the estimation and tracking errors to zero in the free-disturbance case (with the initial vector of disturbances without including the uncertain parameters), and the boundedness of* δA *and* δC *does not imply necessarily the boundedness of* $\delta Ax(t)$ *and* $\delta Cx(t)$*. To overcome this issue, we need to develop an extended LMI condition that consider both the structure and magnitude of uncertainties.*

4 Simulation results by Using Matlab and Carla Platform

We demonstrate the effectiveness of the proposed observer by conducting MATLAB and CARLA simulations with two cyberattack scenarios. The parameters of the CACC system and its controller are given in Table 2.1.

Parameter	τ	L	l_i	h	k_1	k_2	t_s
Value	0.4	7.3	5	0.5	-0.8	2.5	0.01
Unit	s	m	m	s	-	-	-

Table 2.1: Parameters of CACC system and the controller gains.

It is assumed that two vehicles are driving longitudinally on the road, and each vehicle is equipped with a cruise control system. The preceding vehicle should pursue the following desired trajectory: $a_{i-1}(t) = 3\sin(\frac{2\pi}{10})t$. The two simulation scenarios of cyberattacks explored here are outlined below:

Case 1: *Sparse attack, DoS, packet-loss:*

$$f^c(t) = \begin{cases} X(t) & 6 \leq t < 8 \\ a_{i-1}(t) & 10 \leq t < 15 \\ 0 & \text{otherwise} \end{cases} \tag{2.26}$$

Case 2: *False data injected :*

$$f^c(t) = \begin{cases} -5 & 6 \leq t < 8 \\ 2(t-4) & 10 \leq t < 15 \\ 0 & \text{otherwise} \end{cases} \tag{2.27}$$

where $X(t) \sim U(-5, 5)$ is a random variable following a uniform distribution, with probability $P(X(t)) = 0.2$.

4.1 Simulation result using Matlab

By solving the LMI condition (2.23), we obtain the observer gain K given by:

$$K = \begin{bmatrix} 0.5000 & -0.005 & 0.0498 & -8.9295 \\ -0.005 & 1.000 & -99.9950 & 1427.22 \end{bmatrix}^\top .$$

The proposed observer is designed to estimate the state vector, including relative position, relative velocity, and relative acceleration, as well as provide an accurate estimation of any cyberattacks even in the presence of a Gaussian noise $\omega_t = \mathcal{N}(0.02^2[m])$ for both system and measurement. For both attack scenarios, we set the initial conditions of the actual system as $\xi_0 = \begin{bmatrix} 5 & 10 & 0 & 0 \end{bmatrix}^\top$, while the initial state of the observer is set to $\hat{\xi}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top$.

Simulation results for the first scenario of cyberattack (2.26) are presented in Figure 2.4 and Figure 2.5. In Figure 2.4, we can see the error of the relative position, relative velocity, and relative acceleration along with their estimates. The observer demonstrates similar good performance in accurately reconstructing a random cyberattack and packet-loss, as depicted in Figure 2.5. As we can see, the estimated delay attack is exactly 2 steps.

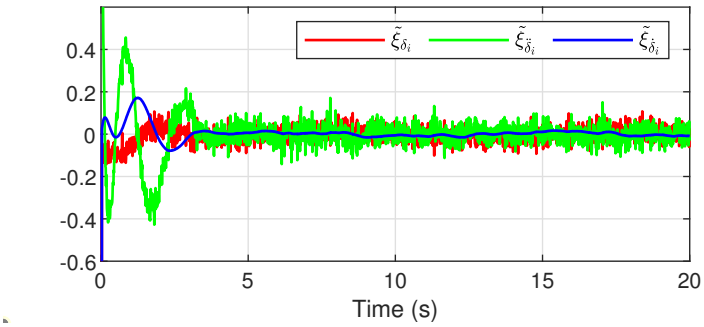


Figure 2.4: State estimation case 1.

For the second case of time-varying and bounded attack signal (2.27), it can be observed from the Figure 2.6 that the cyberattack signal is accurately estimated, and the estimation error is bound.

We can see that the proposed observer can estimate the the full state with good accuracy, while the unknown input is reconstructed with a fixed delay of two units based on the available output measurements.

4.2 Simulation result using Carla

To evaluate the system under more realistic conditions, we conducted simulations within the CARLA simulator. Specifically, we simulated a scenario where a leading vehicle utilized a PID controller for cruise control, aiming to maintain a velocity of $10m/s$. We kept the parameters consistent with those listed in Table 2.1 and adopted a time step of $t_s = 0.04$. We also implemented a deterministic attack signal that varies over time and is defined by equation (2.27). As depicted in Fig 2.8, the error state vectors for relative position and relative velocity exhibit good tracking, but the third vector $\tilde{\xi}_a$ falls short of perfection. This is because the acceleration of the preceding vehicle in Fig 2.9 sent to the network is inconsistent, which affects the attack estimation performance. Fig 2.9 illustrates the observer taking

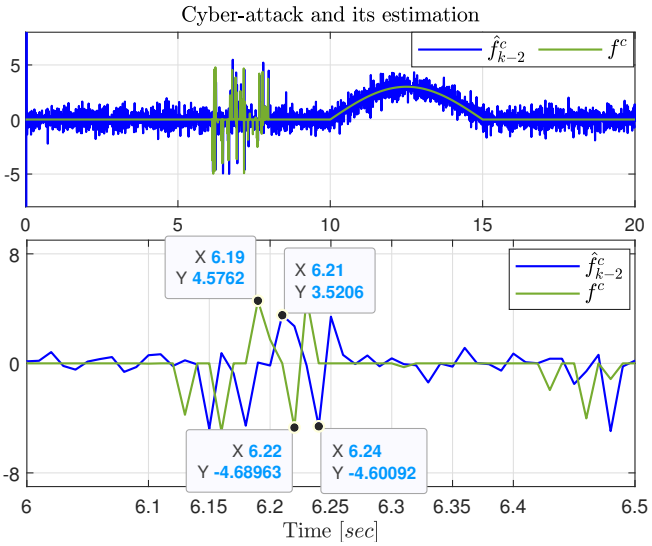


Figure 2.5: Attack estimation using UIO case 1.

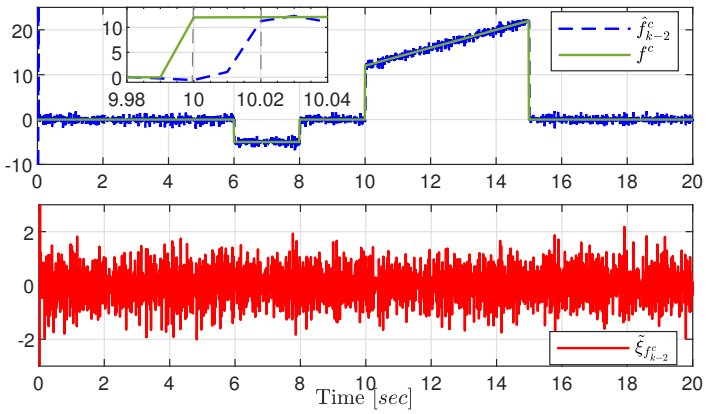


Figure 2.6: Attack estimation and the errors case 2.

at least 5 seconds to converge to the actual state, and the estimated value being influenced by noise and imperfection of the model. However, the leading vehicle retains its ability to detect the cyberattack signal, ensuring a safe distance effectively. A video showcasing the simulation scenario is provided in GitHub repository. Fig. 2.7 displays a thumbnail from the video.

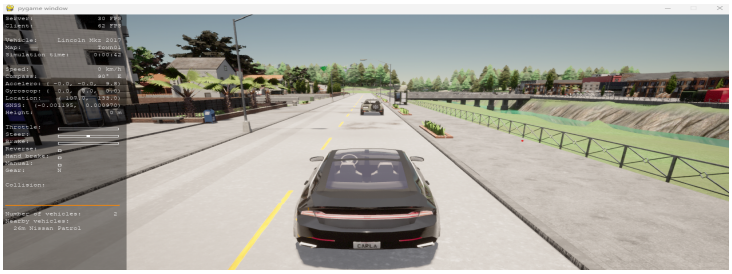


Figure 2.7: Carla Simulator platform (GitHub repository).

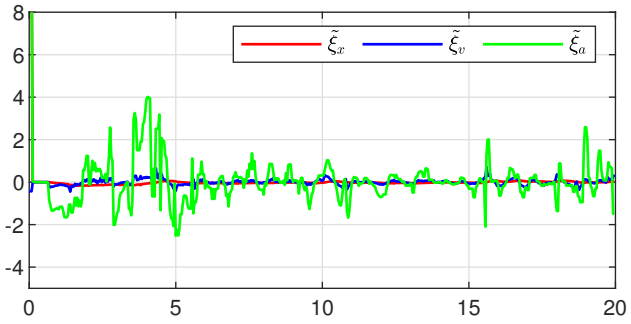


Figure 2.8: State estimation using CARLA under attack case 2.

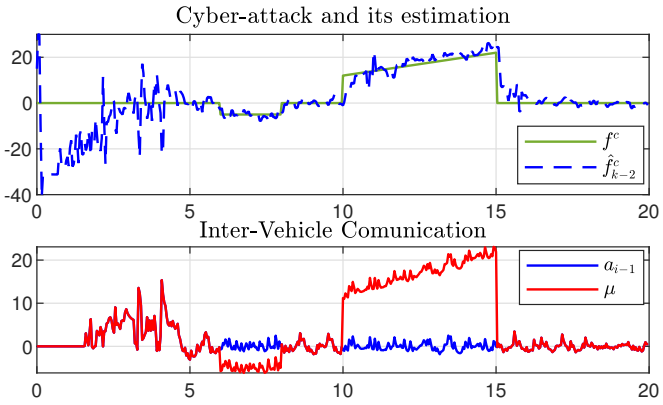


Figure 2.9: Cyberattack signal and inter-vehicle communication case 2.

5 Exact Discretization Approach

The observer design developed in the previous sections relies on a discrete-time model obtained via Euler discretization. This choice is deliberate, as it preserves the structure of the continuous-time system and allows direct application of well-established UIO existence and stability theorems. However, this discretization introduces a structural limitation, namely the violation of the rank condition, which necessitates the use of delayed outputs. The delay introduced in the previous sections is a discretization artifact, not a physical requirement. In this section, we show that the delay requirement is not intrinsic to the continuous-time dynamics of the CACC system, but rather induced by the Euler discretization method. Physically, an input force affects acceleration instantly, which in turn influences velocity and position over any time interval Δt. The Euler method ($p_{k+1} = p_k + \Delta t \cdot v_k$) effectively ignores the direct influence of acceleration on position within the sampling step, severing the algebraic link required for the rank condition. By adopting an exact discretization based on the matrix exponential [Zhang, 2021], the discrete-time model captures the complete evolution of the state trajectories between samples. This integration naturally couples the input with the position and velocity states through the matrix exponential, effectively preserving higher-order dynamics (such as $\frac{1}{2}a\Delta t^2$). This restores the structural coupling between the unknown input and the measured output, allowing the construction of a delay-free UIO. This approach offers significant advantages, including improved responsiveness and a simplified observer structure. Figure 2.10 illustrates the structure of the proposed observer-based cyberattack

estimation framework using the exact discretization model.

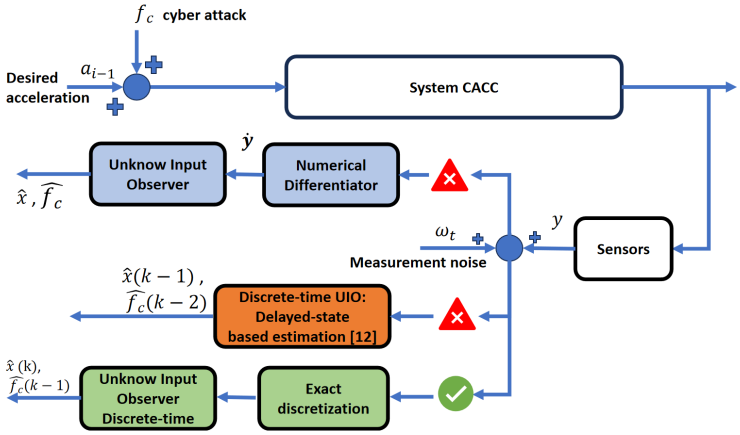


Figure 2.10: Discrete-time model-based UIO

Following the exact discretization method detailed in [Zhang, 2021], the discrete-time model of the CACC system is given by:

$$\begin{cases} x_{k+1} = A_d x_k + B_d v_{i,k} + F_d \mu_k + \hat{\Delta} - W_d f_k^c \\ y_k = C_d x_k \end{cases} \tag{2.28}$$

where the system matrices are computed as:

$$A_d = e^{AT_s}, \quad B_d = \int_0^{T_s} e^{A\eta} B d\eta, \quad C_d = C,$$
$$F_d = \int_0^{T_s} e^{A\eta} F d\eta, \quad W_d = \int_0^{T_s} e^{A\eta} W d\eta, \quad \hat{\Delta} = \int_0^{T_s} e^{A\eta} \Delta d\eta,$$

with T_s denoting the sampling period.

We now verify the rank condition for the discretized system (2.28). Using the system parameters given in Table 2.1, we obtain:

$$C_d W_d = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1.14 \times 10^{-6} \\ 3.41 \times 10^{-4} \\ 6.75 \times 10^{-2} \end{pmatrix} = \begin{pmatrix} 1.14 \times 10^{-6} \\ 3.41 \times 10^{-4} \end{pmatrix} \neq 0.$$

Since $C_d W_d \neq 0$, the rank condition $\mathrm{rank}(C_d W_d) = \mathrm{rank}(W_d) = 1$ is satisfied. This confirms that the unknown input f_k^c can be estimated directly from the output y_k without requiring any delay.

5.1 Simulation Results and Comparison

To demonstrate the effectiveness of the exact discretization method and the proposed observer, we conducted simulations using MATLAB and CARLA, following the procedures outlined in [Nguyen, 2024a]. This section presents the results obtained with our method and compares them with those from [Nguyen, 2024a].

We solve the LMI conditions for the observer gain using the YALMIP Toolbox with the exponential stability parameter $\alpha = 0.5$. The resulting observer gain K is:

$$K = \begin{bmatrix} 0.5000 & -0.001 & -0.2 & 2.2 \\ -0.005 & 0.7 & -68.0 & 1004.2 \end{bmatrix}^\top.$$

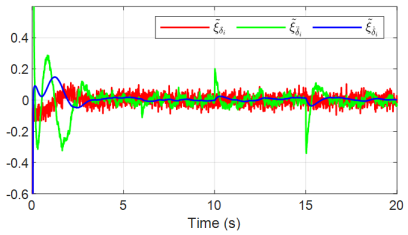


Figure 2.11: State estimation error using the proposed exact discretization method.

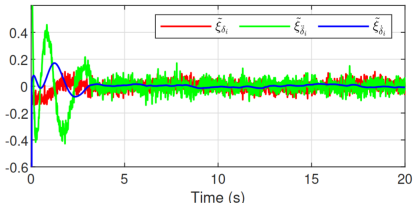


Figure 2.12: State estimation error from [Nguyen, 2024a].

Figures 2.11 and 2.12 illustrate the estimation errors for the system states (position, velocity, and acceleration). Figure 2.12 shows the results from [Nguyen, 2024a], where the estimation errors exhibit significant initial oscillations lasting about 5 seconds before stabilizing. Furthermore, the noise amplitude is substantial, indicating reduced accuracy under cyberattacks.

In contrast, Figure 2.11 shows the results obtained with our exact discretization method. The initial oscillations are less pronounced, and the system stabilizes more rapidly (within approximately 3 seconds). Moreover, the noise amplitude in both position and velocity errors is significantly reduced. This demonstrates the superior accuracy and robustness of the proposed method. Notably, the velocity estimation error is much smaller compared to the method in [Nguyen, 2024a].

Figures 2.13 and 2.14 depict the actual cyberattack signal (blue) and its estimate (green), along with the estimation error. Both methods detect the attack, but our method (Figure 2.13) exhibits superior performance. The estimated signal shows fewer fluctuations and greater stability compared to [Nguyen, 2024a] (Figure 2.14).

The estimation error in our method is centered around zero with minimal fluctuations (amplitude ≈ 0.6), whereas the method from [Nguyen, 2024a] shows errors with an amplitude of around 2. This higher noise level in the previous method could compromise reliability in safety-critical scenarios.

5.1.1 CARLA Simulator Results

To validate the proposed method under more realistic conditions, we conducted tests using the CARLA simulator. We implemented the same scenario as in [Nguyen, 2024a], involving a lead vehicle and a

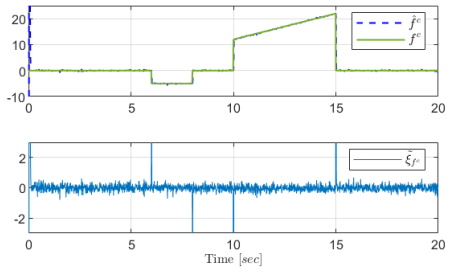


Figure 2.13: Cyberattack estimation and error using the proposed method.

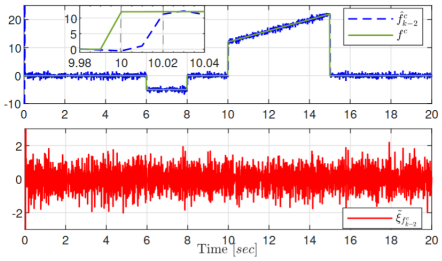


Figure 2.14: Cyberattack estimation and error from [Nguyen, 2024a].

following vehicle. The lead vehicle maintains a reference speed of 10 m/s using a PID controller. The simulation parameters match those used in the MATLAB simulation, with a time step of 0.04 seconds.

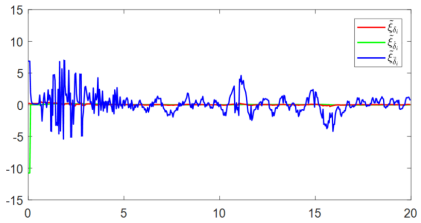


Figure 2.15: State estimation error in CARLA using the proposed method.

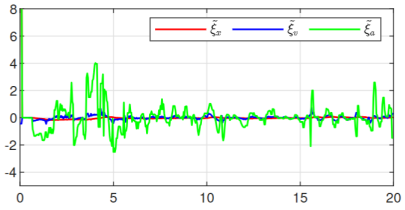


Figure 2.16: State estimation error in CARLA from [Nguyen, 2024a].

Figures 2.15 and 2.16 compare the state estimation errors in the CARLA environment. With our method (Figure 2.15), the position and velocity errors converge to zero more rapidly than in [Nguyen, 2024a] (Figure 2.16). The acceleration estimation takes about 5 seconds to stabilize, which is comparable to the reference method.

Figures 2.17 and 2.18 present the cyberattack estimation results. Our method (Figure 2.17) demonstrates better initial error management with smaller fluctuations compared to the significant transient

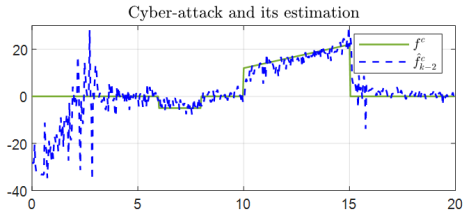


Figure 2.17: Cyberattack estimation in CARLA using the proposed method.

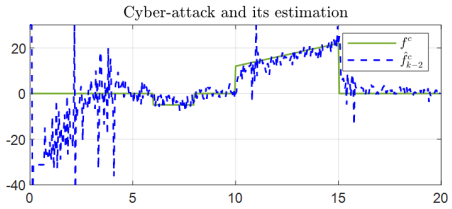


Figure 2.18: Cyberattack estimation in CARLA from [Nguyen, 2024a].

instability observed in [Nguyen, 2024a] (Figure 2.18).

Furthermore, our method achieves faster convergence, accurately estimating the cyberattack within 3 seconds, whereas the reference method requires more than 5 seconds. In terms of long-term accuracy, our method tracks the actual cyberattack signal f^c more closely, providing a more precise estimation than the method in [Nguyen, 2024a].

6 Conclusion

This chapter addressed the problem of resilient state estimation and cyberattack reconstruction for Connected Autonomous Vehicles (CAVs) operating in a Cooperative Adaptive Cruise Control (CACC) platoon. The primary challenge identified was the violation of the rank condition required for standard Unknown Input Observer (UIO) design when using classical Euler discretization.

To overcome this, we first proposed a discrete-time UIO design based on output delay. By shifting the output measurement backward, we recovered the necessary algebraic coupling between the unknown input and the measured output, enabling the simultaneous estimation of vehicle states and attack signals. We derived Linear Matrix Inequality (LMI) conditions to guarantee the Input-to-State Stability (ISS) of the estimation error in the presence of bounded disturbances.

Subsequently, we introduced a refined approach using exact discretization based on the matrix exponential. We demonstrated that the rank condition failure was an artifact of the Euler approximation rather than an intrinsic system limitation. The exact discretization method naturally preserves the higher-order dynamics, restoring the rank condition without the need for artificial delays.

Comparative simulations in both MATLAB and the high-fidelity CARLA simulator validated the effectiveness of both approaches. The results confirmed that while the delayed-output method is effective, the exact discretization approach offers superior performance in terms of convergence speed, estimation accuracy, and robustness to noise.

Future work will focus on extending these resilient estimation strategies to multi-vehicle platoons with

heterogeneous dynamics and investigating the impact of model parameter uncertainties. Additionally, we aim to integrate this observer-based detection with active fault-tolerant control strategies to further enhance the safety and resilience of autonomous transportation systems.

Chapter 3

Learning with Unknown Input Observers for Robust Nonlinear Estimation

Objectives

This chapter introduces a robust hybrid state estimation framework for vehicle motion tracking that combines physics-based guarantees with data-driven adaptability. Standard model-based observers often fail when strict rank conditions are not met, while pure neural network approaches lack stability guarantees. To address this, we propose a two-stage architecture. First, we develop a Generalized Unknown Input Observer (UIO) that utilizes output derivatives to relax the standard rank condition, ensuring the existence of a bounded estimation of unmodeled dynamics. Second, we introduce a Neural Adaptive Observer that treats the UIO estimates as a "teacher" signal. Unlike the UIO, which provides instantaneous signal estimation, the neural observer learns the underlying structure of the uncertainty as a function of the state. This allows the system to predict disturbances along reference trajectories and refine the estimation error asymptotically. The approach is validated via Linear Matrix Inequalities (LMIs) ensuring $\mathcal{H}^1$ and $\mathcal{L}_2$ stability.

Contents

1 Introduction

State estimation is a cornerstone of autonomous vehicle control, where safety depends on accurate knowledge of system states and external disturbances. Traditional methods rely on explicit physics-based models [**State_Esti2017**]. However, in vehicle dynamics, highly nonlinear phenomena-such as variable tire-road friction or aerodynamic drag are difficult to model mathematically, leading to estimation errors that compromise control performance.

Data-driven methods, particularly Neural Networks (NNs), have emerged as powerful tools for approximating these complex nonlinear functions [**2006Apdat**; **HO_GP**]. While NNs offer superior adaptability, they pose significant risks in safety-critical applications. In online learning scenarios, small shifts in input data can cause NNs to behave unpredictably or diverge, lacking the stability guarantees provided by control theory. Consequently, recent research has shifted toward hybrid architectures that embed learning within observer frameworks.

A major limitation in existing hybrid observers is their reliance on the standard observer matching condition. For many vehicle systems, this condition does not hold, making it theoretically impossible to decouple the unknown input from the state using standard techniques. While some methods attempt to bypass this using approximations, they often result in unbounded errors or require expensive additional sensors.

To overcome these limitations, this paper proposes a layered Teacher-Student estimation architecture consisting of two distinct observers:

- A Generalized UIO (The Teacher): We introduce a regularization technique using output derivatives that ensures the observer exists even when the rank condition fails. This observer provides a robust, physics-guaranteed signal of the unknown input, ensuring the estimation error remains bounded.
- A Neural Adaptive Observer (The Student): While the UIO estimates the current value of the disturbance, it cannot predict how that disturbance behaves at future states (e.g., along a reference trajectory). The Neural Observer utilizes the UIO's signal to learn the unknown dynamics as a function of the state ($\mu_\theta(x)$). This allows for refined precision and enables the controller to anticipate unmodeled dynamics.

This combination offers a specific advantage: the Generalized UIO guarantees the system never diverges (safety), while the Neural Network minimizes the residual error over time (performance). The contributions of this paper are:

- A Generalized UIO design that relaxes the rank condition using output derivatives ;
- A hybrid learning framework where the UIO acts as a supervisor, enforcing consistency in the neural training loop;
- LMI-based stability conditions that guarantee joint convergence of both the state and model parameters.

2 Problem Formulation and Motivation

In this paper, we consider the class of continuous-time nonlinear systems in state-space form described by the following equations:

$$\begin{aligned} \dot{x} &= \varphi(x,u) + B\mu_x \\ y &= Cx, \end{aligned} \tag{3.1}$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ is the state vector, $y \in \mathbb{R}^{n_y}$ is the output measurement, $u \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ is the control input, and $\mu_x \in \mathbb{R}^{n_\mu}$ is an unknown nonlinear function that encapsulates the system uncertainty including both unmodeled dynamics and external disturbances. The matrices $C \in \mathbb{R}^{n_y} \times \mathbb{R}^{n_x}$ and $B \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_\mu}$ are constant and known matrices. The function $\varphi : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^{n_x}$ represents the known nonlinear component of the system.

Remark 3. *For simplicity, we consider systems with linear output. Otherwise, if we have nonlinear output $y = h(x)$, we introduce a new state $\eta(t)$, with $\dot{\eta} \triangleq A_\eta \eta + \gamma y$ and $\eta_0 = \eta(0)$ known. Then, a new system with the augmented vector $\zeta = \begin{bmatrix} \eta \\ x \end{bmatrix}$ may be considered, which has as linear output the vector η.*

Now, let us introduce the following assumptions which are required for our unknown input observer design methods.

Assumption 2. *The output vector y has dimension greater than or equal to the number of unknown terms μ_x; equivalently, $n_\mu \leq n_y$.*

Assumption 3. *The nonlinear function $\varphi(.)$ is $\gamma_\varphi-$Lipschitz continuous on $\mathcal{X}$, uniformly on $u \in \mathcal{U}$.*

Assumption 4. *The unknown nonlinearity $\mu_x(.)$ is Lebesgue measurable and $\mathcal{L}_2-$bounded.*

Remark 4. *Assumption 4 corresponds to a* bounded-energy *uncertainty over $\Omega = [0, +\infty[$, which is standard in $\mathcal{H}_\infty$/ $\mathcal{L}_2$-gain observer analysis. In the vehicle context, μ_x may represent aggregated effects such as unmodeled tire-force nonlinearities, residual parameter mismatches, aerodynamic disturbances, or actuator/sensor fault components. Moreover, since the regularization introduced later contains $\omega_{\delta_1}(t) = \delta_1 \dot{\mu}_x(t)$, the subsequent $\mathcal{H}^1$ performance bound implicitly requires $\mu_x \in \mathcal{H}^1(\Omega)$ (i.e., $\mu_x \in \mathcal{L}_2(\Omega)$ and $\dot{\mu}_x \in \mathcal{L}_2(\Omega)$), which can be interpreted as* bounded-energy variations *of the unknown input.*

The objective of this paper is to estimate the unmodeled dynamics μ_x. More specifically, we propose to estimate μ_x by combining an unknown input observer (UIO) with a neural network–based approach, leveraging gradient-based methods to better capture the unmodeled components of the system. The key idea is to incorporate the UIO-based estimation into the loss function of the neural network, thereby enhancing residual regularization and ensuring consistency with the UIO. This, in turn, improves the robustness and performance of the overall estimation scheme. Unfortunately, it is not possible to estimate the unknown input, μ_x, if the system fails to satisfy the following main necessary condition:

$$\text{rank}(CB) = \text{rank}(B). \tag{3.2}$$

In addition, to be able to directly construct a UIO for system (3.1) to estimate simultaneously x and μ_x, the constraint (3.2) is not sufficient. For this, we should be able to write the system under the form:

$$\begin{aligned} \mathbb{E}\dot{\zeta} &= \varphi_\zeta(\zeta, u) \\ y &= \mathcal{H}\zeta \end{aligned} \tag{3.3}$$

with

$$\text{rank}\left(\begin{bmatrix}\mathbb{E} \\ \mathcal{H}\end{bmatrix}\right) = n_x + n_\mu. \tag{3.4}$$

For further details on this requirement, the reader is referred to [Chaouche, 2022; Trinh, 2011; Zemouche, 2009; Hassan, 2013; Trinh, 2008] and the references therein.

To address this challenge, various techniques have been proposed in the literature, each offering a different approach to the problem. Without aiming for exhaustiveness, these techniques can be summarized as follows. Nevertheless, the problem remains open, and new ideas or improvements are still possible, although often at the cost of additional assumptions:

- *Inverting the system dynamics:* The system state is first estimated, and the unknown inputs are then obtained by inverting the dynamics [Phanomchoeng, 2014; Benallouch, 2017; Charandabi, 2014]. However, the presence of disturbances and nonlinearities in the output signal or in the dynamics can make the problem particularly challenging.

- *Deriving the output vector $y(t)$:* In some cases, the derivatives of the output measurements are required [Hou, 1992; Phanomchoeng, 2014]. This leads to the construction of a pseudo-output vector y_{new}, which enables the simultaneous estimation of x and μ_x. For instance, in the linear case without disturbances, one obtains

  $$y_{\text{new}} = C_{\text{new}}x + C_\mu\mu_x,$$

  where $\text{rank}(C_\mu) = n_\mu$. However, this approach is not suitable for nonlinear disturbed systems, since differentiating $y(t)$ introduces additional variables arising from the derivatives of the disturbance vector, as well as new nonlinearities in the pseudo-output vector. Consequently, satisfying the rank condition (3.2) with the new matrices becomes very difficult.

- *Adding sensors:* In some cases, augmenting the system with additional sensors is a natural way to overcome the rank condition (3.2). This approach introduces a new measurement vector y_{new}, where the original output $y(t)$ is used to estimate the state $x(t)$, and the additional outputs y_{new} are exploited to estimate the unknown input, μ_x. For example, in [Phanomchoeng, 2014] (and references therein), four sensors (i.e., four output measurements) were employed to estimate only two unknown inputs. However, the presence of disturbances can significantly decrease estimation performance. Moreover, additional sensors may be very expensive or, in some cases, unavailable.

- *Proportional–Integral Observer and Null Dynamics:* Some works in the literature have employed a Proportional–Integral Observer (PIO) together with null dynamics imposed on the estimated unknown input [**PIO_1995**; **PIOreview**]. However, this assumption does not reflect the true

dynamics of the unknown input, leading to inaccurate estimation. Other approaches assume constant inputs [Jeon, 2020], i.e., $\dot{\mu}_x \equiv 0$, but this restriction is overly conservative. Recently, a method for discrete-time systems was proposed in [Nguyen, 2024b]; however, this approach yields delayed estimates of both the system state and the unknown inputs, which is often impractical to design reliable controllers.

The limitations of the aforementioned methods motivated us to adopt a different approach and propose a simple strategy to estimate the system state and the unknown input without relying on conservative assumptions. The key idea is to recognize that this initial estimation is not final. Rather, the objective is to provide a first-level approximation of x and μ_x, which will subsequently be refined through the neuro-adaptive observer we introduce later. This refinement enhances estimation accuracy and ultimately yields a definitive estimate of the unmodeled nonlinearity, μ_x.

The strategy consists in rewriting system (3.1) without modifying it. Then, the UIO we will propose will be based on reliable approximation of (3.1).

$$\begin{aligned} \dot{x} + E_{\delta_1}\dot{\mu}_x &= \varphi(x,u) + B\mu_x + E\omega_{\delta_1} \\ y &= Cx + D_{\delta_2}\mu_x + D\,\omega_{\delta_2} \end{aligned} \tag{3.5}$$

where $D \in \mathbb{R}^{n_y \times n_\mu}$ and $E \in \mathbb{R}^{n_x \times n_\mu}$ are given constant, full-column-rank matrix specified by the user; $E_{\delta_1} = \delta_1 E, D_{\delta_2} = \delta_2 D$; and $\omega_{\delta_1}(t) = \delta_1 \dot{\mu}_x(t), \omega_{\delta_2}(t) = \delta_2 \mu_x(t)$, with $\delta_j \geq 0$, with $(\delta_1, \delta_2) \neq (0,0)$, chosen sufficiently small to ensure the feasibility of some sufficient conditions that will be stated later. The purpose of this reformulation is to recover the rank condition (3.4), at the expense of introducing the additive terms $\omega_{\delta_j}(t), j = 1,2$, which are handled as disturbances.

- **Design guideline for E and D:** in practice, E and D can be selected as sparse (often binary) full-column-rank matrices that *inject* the regularization terms into a subset of state/output channels so that the augmented pair $(\mathbb{E}, \mathcal{H})$ satisfies Assumption 5. A simple procedure is to (i) choose D so that $\mathrm{rank}(D) = n_\mu$ (typically by selecting n_μ independent measured outputs), and then (ii) choose E (possibly also sparse/binary) to complete the rank condition (3.4) with minimal additional coupling.

- **Interpretation of δ_1, δ_2:** the scalars δ_1 and δ_2 quantify the trade-off between *rank recovery* (feasibility of Assumption 5) and *disturbance amplification* through the induced terms ω_{δ_1} and ω_{δ_2}.

The next section presents LMI conditions that guarantee accurate estimation of x and μ_x while satisfying an $\mathcal{H}_\infty$ criterion, or more specifically, an $\mathcal{L}_2$-optimality criterion—where the performance bound is proportional to $\delta_j, j = 1,2$.

3 LMI-Based $\mathcal{H}^1$ UIO Design

This section is devoted to the numerical LMI-based design procedure ensuring the asymptotic estimation of the system state x and the unmodeled dynamics μ_x following a $\mathcal{H}^1-$optimality criterion.

3.1 System transformation

Recall that $\mathcal{H}^1$ is the *Hilbert* space corresponding to the *Sobolev* space, $\mathcal{W}^{1,2}$, of square–integrable functions whose weak first derivatives are also square–integrable. Mathematically, we define the set $\mathcal{H}^1(\Omega)$ as

$$\mathcal{H}^1(\Omega) := \left\{ \psi \in \mathcal{L}^2(\Omega) \;\middle|\; \frac{\mathrm{d}\psi}{\mathrm{d}t} \in \mathcal{L}^2(\Omega) \right\}$$

with $\Omega = [0 \ +\infty[$ in our case. This space is endowed by the norm $\|.\|_{\mathcal{H}^1}$ defined as follows:

$$\|\psi\|_{\mathcal{H}^1} := \left(\|\psi\|^2_{\mathcal{L}^2} + \left\| \frac{\mathrm{d}\psi}{\mathrm{d}t} \right\|^2_{\mathcal{L}^2} \right)^{1/2}.$$

First, system (3.5) can be rewritten under the compact form:

$$\begin{aligned} \mathbb{E}\dot{\zeta} &= \varphi_\zeta(\zeta, u) + \bar{E}\boldsymbol{\omega}_\delta \\ y &= \mathcal{H}\zeta + \bar{D}\boldsymbol{\omega}_\delta \end{aligned} \tag{3.6}$$

where

$$\mathbb{E} := \begin{bmatrix} \mathbb{I}_{n_x} & E_{\delta_1} \end{bmatrix}, \ \mathcal{H} := \begin{bmatrix} C & D_{\delta_2} \end{bmatrix}, \tag{3.7a}$$

$$\bar{E} := \begin{bmatrix} E & 0_{n_x \times n_\mu} \end{bmatrix}, \ \bar{D} := \begin{bmatrix} 0_{n_y \times n_\mu} & D \end{bmatrix}, \tag{3.7b}$$

$$\zeta := \begin{bmatrix} x \\ \mu_x \end{bmatrix}, \ \boldsymbol{\omega}_\delta := \begin{bmatrix} \omega_{\delta_1} \\ \omega_{\delta_2} \end{bmatrix}, \tag{3.7c}$$

$$\varphi_\zeta(\zeta, u) := \varphi(x, u) + B\mu_x. \tag{3.7d}$$

Now, let us introduce the following necessary assumption:

Assumption 5. *The matrices E and D are chosen such that $\mathbb{E}$ and $\mathcal{H}$ satisfy the rank condition (3.4).*

The objective is to design a state observer that provides an estimate $\hat{\zeta}$ of ζ, such that the following $\mathcal{H}^1$-optimality inequality holds:

$$\|\tilde{\zeta}\|_{\mathcal{H}^1} \leq \lambda_\delta \ \|\mu_x\|_{\mathcal{H}^1} + \beta \ \|\xi_0\| \tag{3.8}$$

where $\tilde{\zeta} := \zeta - \hat{\zeta}$, λ_δ is the disturbance attenuation level, explicitly depending on δ_1 and δ_2, $\beta > 0$ is a weighting real constant, and ξ_0 is a vector depending on $\tilde{\zeta}(0)$ and $\mu_x(0)$.

3.2 UIO structure and error dynamics

From Assumption 5, there exist two matrices P_ζ and Q_ζ such that

$$P_\zeta \mathbb{E} + Q_\zeta \mathcal{H} = \mathbb{I}_{n_\zeta}. \tag{3.9}$$

where $n_\zeta := n_x + n_\mu$. These matrices are exploited in the following observer structure:

$$\begin{cases} \dot{\eta} &= P_\zeta \varphi_\zeta(\hat{\zeta}, u) + K\left(y - \mathcal{H}\hat{\zeta}\right) \\ \hat{\zeta} &= \eta + Q_\zeta y \end{cases} \tag{3.10}$$

where $\hat{\zeta}$ is the estimate of ζ and $K \in \mathbb{R}^{n_\zeta} \times \mathbb{R}^{n_y}$ is the observer gain to be determined such that the estimation error, $\tilde{\zeta} := \zeta - \hat{\zeta}$, satisfies the $\mathcal{H}^1$-optimality criterion (3.8) with appropriate λ_δ, β, and ξ_0.

We have

$$\begin{aligned} \tilde{\zeta} &= \zeta - \eta - Q_\zeta y \\ &= \left(\mathbb{I}_{n_\zeta} - Q_\zeta \mathcal{H}\right)\zeta - \eta - Q_\zeta \bar{D}\boldsymbol{\omega}_\delta \\ &\overset{(3.9)}{=\!=} P_\zeta \mathbb{E}\zeta - \eta - Q_\zeta \bar{D}\boldsymbol{\omega}_\delta \end{aligned} \tag{3.11}$$

which gives equivalently

$$\overbrace{\tilde{\zeta} + Q_\zeta \bar{D}\boldsymbol{\omega}_\delta}^{\xi(t)} = P_\zeta \mathbb{E}\zeta(t) - \eta(t). \tag{3.12}$$

It is worth noting that the reformulation (3.12) conveniently avoids the derivative of $\boldsymbol{\omega}_\delta$ in the derivation of the error dynamics.

By using the dynamics (3.6) and (3.10), we obtain

$$\begin{aligned} \dot{\xi} &= P_\zeta\left[\varphi_\zeta(\zeta, u) - \varphi_\zeta(\hat{\zeta}, u)\right] - K\mathcal{H}\tilde{\zeta} \\ &\quad + \left(P_\zeta \bar{E} - K\bar{D}\right)\boldsymbol{\omega}_\delta \\ &= \left(P_\zeta \nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z}, u) - K\mathcal{H}\right)\xi \\ &\quad + \left(\mathcal{E}(\boldsymbol{z}, u) - K\mathcal{D}\right)\boldsymbol{\omega}_\delta \end{aligned} \tag{3.13}$$

where

$$\mathcal{E}(\boldsymbol{z}, u) := \bar{E} - P_\zeta \nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z}, u) Q_\zeta \bar{D}$$

$$\mathcal{D} := \left(\mathbb{I}_{n_y} + \mathcal{H}Q_\zeta\right)\bar{D}.$$

and from the differential mean value theorem [**Hasni_LCSS_24**], we have

$$\varphi_\zeta(\zeta, u) - \varphi_\zeta\left(\hat{\zeta}, u\right) = \nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z}, u)\tilde{\zeta}$$

for a given $\boldsymbol{z}$.

From this point onward, the analysis will be carried out with the vector ξ instead of $\tilde{\zeta}$, and we will return to $\tilde{\zeta}$ at the end to derive the final $\mathcal{H}^1$ bound (3.8).

3.3 New LMI-based UIO design conditions

Before presenting the proposed LMI-based UIO design procedure, we state an intermediate and general result in the following proposition.

Proposition 1. *Le $\vartheta(.)$ be a Lyapunov function such that there exists $\vartheta_{\max} > 0$ such that $\vartheta(z) \leq \vartheta_{\max}\|z\|^2$, for all $z \in \mathbb{R}^{n_\zeta}$. Consider a matrix $\mathcal{S} = \mathcal{S}^\top > 0$ and a real constant $\lambda > 0$ such that the following inequality holds:*

$$\dot{\vartheta}(z) + \|z\|^2 + \dot{z}^\top \mathcal{S}\dot{z} - \lambda\|w\|^2 \leq 0, \forall z \in \mathbb{R}^{n_\zeta}, w \in \mathbb{R}^{2n_\mu} \tag{3.14}$$

with $z \in \mathcal{H}^1$ and $w \in \mathcal{L}^2$. Then, the following inequality is satisfied:

$$\|z\|_{\mathcal{H}^1} \leq \sqrt{\frac{\lambda}{\min\left(1, \lambda_{\min}(\mathcal{S})\right)}}\|w\|_{\mathcal{L}^2} + \sqrt{\frac{\vartheta_{\max}}{\min\left(1, \lambda_{\min}(\mathcal{S})\right)}}\|z_0\|. \tag{3.15}$$

Moreover, if (3.14) is satisfied with $z := \xi$ and $w := \boldsymbol{\omega}_\delta$, where ξ and $\boldsymbol{\omega}_\delta$ are defined by (3.12) and (3.7c), respectively, then the estimation error, $\tilde{\zeta}$, satisfies the $\mathcal{H}^1$ bound (3.8) with λ_δ, β, and ξ_0 given as follows:

$$\lambda_\delta := \delta_2 \sigma_{\max}\left(Q_\zeta D\right) + \max\left(\delta_1, \delta_2\right)\sqrt{\frac{\lambda}{\min\left(1, \lambda_{\min}(\mathcal{S})\right)}} \tag{3.16a}$$

$$\beta := \sqrt{\frac{\vartheta_{\max}}{\min\left(1, \lambda_{\min}(\mathcal{S})\right)}} \tag{3.16b}$$

$$\xi_0 := \tilde{\zeta}(0) + Q_\zeta D \mu_x(0). \tag{3.16c}$$

where $\sigma_{\max}\left(Q_\zeta D\right)$ represents the largest singular value of the matrix $Q_\zeta D$.

Proof. The first part of the proof is relatively straightforward. Before integrating (3.14) to get the norms $\|.\|_{\mathcal{H}^1}$ and $\|.\|_{\mathcal{L}^2}$, take in mind that

$$\min\left(1, \lambda_{\min}(\mathcal{S})\right)\left(\|z\|^2 + \|\dot{z}\|^2\right) \leq \|z\|^2 + \dot{z}^\top \mathcal{S}\dot{z}.$$

Hence, since

$$\int_0^{+\infty}\left(\|z(s)\|^2 + \|\dot{z}(s)\|^2\right)\mathrm{d}s \;=\; \|z\|^2_{\mathcal{H}^1},$$

$$\int_0^{+\infty}\|w(s)\|^2\,\mathrm{d}s \;=\; \|w\|^2_{\mathcal{L}^2},$$

and because $\vartheta(z(t)) \geq 0$ for all $t \geq 0$, implying that

$$\int_0^{+\infty}\vartheta(z(s))\,\mathrm{d}s \;\geq\; -\vartheta(z(0)),$$

we can readily conclude (3.15).

As for the second part of the proof, we will exploit the structure of ξ and $\boldsymbol{\omega}_\delta$. First, ξ and $\boldsymbol{\omega}_\delta$ satisfy (3.15)

proved in the first part. In addition, we have

$$\begin{aligned}\|\tilde{\zeta}\|_{\mathcal{H}^1} &= \left\|\xi - Q_\zeta \bar{D}\boldsymbol{\omega}_\delta\right\|_{\mathcal{H}^1} \\ &\leq \|\xi\|_{\mathcal{H}^1} + \left\|Q_\zeta \bar{D}\boldsymbol{\omega}_\delta\right\|_{\mathcal{H}^1} \\ &= \|\xi\|_{\mathcal{H}^1} + \left\|\begin{bmatrix}0 & Q_\zeta D\end{bmatrix}\boldsymbol{\omega}_\delta\right\|_{\mathcal{H}^1} \\ &= \|\xi\|_{\mathcal{H}^1} + \delta_2 \left\|Q_\zeta D\mu_x\right\|_{\mathcal{H}^1} \\ &\leq \|\xi\|_{\mathcal{H}^1} + \delta_2 \sigma_{\max}\left(Q_\zeta D\right) \|\mu_x\|_{\mathcal{H}^1} \end{aligned} \tag{3.17}$$

and

$$\begin{aligned}\|\boldsymbol{\omega}_\delta\|_{\mathcal{L}^2} &= \sqrt{\delta_1^2 \|\dot{\mu}_x\|_{\mathcal{L}^2}^2 + \delta_2^2 \|\mu_x\|_{\mathcal{L}^2}^2} \\ &\leq \max\left(\delta_1, \delta_2\right) \sqrt{\|\dot{\mu}_x\|_{\mathcal{L}^2}^2 + \|\mu_x\|_{\mathcal{L}^2}^2} \\ &= \max\left(\delta_1, \delta_2\right) \|\mu_x\|_{\mathcal{H}^1} . \end{aligned} \tag{3.18}$$

Hence, by substituting (3.17) and (3.18) in (3.15), the bound (3.8) is inferred with the parameters given in (3.16a)–(3.16c). □

Before stating the main theorem, notice that from Assumption 3, there exist constant matrices $\mathcal{A}_j \in \mathbb{R}^{n_\zeta \times n_\zeta}$ and functions $\alpha_j(\boldsymbol{z})$, $j = 1, \ldots \bar{n}_\zeta$ such that the Jacobian $\nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z}, u)$ belongs to the convex polytopic set defined as: This polytopic embedding is often less conservative than working with a single global Lipschitz constant, since it captures the variation of the Jacobian through a convex hull of vertex matrices and allows the LMI conditions to be enforced only at the vertices.

$$\mathcal{H}_\varphi \triangleq \left\{ \sum_{j=1}^{\bar{n}_\zeta} \alpha_j(\boldsymbol{z})\mathcal{A}_j, \sum_{j=1}^{\bar{n}_\zeta} \alpha_j(\boldsymbol{z}) = 1, \alpha_j(\boldsymbol{z}) \geq 0 \right\} \tag{3.19}$$

where $\mathcal{A}_j \in \mathbb{R}^{n_\zeta \times n_\zeta}$, are the vertices of the polytope $\mathcal{H}_\varphi$.

Now, we are ready to state the main theorem, which provides new LMI conditions ensuring the bound (3.8).

Theorem 2. *Suppose that Assumptions 2, 3, and 4 hold. Let the matrices E and D, together with the scalars δ_1 and δ_2, be chosen such that the Assumption 5 is satisfied. For a given constant $\epsilon > 0$, assume further that there exist a symmetric positive definite matrix $\mathcal{P} \in \mathbb{R}^{n_\zeta \times n_\zeta}$ and a matrix $\mathcal{Q} \in \mathbb{R}^{n_y \times n_\zeta}$ such that the LMI conditions (3.26) are fulfilled. Then, the estimation error $\tilde{\zeta}$, with $K = \mathcal{P}^{-1}\mathcal{Q}^\top$, satisfies the $\mathcal{H}^1$-optimality bound (3.8) with the parameters given in (3.16a)–(3.16c), where $\vartheta_{\max} = \lambda_{\max}(\mathcal{P})$ and $\mathcal{S} = \epsilon\mathcal{P}$ with $\lambda_{\min}(\mathcal{S}) = \epsilon\,\lambda_{\min}(\mathcal{P})$.*

Roadmap of the proof: the proof proceeds in three steps. First, a quadratic Lyapunov analysis is performed on the error dynamics to obtain a differential inequality of the form (3.14). Second, a Schur-complement (Schur lemma) argument is used to express this inequality as a matrix negativity condition. Third, convexification is applied by combining a change of variables with the polytopic representation (3.19), yielding vertex-wise LMIs.

Proof. Consider the quadratic Lyapunov function $\vartheta(\xi) := \xi^\top \mathcal{P}\xi$. Then, the derivative of $\vartheta(.)$ along the trajectories of (3.13) is given as follows:

$$\dot{\vartheta}(\xi) = \xi^\top \left[\left(P_\zeta \nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z}, u) - K\mathcal{H} \right)^\top \mathcal{P} + \mathcal{P} \left(P_\zeta \nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z}, u) - K\mathcal{H} \right) \right] \xi + \xi^\top \mathcal{P} \Big(\mathcal{E}\left(\boldsymbol{z}, u\right) - K\mathcal{D} \Big) \boldsymbol{\omega}_\delta + \boldsymbol{\omega}_\delta^\top \Big(\mathcal{E}\left(\boldsymbol{z}, u\right) - K\mathcal{D} \Big)^\top \mathcal{P}\xi. \quad (3.20)$$

and

$$\dot{\xi}^\top \mathcal{S}\dot{\xi} = \xi^\top \left(P_\zeta \nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z}, u) - K\mathcal{H} \right)^\top \mathcal{S} \times \left(P_\zeta \nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z}, u) - K\mathcal{H} \right) \xi + \boldsymbol{\omega}_\delta^\top \Big(\mathcal{E}\left(\boldsymbol{z}, u\right) - K\mathcal{D} \Big)^\top \mathcal{S} \Big(\mathcal{E}\left(\boldsymbol{z}, u\right) - K\mathcal{D} \Big) \boldsymbol{\omega}_\delta + 2\xi^\top \left(P_\zeta \nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z}, u) - K\mathcal{H} \right)^\top \mathcal{S} \Big(\mathcal{E}\left(\boldsymbol{z}, u\right) - K\mathcal{D} \Big) \boldsymbol{\omega}_\delta. \quad (3.21)$$

Then, we have

$$\dot{\vartheta}(\xi) + \|\xi\|^2 + \dot{\xi}^\top \mathcal{S}\dot{\xi} - \lambda \|\boldsymbol{\omega}_\delta\|^2 = \begin{bmatrix} \xi \\ \boldsymbol{\omega}_\delta \end{bmatrix}^\top \mathbb{M}\left(z, u\right) \begin{bmatrix} \xi \\ \boldsymbol{\omega}_\delta \end{bmatrix} \quad (3.22)$$

where $\mathbb{M}\left(z, u\right)$ is defined in (3.24). By applying Schur lemma, we deduce that $\mathbb{M}(z, u) < 0$ whenever inequality (3.25) holds. Consequently, by setting $\mathcal{S} = \epsilon\mathcal{P}$, introducing the change of variables $\mathcal{Q} := K^\top \mathcal{P}$, and invoking the convexity principle, we conclude that the inequality $\mathbb{M}(z, u) < 0$ is satisfied provided that the LMI conditions (3.26) hold. This, in turn, yields

$$\dot{\vartheta}(\xi) + \|\xi\|^2 + \dot{\xi}^\top \mathcal{S}\dot{\xi} - \lambda \|\boldsymbol{\omega}_\delta\|^2 \leq 0. \quad (3.23)$$

Hence, Proposition 1 applies and the desired result follows.

□

Remark 5. *It is worth emphasizing that the choice $\mathcal{S} = \epsilon\mathcal{P}$ is neither arbitrary nor overly restrictive, although it represents a specific selection. The introduction of the matrix $\mathcal{P}$ serves to recover the decision variable $\mathcal{Q}$ and then to avoid bilinear matrix inequalities, which are generally unsuitable for numerical solvers. The scalar parameter ϵ is included as a feasibility factor to improve the feasibility of the LMI conditions (3.26), and it should be chosen a priori.*

$$\begin{bmatrix} \left(P_\zeta \nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z},u) - K\mathcal{H}\right)^\top \mathcal{P} + \mathcal{P}\left(P_\zeta \nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z},u) - K\mathcal{H}\right) & \left[\mathcal{P} + \left(P_\zeta \nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z},u) - K\mathcal{H}\right)^\top \mathcal{S}\right]\left(\mathcal{E}(\boldsymbol{z},u) - K\mathcal{D}\right) \\ \left(\mathcal{E}(\boldsymbol{z},u) - K\mathcal{D}\right)^\top \left[\mathcal{P} + \mathcal{S}\left(P_\zeta \nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z},u) - K\mathcal{H}\right)\right] & -\lambda \mathbb{I}_{2n_\mu} + \left(\mathcal{E}(\boldsymbol{z},u) - K\mathcal{D}\right)^\top \mathcal{S}\left(\mathcal{E}(\boldsymbol{z},u) - K\mathcal{D}\right) \end{bmatrix} \tag{3.24}$$

$$\begin{bmatrix} \left(P_\zeta \nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z},u) - K\mathcal{H}\right)^\top \mathcal{P} + \mathcal{P}\left(P_\zeta \nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z},u) - K\mathcal{H}\right) & \mathcal{P}\left(\mathcal{E}(\boldsymbol{z},u) - K\mathcal{D}\right) & \left(P_\zeta \nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z},u) - K\mathcal{H}\right)^\top \mathcal{S} \\ \left(\mathcal{E}(\boldsymbol{z},u) - K\mathcal{D}\right)^\top \mathcal{P} & -\lambda \mathbb{I}_{2n_\mu} & \left(\mathcal{E}(\boldsymbol{z},u) - K\mathcal{D}\right)^\top \mathcal{S} \\ \mathcal{S}\left(P_\zeta \nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z},u) - K\mathcal{H}\right) & \mathcal{S}\left(\mathcal{E}(\boldsymbol{z},u) - K\mathcal{D}\right) & -\mathcal{S} \end{bmatrix} < 0 \tag{3.25}$$

$$\begin{bmatrix} (P_\zeta \mathcal{A}_j)^\top \mathcal{P} + \mathcal{P}(P_\zeta \mathcal{A}_j) - \mathcal{H}^\top \mathcal{Q} - \mathcal{Q}^\top \mathcal{H} & \mathcal{P}\mathcal{E}_j - \mathcal{Q}^\top \mathcal{D} & (P_\zeta \mathcal{A}_j)^\top \mathcal{P} - \mathcal{H}^\top \mathcal{Q} \\ \mathcal{E}_j^\top \mathcal{P} - \mathcal{D}^\top \mathcal{Q} & -\lambda \mathbb{I}_{2n_\mu} & \mathcal{E}_j^\top \mathcal{P} - \mathcal{D}^\top \mathcal{Q} \\ \mathcal{P}(P_\zeta \mathcal{A}_j) - \mathcal{Q}^\top \mathcal{H} & \mathcal{P}\mathcal{E}_j - \mathcal{Q}^\top \mathcal{D} & -\frac{1}{\epsilon}\mathcal{P} \end{bmatrix} < 0 \tag{3.26}$$

where

$$\mathcal{E}_j := \bar{E} - P_\zeta \mathcal{A}_j Q_\zeta \bar{D}$$

4 Output Derivative-Based Generalized UIO

The main limitation of the previous result is that the proposed method does not guarantee exponential convergence of the estimation error, even when condition (3.2) is satisfied. This represents a drawback of the proposed *regularization* technique and has motivated the development of a more general and robust method that overcomes this shortcoming.

4.1 Output derivative-based generalized model

The key idea consists in introducing a generalized system imbedding the original model, the output y and its derivative. To this end, let us consider $\chi_1 \triangleq y$ and $\chi_2 \triangleq \dot{y}$, which leads to the following generalized system:

$$\begin{cases} \dot{\chi}_1 &= \chi_2 \\ \dot{\chi}_2 &= C\bar{\varphi}_x(x,u) + C\frac{\partial \varphi}{\partial x}(x,u)B\mu_x + CB\dot{\mu}_x \\ \dot{x} &= \varphi(x,u) + B\mu_x \\ y_\chi &= \chi_1 + C_\chi \chi_2 + Cx \end{cases} \tag{3.27}$$

where $\bar{\varphi}_x(x,u) \triangleq \frac{\partial \varphi}{\partial x}(x,u)\varphi(x,u)$, and $C_\chi \in \mathbb{R}^{n_y \times n_y}$ is a given weighting matrix associated with the measured derivative $\dot{y}$.

Clearly, if (3.2) is satisfied, then (3.27) can be rewritten in the form of (3.6), for which Assumption 5 holds. In this case, an exponential UIO for (3.27) can be directly constructed if we add χ_2 as output measurement. However, if (3.2) is not satisfied, the construction of an exponential UIO is not possible. In this situation, we can proceed as in the previous section by first regularizing the system (3.27). This approach leads to a design method that is more general than that of the previous section, as it ensures exponential convergence of the UIO whenever condition (3.2) is satisfied and the derivative $\dot{y}$ is used as a measurement.

4.2 Regularization of the generalized model

System (3.27) can be regularized as follows

$$\begin{cases} \dot{\chi}_1 = & \chi_2 \\ \dot{\chi}_2 + (E_{\delta_1} - CB)\,\dot{\mu}_x = & C\bar{\varphi}_x(x,u) + E\,\omega_{\delta_1} + C\dfrac{\partial \varphi}{\partial x}(x,u)B\mu_x \\ \dot{x} = & \varphi(x,u) + B\mu_x \\ y_\chi = & \chi_1 + C_\chi\chi_2 + Cx + D_{\delta_2}\mu_x + D\,\omega_{\delta_2} \end{cases} \tag{3.28}$$

where the scalars $\delta_1 \geq 0$, $\delta_2 \geq 0$, and the matrices $E \in \mathbb{R}^{n_y \times n_\mu}$ and $D \in \mathbb{R}^{n_y \times n_\mu}$ are the *regularization parameters*, with E and D chosen to be binary, i.e., their entries are restricted to $\{0,1\}$. This choice is motivated by the aim of controlling the estimation error bound exclusively through the scalar parameters δ_1 and δ_2. Moreover, E and D are selected to satisfy an appropriate rank condition, which will be specified later.

Practical choice: choosing E and D binary amounts to selecting which measured channels are used to restore the rank condition while keeping the design interpretable and sparse. In practice, one may

start from a binary D with $\mathrm{rank}(D) = n_\mu$ (select n_μ independent outputs), and then adjust E (binary or structured) so that the resulting augmented matrices satisfy the required rank condition. A concrete selection of such a binary D is given in the numerical vehicle example (Subsubsection *UIO for the LPV Vehicle Model*).

First, system (3.5) can be rewritten under the compact form:

$$\begin{aligned} \mathbb{E}\dot{\zeta} &= \varphi_\zeta(\zeta, u) + \bar{E}\boldsymbol{\omega}_\delta \\ y_\chi &= \mathcal{H}\zeta + \bar{D}\boldsymbol{\omega}_\delta \end{aligned} \tag{3.29}$$

where

$$\mathbb{E} := \begin{bmatrix} \mathbb{I}_{n_y} & 0 & 0 & 0 \\ 0 & \mathbb{I}_{n_y} & 0 & -CB \\ 0 & 0 & \mathbb{I}_{n_x} & 0 \end{bmatrix}, \ \mathcal{H} := \begin{bmatrix} \mathbb{I}_{n_y} & C_\chi & C & D_{\delta_2} \end{bmatrix}, \tag{3.30a}$$

$$\bar{E} := \begin{bmatrix} 0 & 0_{n_y \times n_\mu} \\ E & 0_{n_y \times n_\mu} \\ 0 & 0_{n_x \times n_\mu} \end{bmatrix}, \ \bar{D} := \begin{bmatrix} 0_{n_y \times n_\mu} & D \end{bmatrix}, \tag{3.30b}$$

$$\zeta := \begin{bmatrix} \chi_1 \\ \chi_2 \\ x \\ \mu_x \end{bmatrix}, \ \boldsymbol{\omega}_\delta := \begin{bmatrix} \omega_{\delta_1} \\ \omega_{\delta_2} \end{bmatrix}, \tag{3.30c}$$

$$\varphi_\zeta(\zeta, u) := \begin{bmatrix} \chi_2 \\ C\bar{\varphi}_x(x, u) + C\frac{\partial \varphi}{\partial x}(x, u)B\mu_x \\ \varphi(x, u) + B\mu_x. \end{bmatrix} \tag{3.30d}$$

with E_{δ_1}, D_{δ_2}, ω_{δ_1}, and ω_{δ_2} defined as in (3.5).

4.3 Generalized UIO

Here we provide the generalized UIO corresponding to the generalized and regularized system (3.29).

Assumption 6. *The matrices E, D, and C_χ, and the scalars δ_1 and δ_2 are chosen such that $\mathbb{E}$ and $\mathcal{H}$ satisfy the rank condition (3.4).*

Assumption 7. *The function $\varphi_\zeta(., u)$ defined in (3.30d) is globally Lipschitz, uniformly on u.*

As in the previous section, from Assumption 6, there exist two matrices P_ζ and Q_ζ such that

$$P_\zeta \mathbb{E} + Q_\zeta \mathcal{H} = \mathbb{I}_{n_\zeta}. \tag{3.31}$$

where $n_\zeta := 2n_y + n_x + n_\mu$, and then an observer is constructed as in (3.10) as follows:

$$\begin{cases} \dot{\eta} &= P_\zeta \varphi_\zeta(\hat{\zeta}, u) + K\left(y_\chi - \mathcal{H}\hat{\zeta}\right) \\ \hat{\zeta} &= \eta + Q_\zeta y_\chi \end{cases} \tag{3.32}$$

To avoid repetitions, under Assumption 7, assume that the Jacobian $\nabla_\zeta^{\varphi_\zeta}(\boldsymbol{z}, u)$ belongs to the convex polytopic set $\mathcal{H}_\varphi$ defined in (3.19).

It is unnecessary to restate a new theorem or proposition here. Since we use the same notation as in Section 3, the results of Proposition 1 and Theorem 2 apply directly under Assumptions 6 and 7. Specifically, the observer (3.32) with $K = \mathcal{P}^{-1}\mathcal{Q}^\top$ satisfies the $\mathcal{H}^1$-optimality bound (3.8), with parameters given in (3.16a)–(3.16c), where $\vartheta_{\max} = \lambda_{\max}(\mathcal{P})$ and $\mathcal{S} = \epsilon\mathcal{P}$, with $\lambda_{\min}(\mathcal{S}) = \epsilon\,\lambda_{\min}(\mathcal{P})$. The matrices $\mathcal{P}$, $\mathcal{Q}$, and the scalar ϵ are defined in Theorem 2.

4.4 Specific cases and discussion

This section is devoted to discussing the general result established in Section 4. In addition, several specific cases are analyzed to illustrate the benefits of the generalized UIO in Section 4.3 relative to that of Section 3.

4.4.1 Case $\mathrm{rank}(CB) = \mathrm{rank}(B)$

We show that in this case, exponential convergence of the estimation error to zero can be guaranteed. Indeed, if $\mathrm{rank}(CB) = \mathrm{rank}(B)$, then to satisfy Assumption 6, it suffices to choose an appropriate matrix C_χ, independently of δ_1 and δ_2. We can then set $\delta_1 = \delta_2 = 0$, which implies $\lambda_\delta = 0$ in (3.16a) and $\boldsymbol{\omega}_\delta(t) \equiv 0$ in (3.23). Under these conditions, the bound (3.8) reduces to

$$\|\tilde{\zeta}\|_{\mathcal{H}^1} \leq \beta\,\|\xi_0\|, \tag{3.33}$$

which, combined with the uniform continuity of $\tilde{\zeta}(\cdot)$, implies

$$\lim_{t\to+\infty} \tilde{\zeta}(t) = 0.$$

In other words, we have $\xi(t) = \tilde{\zeta}(t)$, and the LMIs (3.26) ensure (3.23), which reduces to

$$\dot{\vartheta}(\tilde{\zeta}(t)) + \|\tilde{\zeta}(t)\|^2 \leq 0,$$

thus guaranteeing the *exponential convergence of* $\tilde{\zeta}(t)$ *to zero*.

4.4.2 Case $C_\chi = 0$

In this case, the derivative of the output measurement, $\dot{y}$, is not used as an additional input to the observer. Unfortunately, even when $\mathrm{rank}(CB) = \mathrm{rank}(B)$, the only way to satisfy Assumption 6 is to choose an appropriate nonzero matrix D and $\delta_2 > 0$. Consequently, we may set $\delta_1 = 0$ to achieve a tighter $\mathcal{H}^1$ bound; however, exponential convergence of the estimation error to zero cannot be guaranteed.

4.4.3 Case $\mathrm{rank}(CB) < \mathrm{rank}(B)$

In this case, exponential convergence of $\tilde{\zeta}$ to zero cannot be guaranteed; however, an $\mathcal{H}^1$ bound can still be obtained through regularization. By appropriately selecting the matrices E and D, and choosing the scalars δ_1 and δ_2 such that Assumption 6 is satisfied, the $\mathcal{H}^1$ bound (3.8) can be effectively adjusted by tuning the parameters δ_1 and δ_2.

4.4.4 On the use of $\dot{y}$

The generalized UIO (3.32) requires real-time knowledge of the derivative $\dot{y}$, which can be viewed as a drawback since this derivative must be estimated online. Nevertheless, the structure of the proposed observer naturally incorporates the estimation of $\dot{y}$ through the auxiliary state variable χ_2. In particular, when $\mathrm{rank}(CB) = \mathrm{rank}(B)$, the estimate $\hat{\chi}_2$ converges exponentially to χ_2, thereby providing an accurate estimation of $\dot{y}$. This represents a major improvement over conventional UIO design approaches, which typically estimate the unknown inputs by differentiating the estimated states or the measured output y a posteriori.

Table 3.1: Summary of specific cases: assumptions, design choices, and convergence properties.

Case / setting	**Design choice to satisfy Assumption 6**	**Main guarantee (what you get)**	**Use of $\dot{y}$**
$\mathrm{rank}(CB) = \mathrm{rank}(B)$ (classical matching holds)	Pick $C_\chi \neq 0$; can set $\delta_1 = \delta_2 = 0$ (no regularization, $\boldsymbol{\omega}_\delta \equiv 0$)	Exponential convergence of $\tilde{\zeta}(t)$ to 0 (strongest case)	Required by the generalized structure, but estimated internally via $\hat{\chi}_2$
$C_\chi = 0$ (do not inject output derivatives)	Must rely on regularization through $D \neq 0$ and $\delta_2 > 0$ (often take $\delta_1 = 0$ for a tighter bound)	Only an $\mathcal{H}^1$ bound; exponential convergence to 0 cannot be ensured	Not used
$\mathrm{rank}(CB) < \mathrm{rank}(B)$ (classical UIO fails)	Regularize: choose E, D, and $\delta_1, \delta_2 > 0$ so that the augmented pair $(\mathbb{E}, \mathcal{H})$ satisfies (3.4)	$\mathcal{H}^1$ bound with tunable attenuation level λ_δ (via δ_1, δ_2); exponential convergence not guaranteed	Typically used if $C_\chi \neq 0$; otherwise not needed

5 UIO-Based Neural Online Approximation of Unmodeled Nonlinearity μ_x

As discussed in the previous section, the UIO provides an estimation of the unmodeled nonlinearity μ_x. However, this estimation is instantaneous and does not provide a predictive model of the nonlinearity. To address this, we employ a neural network to learn the function μ_x explicitly. This learned model can then be used for control design or improved state estimation. The UIO scheme developed earlier serves as a reliable source of training data (ground truth proxy) for this online learning process.

Regarding the estimation of the system state x during learning, two approaches can be considered: (1) using the state estimate $\hat{x}$ directly from the existing UIO, or (2) designing a separate neuro-adaptive

observer that simultaneously estimates x and the neural network parameters. In this work, we focus on the first approach, where the UIO provides the state estimate used to query the neural network.

We begin by introducing the following assumption regarding the closed-loop system:

Assumption 8. *The system is capable of tracking a known reference trajectory x_{ref} under an observer-based controller of the form:*

$$u := \kappa\left(x_{ref}, u_{ref}, \hat{x}\right) \tag{3.34}$$

where $\hat{x}$ is the state estimate, and u_{ref} is a reference control input such that the pair (x_{ref}, u_{ref}) constitutes an admissible trajectory for the system (3.1).

5.1 Fully UIO-based Learning Architecture

The generalized UIO derived in Section 4 provides an augmented state estimate $\hat{\zeta}$. To facilitate the learning strategy, we partition this vector to isolate the physics-based state estimate, denoted as $\hat{x}_{\text{UIO}}$, and the unknown input estimate, $\hat{\mu}_x$:

$$\hat{\zeta} = \begin{bmatrix} \hat{\chi}_1 \\ \hat{\chi}_2 \\ \hat{x}_{\text{UIO}} \\ \hat{\mu}_x \end{bmatrix} \in \mathbb{R}^{2n_y+n_x+n_\mu}. \tag{3.35}$$

Using these estimates, we introduce a neural network approximation $\mu_\theta(\cdot)$ to capture the unmodeled dynamics. This data-driven model is integrated into a secondary observer structure, referred to as the Neural Observer, to refine the state estimation.

Let $\mu_\theta(\hat{x}_{\text{nn}})$ denote the neural network approximation of μ_x, parameterized by θ:

$$\mu_\theta(\hat{x}_{\text{nn}}) = \mathcal{N}_\theta(\hat{x}_{\text{nn}}, u) \tag{3.36}$$

where $\mathcal{N}_\theta(\cdot)$ is the neural network function. This approximation is incorporated into the observer dynamics as follows:

$$\begin{cases} \dot{\hat{x}}_{\text{nn}} = \varphi(\hat{x}_{\text{nn}}, u) + B\mu_\theta(\hat{x}_{\text{nn}}) + L_{\text{nn}}\left(y - y_{\text{nn}}\right) \\ y_{\text{nn}} = C\hat{x}_{\text{nn}} \end{cases} \tag{3.37}$$

where $\hat{x}_{\text{nn}} \in \mathbb{R}^{n_x}$ is the state of the Neural Observer and L_{nn} is the observer gain to be designed.

The proposed hybrid strategy is illustrated in Figure 3.1. The physics-based UIO uses the measured output y to provide robust initial estimates $\hat{x}_{\text{UIO}}$ and $\hat{\mu}_x$. Simultaneously, the Neural Observer utilizes the learned model μ_θ to produce a refined state estimate $\hat{x}_{\text{nn}}$. The neural network parameters θ are updated online by minimizing a composite loss function that enforces consistency between the data-driven model and the physics-based UIO estimates.

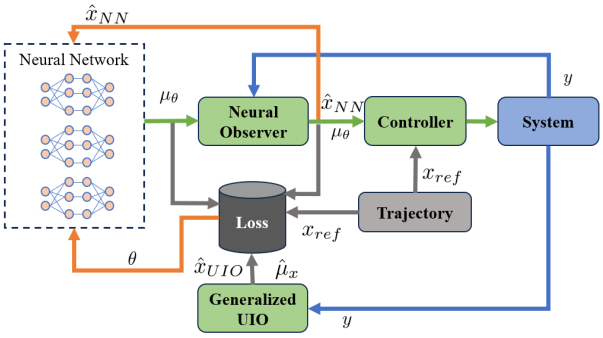


Figure 3.1: Diagram of the proposed hybrid estimation strategy.

The dynamics of the original system (3.1) can be rewritten as:

$$\dot{x} = \varphi(x, u) + B\mu_\theta(\hat{x}_{\text{nn}}) + B\Delta_x(t) \tag{3.38}$$

where $\Delta_x(t) \triangleq \mu_x(t) - \mu_\theta(\hat{x}_{\text{nn}}(t))$ represents the approximation error of the neural network.

5.1.1 Design of the Neural Observer

The objective is to determine the observer gain L_{nn} such that the estimation error $\epsilon_{\text{nn}} \triangleq x - \hat{x}_{\text{nn}}$ satisfies an $\mathcal{L}^2$ performance criterion:

$$\|\epsilon_{\text{nn}}\|_{\mathcal{L}^2} \leq \lambda_{\text{nn}} \, \|\Delta_x\|_{\mathcal{L}^2} + \beta_{\text{nn}} \, \|\epsilon_{\text{nn}}(0)\| \tag{3.39}$$

where λ_{nn} and β_{nn} are positive scalars. We consider the quadratic Lyapunov function $V(\epsilon_{\text{nn}}) = \epsilon_{\text{nn}}^\top \mathcal{P} \epsilon_{\text{nn}}$ with $\mathcal{P} > 0$. To satisfy (3.39), it suffices to ensure:

$$\dot{V}(\epsilon_{\text{nn}}) + \epsilon_{\text{nn}}^\top \epsilon_{\text{nn}} - \lambda_{\text{nn}}^2 \Delta_x^\top \Delta_x < 0.$$

The error dynamics are given by:

$$\dot{\epsilon}_{\text{nn}} = \left(\nabla_x^\varphi(\boldsymbol{z}_x, u) - L_{\text{nn}} C\right)\epsilon_{\text{nn}} + B\Delta_x(t) \tag{3.40}$$

where $\nabla_x^\varphi(\boldsymbol{z}_x, u)$ is the Jacobian of φ evaluated at some $\boldsymbol{z}_x$. Under Assumption 3, this Jacobian belongs to the convex polytope $\mathcal{H}_\varphi^x$ defined in (**??**).

The following theorem provides sufficient LMI conditions for the existence of such an observer.

Theorem 3. *Under Assumption 3, if there exist a symmetric positive definite matrix $\mathcal{P} \in \mathbb{R}^{n_x \times n_x}$, a matrix $\mathcal{Q} \in \mathbb{R}^{n_y \times n_x}$, and a positive scalar σ satisfying the following LMI conditions for all vertices $j = 1, \ldots, \bar{n}_x$:*

$$\begin{bmatrix} \left(\mathcal{A}_j^x\right)^\top \mathcal{P} + \mathcal{P}\mathcal{A}_j^x - C^\top \mathcal{Q} - \mathcal{Q}^\top C & \mathcal{P}B \\ B^\top \mathcal{P} & -\sigma \mathbb{I}_{n_\mu} \end{bmatrix} < 0 \tag{3.41}$$

then the observer gain $L_{nn} = \mathcal{P}^{-1}\mathcal{Q}^\top$ ensures that the estimation error ϵ_{nn} satisfies the $\mathcal{L}^2$ criterion (3.39) with $\lambda_{nn} = \sqrt{\sigma}$ and $\beta_{nn} = \sqrt{\lambda_{\max}(\mathcal{P})}$.

Proof. The proof follows standard $\mathcal{L}^2$ stability analysis for LPV systems and is omitted for brevity. □

5.1.2 Optimization of the Learning Parameter θ

To approximate the unknown dynamics $\mu_x(t)$, the neural network parameters θ are optimized to minimize a specific loss function. Since the true value of $\mu_x(t)$ is unknown, we utilize the UIO estimate $\hat{\mu}_x(t)$ as a target. We define a composite loss function that balances tracking performance, measurement consistency, and regularization:

$$\begin{aligned}\mathcal{L}(\theta) = &\underbrace{(\hat{x}_{\text{nn}} - x_{\text{ref}})^\top \mathcal{T}_{\text{ref}} (\hat{x}_{\text{nn}} - x_{\text{ref}})}_{\textit{Tracking Error}} + \underbrace{(y - y_{\text{nn}})^\top \mathcal{T}_y (y - y_{\text{nn}})}_{\textit{Output Error}} \\ &+ \underbrace{(\hat{x}_{\text{nn}} - \hat{x}_{\text{uio}})^\top \mathcal{T}_{\text{uio}} (\hat{x}_{\text{nn}} - \hat{x}_{\text{uio}})}_{\textit{UIO Consistency}} + \underbrace{\lambda \|\mu_\theta(\hat{x}_{\text{nn}}) - \hat{\mu}_x\|^2}_{\textit{Regularization}}\end{aligned} \tag{3.42}$$

where $\mathcal{T}_{\text{ref}}, \mathcal{T}_y, \mathcal{T}_{\text{uio}}$ are positive definite weighting matrices, and $\lambda > 0$ is a scalar weight.

The terms in (3.42) are motivated as follows:

- **Tracking Error:** Penalizes deviations of the neural observer state $\hat{x}_{\text{nn}}$ from the reference trajectory x_{ref}, ensuring the learned model supports the control objective.
- **Output Error:** Enforces consistency between the predicted output y_{nn} and the actual measurement y, providing direct feedback from sensor data.
- **UIO Consistency:** Constrains $\hat{x}_{\text{nn}}$ to remain close to the robust UIO estimate $\hat{x}_{\text{uio}}$, preventing divergence during the early phases of learning.
- **Regularization:** Directly guides the neural network output μ_θ towards the UIO-estimated disturbance $\hat{\mu}_x$, acting as a supervised learning signal.

5.2 Minimization of the Loss Function

This section details the numerical optimization procedure for tuning the neural network parameters θ. We employ a gradient-based approach using a discretized version of the neural observer and sensitivity propagation.

5.2.1 Discretization of the Neural Observer

Consider the continuous-time neural observer dynamics:

$$\dot{\hat{x}}_{\text{nn}}(t) = \varphi(\hat{x}_{\text{nn}}(t), u(t)) + B\,\mu_\theta(\hat{x}_{\text{nn}}(t)) + L_{\text{nn}}\big(y(t) - C\hat{x}_{\text{nn}}(t)\big).$$

Applying a forward Euler discretization with sampling period Δt, we obtain the discrete-time update law:

$$\hat{x}_{k+1} = \hat{x}_k + \Delta t\Big(\varphi(\hat{x}_k, u_k) + B\,\mu_\theta(\hat{x}_k) - L_{\text{nn}} C\hat{x}_k + L_{\text{nn}} y_k\Big).$$

This can be written compactly as:

$$\hat{x}_{k+1} = f_{\text{obs}}(\hat{x}_k, \theta, k). \tag{3.43}$$

5.2.2 Gradient Computation via Sensitivity Analysis

The optimal parameters θ are obtained by solving the following optimization problem over a horizon K:

$$\begin{aligned} \min_{\theta} \quad & J(\theta) = \textstyle\sum_{k=0}^{K} \mathcal{L}_k(\hat{x}_k(\theta)) \\ \text{s.t.} \quad & \hat{x}_{k+1} = f_{\text{obs}}(\hat{x}_k, \theta, k). \end{aligned} \tag{3.44}$$

Since the state $\hat{x}_k$ depends implicitly on θ through the observer dynamics, we compute the gradient $\nabla_\theta J$ using sensitivity analysis.

The total gradient of the loss at step k with respect to θ is given by the chain rule:

$$\nabla_\theta \mathcal{L}_k = \frac{\partial \mathcal{L}_k}{\partial \hat{x}_k} S_k + \frac{\partial \mathcal{L}_k}{\partial \mu_k} \left(\frac{\partial \mu_\theta}{\partial \theta} + \frac{\partial \mu_\theta}{\partial \hat{x}_k} S_k \right) \tag{3.45}$$

where $S_k \triangleq \frac{\partial \hat{x}_k}{\partial \theta} \in \mathbb{R}^{n_x \times n_\theta}$ is the sensitivity matrix, and $\mu_k \triangleq \mu_\theta(\hat{x}_k)$.

The partial derivatives of the loss function (3.42) are:

$$\frac{\partial \mathcal{L}_k}{\partial \hat{x}_k} = 2(\hat{x}_k - x_{\text{ref},k})^\top \mathcal{T}_{\text{ref}} - 2(y_k - C\hat{x}_k)^\top \mathcal{T}_y C + 2(\hat{x}_k - \hat{x}_{\text{UIO},k})^\top \mathcal{T}_{\text{uio}} \tag{3.46}$$

$$\frac{\partial \mathcal{L}_k}{\partial \mu_k} = 2\lambda(\mu_\theta(\hat{x}_k) - \hat{\mu}_{x,k})^\top. \tag{3.47}$$

The sensitivity matrix S_k propagates according to the linearized dynamics of the observer:

$$S_{k+1} = S_k + \Delta t \, (A_k S_k + B_k) \tag{3.48}$$

where

$$A_k = \frac{\partial f_{\text{obs}}}{\partial \hat{x}_k} = \frac{\partial \varphi}{\partial \hat{x}_k} + B \frac{\partial \mu_\theta}{\partial \hat{x}_k} - L_{\text{nn}} C \tag{3.49}$$

$$B_k = \frac{\partial f_{\text{obs}}}{\partial \theta} = B \frac{\partial \mu_\theta}{\partial \theta}. \tag{3.50}$$

The Jacobians $\frac{\partial \mu_\theta}{\partial \hat{x}_k}$ and $\frac{\partial \mu_\theta}{\partial \theta}$ are computed via automatic differentiation. The parameters are then updated using a gradient descent algorithm (e.g., Adam):

$$\theta \leftarrow \theta - \eta \sum_{k=0}^{K} \nabla_\theta \mathcal{L}_k.$$

Remark 6 (Learning stability). *Due to the nonlinear parameterization of deep neural networks, global convergence of the learning parameters θ cannot, in general, be guaranteed using classical adaptive control arguments. Nevertheless, the observer gain is synthesized such that the nominal estimation error dynamics are exponentially stable (i.e., when the approximation error is zero). The neural network enters the estimation error dynamics through the approximation error $\Delta_x(t) := \mu_x(t) - \mu_\theta(\hat{x}_{nn}(t))$, which acts as an additive disturbance. Therefore, the estimation error system is input-to-state stable with respect to Δ_x; in particular, if Δ_x remains bounded during online learning (e.g., by using bounded learning rates and, when needed in practice, weight constraints such as projection/clipping), then all internal estimation*

signals are uniformly ultimately bounded.

6 Continuous Learning Strategy

A critical challenge in online learning is the "catastrophic forgetting" phenomenon, where the model loses previously acquired knowledge when trained on new data. In our context, while the unmodeled dynamics μ_x may be state-dependent and time-invariant, the system operates in different regions of the state space over time. Training solely on recent data may degrade performance in previously visited regions.

To address this, we adopt an experience replay mechanism using a fixed-size queue of neural networks, as proposed in [**Jiahao2022OnlineDL**] and illustrated in Fig. 3.2. The strategy proceeds as follows:

1. A queue of size p maintains historical versions of the neural network model.
2. When new data is available, the current model is updated and added to the queue.
3. If the queue is full, the oldest model is discarded.
4. The effective model used for prediction, $\hat{f}_{\text{nn}}^{(i+1)}$, is a weighted ensemble of the models in the queue:

$$\hat{f}_{\text{nn}}^{(i+1)} = M_\psi \left(\hat{f}_{\text{nn}}^{i}, e^{i+1-p} f_\theta^{(i+1)} \right) \tag{3.51}$$

where $f_\theta^{(i+1)}$ is the $(i+1)$-th network added to the queue, and M_ψ is an aggregation function.

This approach ensures that the learned model retains information from past operating conditions while adapting to new data, thereby enhancing stability and robustness across the entire operating envelope.

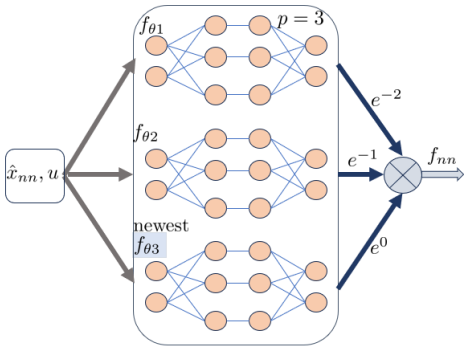


Figure 3.2: Schematic of the continuous learning with forgetting mechanism.

7 Simulation Results

In this section, we validate the proposed hybrid estimation framework through numerical simulations of an autonomous vehicle. The objective is to estimate the vehicle lateral dynamics and the unknown tire forces in the presence of parameter uncertainties and unmodeled nonlinearities.

7.1 Vehicle Lateral Dynamics Model

We consider a dynamic bicycle model to describe the vehicle lateral motion. Using Newton's second law, the equations of motion are given by [**LPV_vehicle**]:

$$\dot{v}_x = a - \frac{F_{yf}\sin(\delta)}{m} - \mu g + \dot{\psi} v_y \tag{3.52a}$$

$$\dot{v}_y = \frac{F_{yr}}{m} + \frac{F_{yf}\cos(\delta)}{m} - \dot{\psi} v_x \tag{3.52b}$$

$$\dot{\psi} = r \tag{3.52c}$$

$$\dot{r} = \frac{l_f F_{yf}\cos(\delta)}{I_z} - \frac{l_r F_{yr}}{I_z} \tag{3.52d}$$

where v_x and v_y are the longitudinal and lateral velocities at the center of gravity (CG), ψ is the yaw angle, and $r = \dot{\psi}$ is the yaw rate. The control inputs are the longitudinal acceleration a and the steering angle δ. The parameters m and I_z denote the vehicle mass and yaw moment of inertia, respectively. l_f and l_r are the distances from the CG to the front and rear axles.

The lateral tire forces F_{yf} and F_{yr} are modeled as a combination of a linear term (nominal model) and a nonlinear term (uncertainty):

$$F_{yf} = C_f \alpha_f + f_{yf}(\alpha_f), \quad F_{yr} = C_r \alpha_r + f_{yr}(\alpha_r) \tag{3.53}$$

where C_f and C_r are the nominal cornering stiffnesses. The tire slip angles are defined as:

$$\alpha_f = \delta - \frac{v_y + l_f r}{v_x}, \quad \alpha_r = -\frac{v_y - l_r r}{v_x}. \tag{3.54}$$

0) Augmented state, input, unknown

State order (8D):

$$x_8 = \begin{bmatrix} v_x & v_y & \psi & r & X & Y & a_x & a_y \end{bmatrix}$$

Input:

$$u = \begin{bmatrix} \delta & a \end{bmatrix}$$

Unknown residual (your NN/UIO output):

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} f_r(\alpha_r) \\ f_f(\alpha_f) \end{bmatrix}$$

Slip angles (same as you):

$$\alpha_f = \delta - \frac{v_y}{v_x} - \frac{l_f r}{v_x}, \qquad \alpha_r = -\frac{v_y}{v_x} + \frac{l_r r}{v_x}$$

Scheduling (qLPV) example:

$$\rho = \left[\frac{1}{v_x},\ \sin\delta,\ \cos\delta,\ v_x,\ v_y,\ \sin\psi,\ \cos\psi\right] \quad (v_x > 0)$$

1) The key "clean IMU" definition (why this is best for w)

Use **body acceleration** identities:

$$a_x^{\text{body}} = \dot{v}_x - r v_y, \qquad a_y^{\text{body}} = \dot{v}_y + r v_x$$

With your bicycle model, the (rv) terms cancel, giving **direct force equations**:

$$a_x^{\text{body}} = a - \frac{\sin\delta}{m}(C_f\alpha_f + w_2) - \mu g$$

$$a_y^{\text{body}} = \frac{1}{m}(C_r\alpha_r + w_1) + \frac{\cos\delta}{m}(C_f\alpha_f + w_2)$$

Now we make the **augmented states** (a_x, a_y) track these with a small time constant:

$$\dot{a}_x = \frac{1}{\tau_x}(a_x^{\text{body}} - a_x), \qquad \dot{a}_y = \frac{1}{\tau_y}(a_y^{\text{body}} - a_y)$$

(τ_x, τ_y small, e.g. 0.05–0.2 s)

This is what makes the measurement **constant** (C) and makes w show up strongly (especially in a_y).

2) Continuous-time qLPV form

$$\boxed{\dot{x}_8 = A_8(\rho)x_8 + B_8(\rho)u + b_8 + E_8(\rho)w}$$

2.1 The full $A_8(\rho)$ (8×8)

Let $s = \sin\delta,\ c = \cos\delta$. Then:

$$A_8(\rho) = \begin{bmatrix}
0 & \frac{C_f s}{m v_x} & 0 & \frac{C_f l_f s}{m v_x} + v_y & 0 & 0 & 0 & 0 \\
0 & -\frac{C_f c + C_r}{m v_x} & 0 & -v_x + \frac{l_r C_r - l_f C_f c}{m v_x} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & -\frac{l_f C_f c - l_r C_r}{I_z v_x} & 0 & -\frac{l_f^2 C_f c + l_r^2 C_r}{I_z v_x} & 0 & 0 & 0 & 0 \\
\cos\psi & -\sin\psi & 0 & 0 & 0 & 0 & 0 & 0 \\
\sin\psi & \cos\psi & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{\tau_x}\frac{C_f s}{m v_x} & 0 & \frac{1}{\tau_x}\frac{C_f l_f s}{m v_x} & 0 & 0 & -\frac{1}{\tau_x} & 0 \\
0 & -\frac{1}{\tau_y}\frac{C_f c + C_r}{m v_x} & 0 & \frac{1}{\tau_y}\frac{l_r C_r - l_f C_f c}{m v_x} & 0 & 0 & 0 & -\frac{1}{\tau_y}
\end{bmatrix}$$

Notes:

- The **top 6×6** is your qLPV bicycle (with $-\mu g$ kept in b_8 below).
- The **last two rows** are the IMU acceleration tracking dynamics using the **body-accel** formula (best for w).

2.2 The full $B_8(\rho)$ (8×2)

$$B_8(\rho) = \begin{bmatrix} -\frac{C_f s}{m} & 1 \\ \frac{C_f c}{m} & 0 \\ 0 & 0 \\ \frac{l_f C_f c}{I_z} & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{1}{\tau_x}\frac{C_f s}{m} & \frac{1}{\tau_x} \\ \frac{1}{\tau_y}\frac{C_f c}{m} & 0 \end{bmatrix}$$

This is **exact** given your modeling choice $F_{yf} = C_f \alpha_f + w_2$ (so the δ term creates the $\delta \sin \delta$ and $\delta \cos \delta$ effects via scheduling).

2.3 The full $E_8(\rho)$ (8×2)

Remember $w = [w_1\ w_2] = [f_r(\alpha_r)\ f_f(\alpha_f)]$. Then:

$$E_8(\rho) = \begin{bmatrix} 0 & -\frac{s}{m} \\ \frac{1}{m} & \frac{c}{m} \\ 0 & 0 \\ -\frac{l_r}{I_z} & \frac{l_f c}{I_z} \\ 0 & 0 \\ 0 & 0 \\ 0 & -\frac{1}{\tau_x}\frac{s}{m} \\ \frac{1}{\tau_y}\frac{1}{m} & \frac{1}{\tau_y}\frac{c}{m} \end{bmatrix}$$

Key benefit: w now affects **both the dynamics** and also **directly the acceleration channels**, which helps estimation a lot.

2.4 Constant affine term (b_8)

$$b_8 = \begin{bmatrix} -\mu g & 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_x}\mu g & 0 \end{bmatrix}$$

- That $-\mu g$ is your longitudinal rolling/friction bias term (not "IMU gravity"; it's your model's constant decel).

If you want, you can also add IMU biases as extra states later, but you didn't ask for that here.

3) Constant measurement matrix C (two modes)

Mode A: GPS + IMU + gyro (recommended)

If you measure $y = [v_x,\ \psi,\ r,\ X,\ Y,\ a_x,\ a_y]$:

$$y = C_{\text{GPS}} x_8, \qquad C_{\text{GPS}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Mode B: no GPS (still constant C)

If you measure $y = [v_x,\ r,\ a_x,\ a_y]$:

$$C_{\text{noGPS}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

4) Observer-ready Luenberger form (still constant C)

$$\boxed{\dot{\hat{x}}_8 = A_8(\hat{\rho})\hat{x}_8 + B_8(\hat{\rho})u + b_8 + E_8(\hat{\rho})\hat{w} + L(\hat{\rho})(y - C\hat{x}_8)}$$

Where $\hat{\rho} = \rho(\hat{x}_8, u)$ (or use measured v_x, ψ, δ directly when available).

In particular, we make the standard assumptions:

1. $v_x(t)$ is measured and strictly positive (used as a scheduling signal), and variations of v_x are slow enough that the lateral–yaw dynamics can be written as an LPV system in v_x.
2. Small-angle approximation for the steering angle: $\cos(\delta) \approx 1$, $\sin(\delta) \approx \delta$.
3. The longitudinal force coupling term $-\frac{F_{yf}\sin(\delta)}{m}$ in $\dot{v}_x$ is neglected in the nominal model (it is second-order under small angles) and any remaining longitudinal effects are treated as modeling error; the estimator focuses on lateral tire-force uncertainties.

$$\begin{cases} \dot{x} = A(v_x)x + Bu + E\,\mu_x(x,u) \\ y = Cx \end{cases} \tag{3.55}$$

where $u := [a,\ \delta]^\top$ and the unknown input collects the nonlinear tire-force components

$$\mu_x(x,u) := \begin{bmatrix} f_{yf}(\alpha_f) \\ f_{yr}(\alpha_r) \end{bmatrix}. \tag{3.56}$$

The nominal matrices are: model not correct yet, to be fixed

$$A(v_x) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{C_f+C_r}{mv_x} & 0 & \frac{l_rC_r-l_fC_f}{mv_x} - v_x \\ 0 & 0 & 0 & 1 \\ 0 & \frac{l_rC_r-l_fC_f}{I_zv_x} & 0 & -\frac{l_f^2C_f+l_r^2C_r}{I_zv_x} \end{bmatrix},$$

$$B = \begin{bmatrix} 1 & 0 \\ 0 & \frac{C_f}{m} \\ 0 & 0 \\ 0 & \frac{l_fC_f}{I_z} \end{bmatrix}, \quad E = \begin{bmatrix} 0 & 0 \\ \frac{1}{m} & \frac{1}{m} \\ 0 & 0 \\ \frac{l_f}{I_z} & -\frac{l_r}{I_z} \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The term $\mu_x(x,u)$ lumps the unmodeled nonlinear tire dynamics (and may also capture residual parameter mismatches through their effect on the lateral tire forces). For the simulation, we introduce a significant parametric uncertainty: the "true" mass and cornering stiffnesses are set to be 20% higher than the nominal values used in the observer design ($m_{true} = 1.2m_{nom}$, etc.). This creates a challenging scenario for the estimator.

7.1.1 Rank Condition Check (Why a Generalized/Regularized UIO Is Needed)

In the vehicle model (3.55), the unknown input is $\mu_x = [f_{yf}(\alpha_f),\ f_{yr}(\alpha_r)]^\top \in \mathbb{R}^2$ and enters the state dynamics through $E \in \mathbb{R}^{4\times 2}$, while the measured output is $y = Cx \in \mathbb{R}^3$. In classical UIO theory, a necessary rank condition for unknown-input reconstruction/decoupling is

$$\operatorname{rank}(CE) = \operatorname{rank}(E). \tag{3.57}$$

With the matrices defined above, we have

$$CE = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{l_f}{I_z} & -\frac{l_r}{I_z} \end{bmatrix},$$

so that, under the standard assumptions $l_f > 0$, $l_r > 0$, $I_z > 0$, one obtains $\operatorname{rank}(E) = 2$ whereas $\operatorname{rank}(CE) = 1$. Therefore, the classical rank condition (3.57) is *not satisfied* for the considered sensor set $y = [v_x,\ \psi,\ r]^\top$. This means that a standard UIO cannot reconstruct the two components of μ_x from the available measurements. This motivates the use of the generalized/regularized UIO developed in

Section 3, which restores the structural rank condition for the augmented descriptor system (Assumption 5 and (3.4)) at the expense of introducing additional bounded disturbance terms.

7.2 Observer Design and Implementation

7.2.1 LPV Scheduling and Polytopic Embedding

The matrix $A(v_x)$ in (3.55) depends on both v_x and $1/v_x$. In the simulations, v_x is directly measured (first component of y), hence the scheduling variables can be taken as

$$\rho_1(t) := v_x(t), \qquad \rho_2(t) := \frac{1}{v_x(t)}.$$

Assuming $v_x(t) \in [\underline{v}_x, \overline{v}_x]$, we have $\rho_1 \in [\underline{v}_x, \overline{v}_x]$ and $\rho_2 \in [1/\overline{v}_x, 1/\underline{v}_x]$. Since $A(v_x)$ is affine in ρ_1 and ρ_2, one can embed $A(v_x)$ into a (conservative) polytopic set

$$A(v_x(t)) \in \mathrm{Co}\{A_1, \dots, A_{\bar{n}}\},$$

where the vertices A_j are obtained by evaluating $A(v_x)$ at the corners of (ρ_1, ρ_2) (typically $\bar{n} = 4$). This polytopic description is used to synthesize constant observer gains via LMIs.

Remark (Number of gains vs number of vertices). The presence of $\bar{n} = 4$ vertices does not necessarily imply that four different observer gains must be implemented. In this chapter, we follow a *common-gain* LPV design: the LMIs are enforced at all vertices A_j, but the decision variables are shared, yielding a *single constant* gain (e.g., $K = \mathcal{P}^{-1}\mathcal{Q}^\top$) that is valid for all $A(v_x)$ in the convex hull. An alternative (not pursued here) is a gain-scheduled design with vertex gains $\{K_j\}_{j=1}^{\bar{n}}$ interpolated online as $K(\rho) = \sum_{j=1}^{\bar{n}} \alpha_j(\rho) K_j$.

7.2.2 UIO for the LPV Vehicle Model (Gain K)

We adopt the LMI-based UIO framework developed in Section 3 to estimate simultaneously the state x and the unknown input μ_x. To satisfy the structural rank condition (3.4) for the augmented variable $\zeta := [x^\top, \mu_x^\top]^\top$, we use the regularized output

$$y = Cx + D_{\delta_2}\mu_x + D\,\omega_{\delta_2}, \tag{3.58}$$

with $D_{\delta_2} = \delta_2 D$, $\delta_2 > 0$ small, and a binary matrix $D \in \{0,1\}^{n_y \times n_\mu}$ chosen such that $\mathrm{rank}(D) = n_\mu$ (hence $\mathrm{rank}([\mathbb{E}^\top, \mathcal{H}^\top]^\top) = n_x + n_\mu$). A convenient choice for $n_y = 3$ and $n_\mu = 2$ is, for example,

$$D = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.59}$$

With this choice, the UIO can be written in the standard form (3.10):

$$\begin{cases} \dot{\eta} &= P_\zeta \varphi_\zeta(\hat{\zeta}, u) + K\left(y - \mathcal{H}\hat{\zeta}\right) \\ \hat{\zeta} &= \eta + Q_\zeta y \end{cases} \tag{3.60}$$

where P_ζ and Q_ζ satisfy (3.9). For the LPV vehicle model, the Jacobian of φ_ζ is polytopic; its vertices can be constructed as

$$\mathcal{A}_{\zeta,j} := \begin{bmatrix} A_j & E \\ 0 & 0 \end{bmatrix}, \qquad j = 1, \ldots, \bar{n}. \tag{3.61}$$

Then, the UIO gain is obtained from the LMI synthesis in Theorem 2: solve the LMIs (3.26) for all vertices j, recover the decision variables $(\mathcal{P}, \mathcal{Q})$, and compute

$$K = \mathcal{P}^{-1}\mathcal{Q}^\top. \tag{3.62}$$

7.2.3 Neural Adaptive Observer (NAO) (Gain L_{NN})

The neural observer uses the measured scheduling variable $v_x(t)$ to implement the LPV dynamics online:

$$\dot{\hat{x}}_{NN} = A(v_x)\hat{x}_{NN} + Bu + E\,\mu_\theta(\hat{x}_{NN}) + L_{NN}(y - C\hat{x}_{NN}). \tag{3.63}$$

The gain L_{NN} is designed by applying Theorem 3 to the polytopic LPV set $\{A_j\}_{j=1}^{\bar{n}}$ (i.e., take $\mathcal{A}_j^x := A_j$ in (3.41)), and then compute $L_{NN} = \mathcal{P}^{-1}\mathcal{Q}^\top$.

7.2.4 Gradient Computation Using the Vehicle LPV Model

Let $\hat{x}_k$ denote the discrete-time NAO state (Euler discretization with step Δt) and $\rho_k := v_{x,k} = y_{1,k}$ be the measured scheduling signal. The observer update can be written as

$$\hat{x}_{k+1} = \hat{x}_k + \Delta t\, f_{\text{obs}}(\hat{x}_k, \theta, \rho_k).$$

The sensitivity recursion used to compute $\nabla_\theta \mathcal{L}$ is

$$S_{k+1} = S_k + \Delta t\,(A_k S_k + B_k), \qquad S_k := \frac{\partial \hat{x}_k}{\partial \theta}, \tag{3.64}$$

with the vehicle-dependent Jacobians

$$A_k = \frac{\partial f_{\text{obs}}}{\partial \hat{x}_k} = A(\rho_k) + E\frac{\partial \mu_\theta}{\partial \hat{x}_k} - L_{NN}C, \tag{3.65}$$

$$B_k = \frac{\partial f_{\text{obs}}}{\partial \theta} = E\frac{\partial \mu_\theta}{\partial \theta}. \tag{3.66}$$

This shows explicitly that the gradient computation uses the physical LPV model through $A(\rho_k)$ at each time step.

7.3 Results and Discussion

Not finshed yet, to be completed The performance of the proposed Neural Adaptive Observer (NAO) is compared against a standard Proportional-Integral (PI) observer and a baseline UIO.

7.3.1 State Estimation

Figure 3.3 and Figure 3.4 illustrate the state estimation results. While the longitudinal velocity v_x and yaw angle ψ are directly measured and easily tracked by all observers, the lateral velocity v_y and yaw rate r are more challenging due to the unmodeled dynamics. The proposed NAO demonstrates superior performance in estimating these states compared to the PI observer. Initially, some oscillations are observed (Fig. 3.3) as the neural network weights are randomly initialized. However, by applying the continuous learning strategy with a loss accumulation batch, these spikes are significantly mitigated (Fig. 3.4), resulting in smooth and accurate tracking.

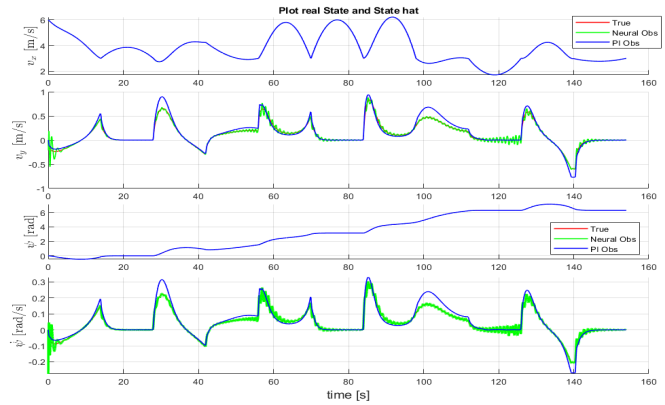


Figure 3.3: State Estimation: Comparison between True State, Neural Observer, and PI Observer (Without Loss Accumulation).

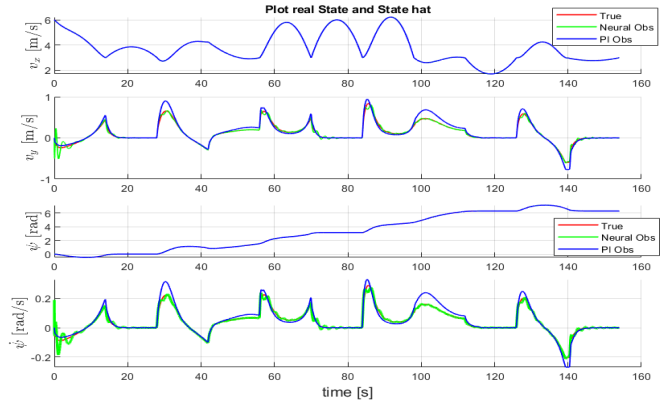


Figure 3.4: State Estimation: Comparison with Loss Accumulation Strategy. Note the reduction in estimation noise and spikes.

7.3.2 Uncertainty Estimation

Figure 3.5 shows the estimation of the unknown dynamics μ_x. The proposed method (labeled "Neural Observer") is compared with the approach from [**GHANI20235685**]. Our method leverages the UIO estimate as a "proxy ground truth" in the loss function, which guides the neural network to quickly converge to the true uncertainty profile. The results show that our observer captures the nonlinear variations of the tire forces more accurately and with less delay than the baseline.

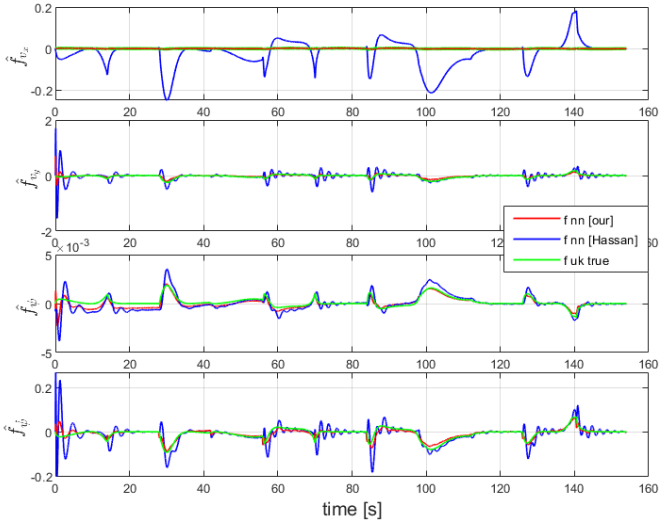


Figure 3.5: Estimation of Unmodeled Dynamics: Comparison of the proposed method against baseline approaches.

7.3.3 Convergence Analysis

The evolution of the loss function during the simulation is depicted in Figure 3.6. The average loss value stabilizes around 0.0297, indicating successful convergence. The transient spikes correspond to aggressive maneuvering phases (e.g., sharp turns) where the unmodeled dynamics change rapidly. The rapid decay of these spikes confirms the ability of the online learning algorithm to adapt quickly to new operating conditions.

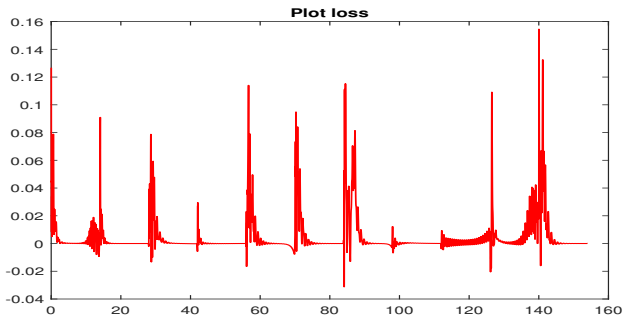


Figure 3.6: Evolution of the Loss Function during the simulation.

8 Open Problems and Future Directions

Despite the promising theoretical guarantees and simulation results, several research questions remain open. The following points outline practical and theoretical directions that can improve the applicability of the proposed Teacher–Student estimation architecture.

- **Output-derivative availability and noise amplification.** The generalized/regularized construction is motivated by the need to enrich the output information (cf. rank recovery). In practice, output derivatives may come from IMU signals or filtered differentiators. A key open problem is to quantify the effect of measurement noise and filtering dynamics on the $\mathcal{H}^1$ performance bounds, and to design derivative estimators that preserve the observer guarantees.

- **Guidelines for choosing E, D and tuning (δ_1, δ_2).** While sparse/binary choices of E and D are effective to recover the rank condition, systematic selection procedures that optimize the trade-off between feasibility and disturbance injection remain to be developed. This includes principled tuning of (δ_1, δ_2) as a function of sensor noise, bandwidth limitations, and expected disturbance energy.

- **Beyond $\mathcal{L}_2$-bounded uncertainties.** Assumption 4 (bounded-energy uncertainty) is convenient for analysis, but some automotive disturbances are better modeled as bounded-amplitude or piecewise-smooth signals. Extending the framework to mixed norms (e.g., $\mathcal{L}_\infty$-to-$\mathcal{L}_2$) or incremental input-to-state stability (ISS) type bounds is an interesting direction.

- **Learning conditions and excitation.** The neural adaptive observer uses the UIO estimate as a supervisory signal; however, learning accuracy still depends on the richness of the visited trajectories. Characterizing persistent excitation conditions (or alternative data coverage metrics) that guarantee parameter convergence and good generalization to unseen maneuvers remains an important open issue.

- **Closed-loop interaction and safety guarantees during learning.** In practical deployments, the estimator is embedded in a feedback loop. A challenging direction is to establish end-to-end stability/performance guarantees for the controller–observer–learner interconnection, especially during transient learning phases and under actuator saturation.

- **Discrete-time and embedded implementation.** Real systems operate with sampled measurements and limited compute. Deriving discrete-time counterparts of the $\mathcal{H}^1$ UIO synthesis and the coupled neural adaptation laws—including rigorous treatment of discretization errors and real-time constraints—is required for embedded deployment.

- **Experimental validation and domain shift.** Simulation studies can underestimate issues such as unmodeled sensor biases, delays, and varying road conditions. A natural future direction is validation on real driving datasets and test vehicles, including robustness to domain shift (e.g., different tires, mass loading, and road friction regimes) and the integration of uncertainty quantification for the learned component.

9 Conclusion

This paper presented a robust hybrid estimation framework designed to address the challenges of vehicle motion tracking in the presence of unmodeled nonlinearities. We proposed a layered architecture that integrates a physics-based Generalized Unknown Input Observer (UIO) with a data-driven Neural Adaptive Observer. To overcome the structural limitations of standard observers, specifically when the rank condition $rank(CB) = rank(B)$ is not met, we introduced a regularization technique combined with an output-derivative based generalized model. This approach ensures the existence of the observer and provides a reliable initial estimate of the unknown dynamics. This estimate subsequently serves as a supervisor for a neural network, which refines the approximation of complex nonlinearities, such as variable tire-road friction, through online learning. Theoretical analysis demonstrated that the proposed method guarantees $\mathcal{H}^1$ and $\mathcal{L}^2$ stability for the estimation errors, provided that the derived Linear Matrix Inequality (LMI) conditions are satisfied. Open problems and future directions are discussed in Section 8, including experimental validation using real driving data and discrete-time formulations for embedded implementation.

Chapter 4

Resilient Trust–Aware Distributed Observer Design for Connected Vehicle Platoons

Objectives

This chapter proposes a trust-aware distributed observer for vehicle platoons that maintains resilient state estimation under cyberattacks. A behavioral divergence metric evaluates the reliability of shared data, forming a dynamic neighbor set used to adapt observer's weighting gains. Stability conditions are derived via Lyapunov analysis. Simulations under bogus, replay, and DoS attacks demonstrate robust performance and stable platoon behavior.

Contents

1 Introduction

1.1 General introduction

The rapid development of autonomous and connected vehicles has introduced new opportunities to improve road safety, traffic flow, and fuel efficiency. In vehicle platoons-where multiple vehicles coordinate through vehicle-to-vehicle (V2V) communication-accurate distributed state estimation is essential for maintaining stability and efficiency. Each vehicle must reconstruct both its own and others' states using locally sensed data and information exchanged with neighbors. However, this distributed structure also exposes the system to cyber and communication attacks that can disrupt coordination or compromise state estimation [**Ali_book**].

Traditional estimation and control methods generally assume all vehicles are cooperative and trustworthy, which is unrealistic in adversarial environments [**Shengya_HG**]. Malicious or faulty agents can transmit falsified information, causing estimation errors that propagate through the network. Consequently, resilient estimation strategies capable of identifying and isolating untrustworthy data have become a key research direction in cyber-physical systems (CPS) and distributed multi-agent systems (DMAS).

Existing work on secure estimation and fault detection can be categorized into three main families:

- *Observer-based detection:* compares model-predicted states with sensor measurements to identify anomalies [**Huy_LCSS**; **2_obs_Angelo**]. These methods are effective for detecting deviations but may propagate errors if compromised neighbors provide false data.
- *Consensus-based detection:* relies on cross-validation among agents to identify inconsistent information [**Guitao_multi_mesure**]. While robust under the assumption that most agents are trustworthy, their performance degrades under large-scale coordinated attacks.
- *Trust-based detection:* assigns reliability scores to neighboring agents based on behavioral or statistical consistency. Physical-signal approaches use device fingerprints for authentication [**trust_physic_connecte**], while statistical approaches compare communicated data against locally observed behaviors [**Wang_2021**]. These methods improve resilience by filtering unreliable inputs before consensus or estimation steps, but they typically require significant computational resources or long-term historical data-making real-time adaptation challenging for dynamic platoons.

Mitigation strategies in the literature often depend on attack detection results. Some switch to fallback modes (e.g., CACC to ACC), while others incorporate trust scores directly into control to down-weight suspicious data [**self_belive**; **Enhancing_control_trust**]. The latter provides finer control adaptation but relies heavily on accurate and timely trust evaluation. To overcome these limitations, we propose a trust-aware distributed observer framework for vehicle platoons. The framework continuously evaluates the trustworthiness of communicated data by comparing reported and predicted behaviors, dynamically adjusting observer weights according to each neighbor's trust score. This mechanism ensures reliable estimation even under cyberattacks or communication faults.

The main contributions of the paper can be summarized in the following items:

- A formal definition of the resilient distributed state estimation problem for connected vehicles under adversarial conditions.

- A behavioral trust model that quantifies each agent's reliability and constructs a dynamic trusted neighbor set.
- A trust-informed distributed observer with provable stability guarantees, validated through simulations under multiple attack scenarios.

By integrating detection, mitigation, and estimation within a unified trust-based layer, the proposed method improves the resilience and stability of cooperative vehicle systems.

1.2 Graph theory

An undirected graph with a nonempty finite set of N nodes can be described by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ where:

- $\mathcal{V} = \{1, 2, \ldots, N\}$ is the set of nodes.
- $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges.
- $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{N \times N}$ is the weighted adjacency matrix of $\mathcal{G}$ defined by

$$[a_{ij}] = \begin{cases} a_{ij}, & \text{if } (i,j) \in \mathcal{E}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

- The *Laplacian* matrix of $\mathcal{G}$ is denoted as $\mathcal{L} = [l_{ij}] \in \mathbb{R}^{N \times N}$ with

$$l_{ij} = \begin{cases} \sum_{k=1}^{N} a_{ik}, & \text{if } i = j, \\ -a_{ij}, & \text{otherwise.} \end{cases} \tag{4.2}$$

- The set of the neighbors of the node i is defined by $\mathcal{N}_i = \{j \in \mathcal{V} \mid (j,i) \in \mathcal{E}\}$.

1.3 Vehicle mathematical model

The platoon of N vehicles considered in this paper can be modeled as a network of N agents, represented by an undirected communication graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$. The discrete-time longitudinal dynamics of each vehicle $i \in \mathcal{V}$ are

$$\begin{aligned} x_i(t+1) &= A_{\mathrm{b}} x_i(t) + B_{\mathrm{b}} u_i(t) + \Delta_i(t), \\ y_i(t) &= C_i x_i(t), \end{aligned} \tag{4.3}$$

where $x = \begin{bmatrix} s_i & v_i & a_i \end{bmatrix}^\top$ and u_i denote the state and control input of vehicle i, respectively. s_i, v_i, and a_i denote position, velocity, and acceleration. The system matrices are given by

$$A_{\mathrm{b}} = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 - \frac{T_s}{\tau_b} \end{bmatrix}, \; B_b = \begin{bmatrix} 0 \\ 0 \\ \frac{T_s}{\tau_b} \end{bmatrix}, C_{\mathrm{b}} = I_3,$$

where T_s is the sampling time and the constant $\tau_b > 0$ is nominal engine time lag. $\Delta_i(t)$ captures the modeling differences and uncertainties caused by difference between nominal τ_b and the real value.

Define the collective state as $x = \text{col}\ (x_1, \cdots, x_N)$, the state-space equation of the platoon is

$$\begin{cases} x(t+1) = (I_N \otimes A_\text{b})x(t) + (I_N \otimes B_\text{b})u(t) + \Delta(t) \\ y_i(t+1) = C_i x(t), \end{cases} \tag{4.4}$$

where $u = \text{col}\ (u_1, \cdots, u_N)$ is the collective control input, $\Delta = \text{col}\ (\Delta_1, \cdots, \Delta_N)$ group unmodeled dynamics and external disturbances and $C_i = \begin{bmatrix} 0 \cdots \underbrace{C_b}_{i\text{th}} \cdots 0 \end{bmatrix}$ is the global output matrix.

Since each vehicle measures its own full states through onboard sensors, ensuring that the pair $(A_\text{b},\ C_\text{b})$ is observable. However, in the context of the platoon dynamic (4.4), the pair $((I_N \otimes A_\text{b}), C_i)$ remain unobservable, motivating the need for distributed estimation through inter-vehicle communication.

2 Distributed State Observer Architecture

The distributed observer provides an effective approach to estimate the full platoon state x. In this section, we introduce a new distributed observer architecture that incorporates trust-function mechanisms, which will be detailed later in Section 3. To achieve full-state estimation of all vehicles, two complementary layers are employed: a Local Observer ($\mathcal{LO}$) that uses onboard sensors, and a Distributed Observer ($\mathcal{DO}$) that fuses information exchanged through communication links. Fig. 4.1 illustrates the architecture of the proposed distributed observer for vehicle platoons.

2.1 General structure of the observer

The distributed state observer architecture considered in this study is characterized by the following set of equations (4.5):

$$(\mathcal{LO}_i) : \hat{x}_0^{(i)}(t+1) = A_\text{b}\hat{x}_0^{(i)}(t) + B_\text{b}u_i(t) + F_i(y_i(t) - C_\text{b}\hat{x}_0^{(i)}(t)) \tag{4.5a}$$

$$(\mathcal{DO}_i^{(j)}) : \hat{x}_i^{(j)}(t+1) = A_\text{b}\left(\hat{x}_i^{(j)}(t) + \sum_{l\in\mathcal{N}_i} w_{il}^{(j)}(t)(\hat{x}_l^{(j)}(t) - \hat{x}_i^{(j)}(t)) + w_{i0}^{(j)}(t)\left(\hat{x}_0^{(j)}(t) - \hat{x}_i^{(j)}(t)\right)\right) + B_\text{b}\hat{u}_j(t) \tag{4.5b}$$

where

- $\hat{x}_0^{(i)}(t)$ is the state of the local observer $\mathcal{LO}_i$, which will try to track the states of the ith vehicle. It means that
$$\lim_{t\to\infty}\left(\hat{x}_0^{(i)}(t) - x^{(i)}(t)\right) = 0. \tag{4.6}$$
- F_i is the gain matrix of the local observer.
- $\mathcal{N}_i$ is the neighbors of the ith vehicle according to the communication graph $\mathcal{G}$.
- $w_{il}^{(j)}(t)$, $l \in \mathcal{N}_i^{(j)}$ is the gain scale to be designed based on the communication topology.
- $\hat{x}_i^{(j)}(t)$ is the state of the distributed observer $\mathcal{DO}_i^{(j)}$, which will try to track the states of the platoon. It means that
$$\lim_{t\to\infty}(\hat{x}_i(t) - x(t)) = 0, \tag{4.7}$$

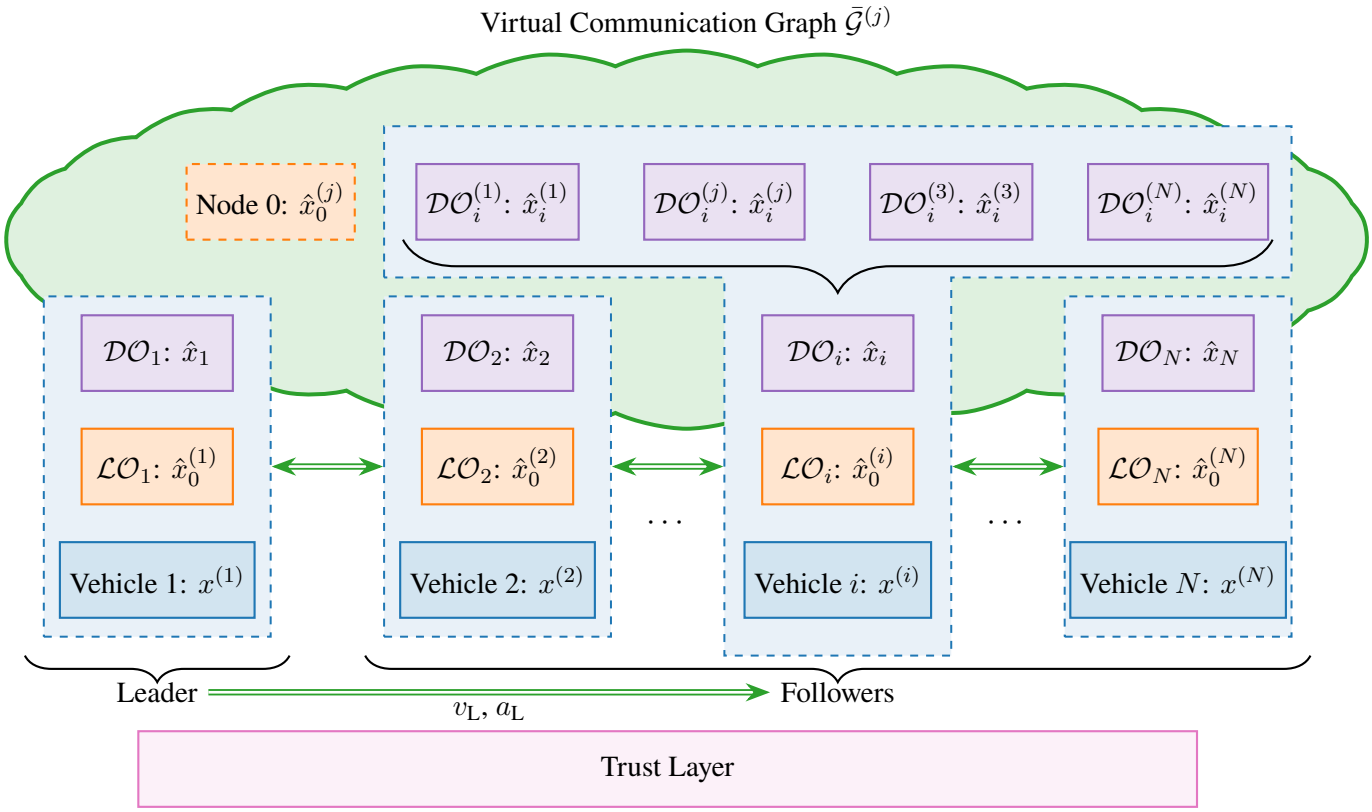


Figure 4.1: Schematic diagram of the estimation method.

where $\hat{x}_i(t) = \text{col}\left(\hat{x}_i^{(1)}(t), \cdots, \hat{x}_i^{(N)}(t)\right)$ is the collective states of all distributed observer.

- $w_{i0}^{(j)}$ is the gain scale to be designed, which is such that

Remark 7. *If a host forms $\hat{u}_j(t)$ (e.g., from a known controller law or an input estimator), the mismatch*

$$\Delta u_j(t) = u_j(t) - \hat{u}_j(t) \tag{4.8}$$

will be treated as part of the disturbance vector. We do not require $\hat{u}_j(t)$ for the $(\mathcal{DO})$ to run; using it only tightens bounds.

2.2 On the structure of the observer (4.5)

The distributed observer $\mathcal{DO}_i^{(j)}$ is designed to estimate the state of each vehicle $j \in \mathcal{V}$ in the platoon from the perspective of vehicle i. The term $\sum_{l\in\mathcal{N}_i} w_{il}^{(j)}\left(\hat{x}_l^{(j)}(t) - \hat{x}_i^{(j)}(t)\right)$ takes into account communication with neighbors $l \in \mathcal{N}_i$ to adjust $\hat{x}_i^{(j)}(t)$ towards a consensus among neighboring estimates of vehicle j's state. The weights $w_{il}^{(j)}$ determine the influence of each neighbor's estimate, promoting consistency across the platoon, which will be designed later. The consensus term alone has a key limitation , it cause all vehicles estimates of a given state $\hat{x}_i^{(j)}(t)$ to converge to a common value. This is theoretically sound only for an ideal platoon with perfectly aligned states. However, practical disturbances, initial condition errors, and imperfect models break this uniformity, rendering a pure consensus estimate inaccurate for any individual vehicle's true state. To mitigate this, the local observer $\mathcal{LO}_i$ provides an accurate estimate of

vehicle i's own state, $\hat{x}_0^{(i)}(t)$. The additional anchoring term $w_{i0}^{(j)}\left(\hat{x}_0^{(j)}(t) - \hat{x}_i^{(j)}(t)\right)$ injects this trusted reference, allowing each vehicle to correct model errors and maintain accurate peer estimates.

2.3 Introduction of a virtual communication graph

Each distributed observer $\mathcal{DO}_i^{(j)}$ requires reference information from the local observer $\mathcal{LO}_j$. To formally include this reference, we define a virtual node labeled as node "0" that connects to the vehicle j, where $j \in \mathcal{V}$ and its neighbors $\mathcal{N}_j$ according to the original graph $\mathcal{G}$. The resulting virtual communication graph $\bar{\mathcal{G}}^{(j)} = \{\bar{\mathcal{V}}, \bar{\mathcal{E}}^{(j)}\}$ is constructed as:

$$\begin{aligned} \bar{\mathcal{V}} &= \mathcal{V} \cup \{0\}, \\ \bar{\mathcal{E}}^{(j)} &= \mathcal{E} \cup \{(0,j)|j \in \mathcal{V}\} \cup \{(0,k)|k \in \mathcal{N}_j\}. \end{aligned} \tag{4.9}$$

The adjacency matrix $\bar{\mathcal{A}}^{(j)} \in \mathbb{R}^{(N+1)\times(N+1)}$ corresponding to $\bar{\mathcal{G}}^{(j)}$ is defined as:

$$\bar{\mathcal{A}}^{(j)} = \begin{bmatrix} 0 & 0 \\ w_0^{(j)} & \mathcal{A} \end{bmatrix}, \tag{4.10}$$

where the vector $w_0^{(j)} = \text{col}\left(w_{10}^{(j)}, \cdots, w_{N0}^{(j)}\right) \in \mathbb{R}^N$ defines the connection strengths from node 0 to the distributed observers.

This virtual communication graph provides a unified representation for integrating local estimation (through node 0) and neighbor-based consensus (through graph G), ensuring that every vehicle's local observer acts as a trusted anchor for distributed state estimation.

2.4 Main objectives

Traditional distributed observers use fixed weights w_{il} that assume all data sources are trustworthy. However, in a connected vehicle platoon, compromised nodes may send falsified information. To address this issue, the proposed design incorporates trust-based and time-varying weighting $w_{il}(t)$, allowing each vehicle to dynamically adjust its observer weights according to the reliability of received information. To address this issue, one of the key challenges is to define a quantitative measure of *trust* that captures how reliably a host vehicle can utilize information from its neighbors, despite potential uncertainties or malicious data. Based on this trust, an *adaptive update law* for the gain scaling factor $w_{il}(t)$ is designed, depending on both the computed trust value and the gain matrix F_i of the local observer $(\mathcal{LO}_i)$. This trust mechanism is incorporated into the observer synthesis conditions, presented later in Section 4, ensuring an appropriate ISS convergence bound.

3 Trust Score Framework

To ensure resilient estimation, each vehicle continuously evaluates the reliability of the data it receives. This is achieved through a multi-layer trust framework that quantifies how much a host vehicle i trusts another vehicle l. The overall structure of this trust framework is illustrated in Fig. 4.2.

In this context, we define the following types of trust:

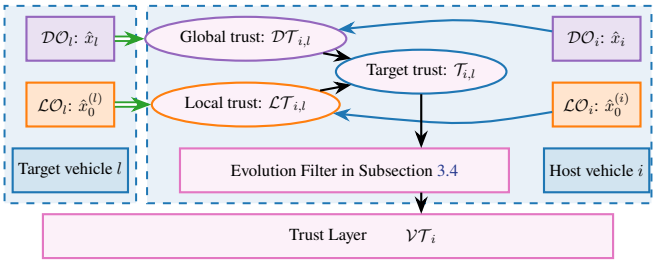


Figure 4.2: Trust framework.

Table 4.1: Summary of observer notation, trust variables, and index meaning.

Category	Notation	Meaning
Observers	$\mathcal{LO}_i$, $\mathcal{DO}_i$	Local observer at vehicle i (sensor-based), and distributed observer at vehicle i (V2V fusion).
	$\hat{x}_0^{(i)}(t)$	Local state estimate of vehicle i produced by $\mathcal{LO}_i$.
	$\mathcal{DO}_i^{(j)}$, $\hat{x}_i^{(j)}(t)$	Vehicle i's estimate of the target vehicle j's state (distributed layer).
Trust variables	$\delta_{i,l}(t)$	Packet validity indicator (authenticated and fresh).
	$\mathcal{LT}_{i,l}$, $\mathcal{DT}_{i,l}$	Local trust (local-observer consistency) and distributed/global trust (agreement of distributed estimates).
	$\gamma_{i,l}^{\text{self}}$	Self-consistency factor: checks whether the host's own distributed observer global estimate is consistent with local relative measurements.
	$\mathcal{T}_{i,l}$, $\mathcal{VT}_i$	Target trust for vehicle l and its smoothed evolution (vehicle trust).
	O_i, $O_i(j)$	Generalized trust/opinion vector of vehicle i over all platoon members, with entry $O_i(j) \in [0,1]$ the opinion toward vehicle j.
	$w_{il}^{(j)}(t)$, $w_{i0}^{(j)}(t)$	Trust-adaptive gain scaling factors (neighbor l and anchor node 0) used in the distributed observer.
Indices / sets	i	Host vehicle (computes trust and runs $\mathcal{DO}_i$).
	l	Target neighbor whose data is assessed (typically $l \in \mathcal{N}_i$).
	j	Target vehicle whose state is being estimated by $\mathcal{DO}_i^{(j)}$.
	t	Discrete-time index.
	0, N, $\mathcal{N}_i$	Anchor node (local-observer reference), number of vehicles, and neighbor set of vehicle i in the communication graph.

- **Local Trust** $\mathcal{LT}_{i,l}$: evaluates the reliability of the local observer $\mathcal{LO}_l$ data from vehicle l based on consistency and timeliness.

- **Distributed Trust** $\mathcal{DT}_{i,l}$: measures how well the global estimates of l agree with those of i.
- **Target Trust** $\mathcal{T}_{i,l}$: Combining the Local Trust and Global Trust, the host vehicle i can evaluate the trust score for each target vehicle at time t.
- **Vehicle Trust** $\mathcal{VT}_i$: Smooth evolution of Target Trust scores over time.

In the following subsections, we will explain how to calculate the trust. After that, we will provide more details on how to determine vehicle trust ($\mathcal{VT}_i$) and how to utilize it effectively.

3.1 Data validity indicator

Before trust computation, every received V2V packet data is validated for authenticity and freshness:

$$\delta_{i,l}(t) = \begin{cases} 1, & \text{if message is authenticated and fresh,} \\ 0, & \text{otherwise.} \end{cases} \tag{4.11}$$

Only packets satisfying $\delta_{i,l}(t) = 1$ use in the following trust evaluation process.

3.2 Local trust

The Local Trust $\mathcal{LT}_{i,l}$ quantifies the confidence that the host vehicle i places in the local observer output of a neighboring vehicle $l \in \mathcal{N}_i$. It is based on three physical-consistency indicators computed from measurable states: velocity, distance, and acceleration. When data are missing, a hold-and-decay rule maintains continuity.

3.2.1 Velocity consistency:

The host estimates the expected velocity of the platoon leader using its last valid packet as a reference value:

$$v_{\text{ref}}(t) = v_{\text{L}}(t - t_{\text{elap}}) + t_{\text{elap}} \cdot a_{\text{L}}(t - t_{\text{elap}}), \tag{4.12}$$

Then, the mismatch score of the target vehicle i with the reference leader can be calculated as

$$v_{l,\text{L}} = \begin{cases} \max\left(1 - \left|\dfrac{\hat{v}_0^{(l)} - v_{\text{ref}}}{v_{\text{ref}}}\right|,\ 0\right), & \text{if } v_{\text{ref}} > 0, \\ \max\left(1 - |\hat{v}_0^{(l)}|,\ 0\right), & \text{otherwise,} \end{cases} \tag{4.13}$$

Similarly, the host vehicle $\hat{v}_0^{(i)}$ compares with its neighbors $\hat{v}_0^{(l)}$, the following mismatch score is:

$$v_{l,\text{H}} = \begin{cases} \max\left(1 - \left|\dfrac{\hat{v}_0^{(l)} - \hat{v}_0^{(i)}}{\hat{v}_0^{(i)}}\right|, 0\right), & \text{if } \hat{v}_0^{(i)} > 0, \\ \max\left(1 - |\hat{v}_0^{(l)}|, 0\right), & \text{otherwise,} \end{cases} \tag{4.14}$$

The velocity mismatch is computed as a weighted average of two estimates:

$$\iota_{i,l}^{\text{velocity}} = ((1 - \sigma)v_{l,\text{L}} + \sigma v_{l,\text{H}}) \tag{4.15}$$

where the weighted σ is selected based on the index position of the target vehicle l. If the target is following the host vehicle i, then $\sigma \in (0, 0.3)$. If the target is leading, $\sigma \in (0.7, 1)$.

3.2.2 Distance consistency:

The host vehicle i calculates the distance with respect to the target l as :

$$d_{i,l} = \hat{s}_{\text{x},0}^{(l)} - \hat{s}_{\text{x},0}^{(i)} - L,$$

where L is the length of the vehicle. So the mismatch between the measured distance d from its sensors and the calculated data $d_{k,i}$ is

$$\iota_{i,l}^{\text{distance}} = \max\left(1 - \left|\frac{d_{i,l} - d}{d}\right|, 0\right). \tag{4.16}$$

Assumption 9. *When a vehicle is connected, it has access to basic information about its neighbors, including their type and length.*

3.2.3 Acceleration consistency:

The host vehicle can calculate the expected relative acceleration as

$$a_{\text{recv}}^{\text{rel}} = \hat{a}_0^{(l)} - \hat{a}_0^{(i)}, \qquad a_{\text{expect}}^{\text{rel}} = \frac{v_{\text{rel}}(t) - v_{\text{rel}}(t-n)}{n},$$

where $v_{\text{rel}} = \hat{v}_0^{(l)} - \hat{v}_0^{(i)}$ and n defines the averaging window used to smooth noise. To account for physical context, the mismatch is normalized by both distance and relative velocity:

$$\iota_{i,l}^{\text{accel}} = \max\left(\left|\frac{v_{\text{rel}}}{d_{i,l}} \cdot a_{\text{recv}}^{\text{rel}} - a_{\text{expect}}^{\text{rel}}\right|, 0\right) \tag{4.17}$$

where d_{norm} are normalization constants reflecting typical spacing in the platoon.

This formulation tightens the trust threshold when vehicles are close or moving fast, where inconsistencies are more critical, and relaxes it when they are farther apart or nearly stationary. It provides smoother and dimensionally consistent trust updates while reducing false trust drops caused by minor sensor fluctuations.

3.2.4 Fusion and update rule:

We combine all three indicators into an integrity measure $\mathcal{LT}_{i,l}$ indicates the trust of local data of the host vehicle i in the target vehicle l at time t:

$$\mathcal{LT}_{i,l}(t) = \begin{cases} f(\iota_{i,l}^{\text{velocity}}, \iota_{i,l}^{\text{distance}}, \iota_{i,l}^{\text{accel}}) & \text{if } \delta_{i,l}(t) = 1 \\ (1 - \lambda_h)\mathcal{LT}_{i,l}(t-1), & \text{otherwise} \end{cases} \tag{4.18}$$

where f is a fusion function that combines the three scores. The parameter λ_h defines the slow decay during temporary losses.

Remark 8. *Function f can be a weighted average or a minimum function, depending on the desired sensitivity to individual indicators. The decay λ_h can be varied based on how many consecutive packets are missed. This prevents a single missed packet from instantly reducing trust to zero, while still penalizing prolonged disconnection.*

3.3 Distributed trust

Same concept of Local Trust but in the context of the distributed data. We denote $\mathcal{DT}_{i,l}$ represents how the host vehicle i trusts the distributed estimation received from target vehicle l, which is defined as

$$\mathcal{DT}_{i,l} = \begin{cases} \gamma_{i,l}^{\text{self}} \cdot \gamma_{i,l}^{\text{local}} \cdot \gamma_{i,l}^{\text{host}} & \text{if } \delta_{i,l}(t) = 1 \\ (1-\lambda_h)\mathcal{DT}_{i,l}(t-1), & \text{otherwise} \end{cases} \tag{4.19}$$

where $\gamma_{i,l}^{\text{host}}$ and $\gamma_{i,l}^{\text{local}}$ are applied to check if the global estimates from neighbor l are consistent with the host's global estimate and the host's local measurements, respectively; $\gamma_{i,l}^{\text{self}}$ checks whether the global estimates from the host's own distributed observer are consistent with the local measurements. They will be illustrated in the following. The resulting $\mathcal{DT}_{i,l}$ is later fused with the local trust to build the target/vehicle trust and ultimately contributes to the generalized trust (opinion) vector O_i in Subsection 3.5.

3.3.1 Consistency with host's global estimate:

To assess whether the global estimate from the target l is trustworthy, the host can compare it directly to its own global estimate. We use the Mahalanobis distance, which normalizes discrepancies based on covariance:

$$r_{i,l}^{(j)} = \left(\hat{x}_i^{(j)} - \hat{x}_l^{(j)}\right)^\top \Sigma_{\text{host}}^{-1} \left(\hat{x}_i^{(j)} - \hat{x}_l^{(j)}\right), \tag{4.20}$$

where Σ_{local} is the diagonal covariance matrix which captures variance and correlations between each state channel defined as

$$\Sigma_{\text{local}} = \text{diag}\left(\sigma_1^2, \sigma_2^2, \cdots, \sigma_n^2\right), \tag{4.21}$$

where n is the number of states in the vehicle, and σ_k^2 is the variance of discrepancies about the different states.

And then, add up these differences to get the total discrepancy score across the platoon as the following

$$R_{i,l} = \sum_{j=1}^{N} r_{i,l}^{(j)}. \tag{4.22}$$

A small $R_{i,l}$ means the global estimate from vehicle l is close to the host's, suggesting higher trustworthiness.

Based on the total discrepancy score across the platoon, the trust score can be converted to be

$$\gamma_{i,l}^{\text{host}} = \exp\left(-R_{i,l}\right). \tag{4.23}$$

where $\gamma_{i,l}^{\text{host}} \in (0, 1]$. A value near 1 means high trust (small discrepancy), while a value near 0 means low trust (large discrepancy).

3.3.2 Consistency with host's local measurement:

Host vehicle i has relative measurements to nearby vehicles (e.g., distance and relative velocity to a predecessor and a follower) and uses them to check if the distributed estimates received from target l are consistent with these measurements.

Let $y_{i,j}(t) \in \mathbb{R}^{m_{ij}}$ be the relative measurement of vehicle i to vehicle j (e.g., $y_{i,j}(t) = s_j(t) - s_i(t)$ for relative position).

The distributed estimate received from target l should satisfy, ideally:

$$H(\hat{x}_l^{(j)}(t) - \hat{x}_l^{(i)}(t)) \approx y_{i,j}(t).$$

where $H \in \mathbb{R}^{m_{ij} \times n}$ is the measurement matrix that extracts the relevant states (e.g., position) from the state vector.

Define the error between the predicted relative output and the local measurement as

$$\varepsilon_{i,l}^{(j)}(t) = H(\hat{x}_l^{(j)}(t) - \hat{x}_l^{(i)}(t)) - y_{i,j}(t), \tag{4.24}$$

Similar with the (4.20), by applying the covariance matrix Σ_{local}, we can get

$$E_{i,l}^{(j)}(t) = (\varepsilon_{i,l}^{(j)}(t))^\top \Sigma_{\text{local}}^{-1} \varepsilon_{i,l}^{(j)}(t). \tag{4.25}$$

Then, the total consistency error can be get by summing up the errors over all measurable vehicles:

$$E_{i,l}(t) = \sum_{j \in \mathcal{N}_i} E_{i,l}^{(j)}(t). \tag{4.26}$$

A higher $E_{i,l}(t)$ suggests that the distributed estimate of target l contradicts with local sensor relative measurements.

Convert the error into a trust factor as

$$\gamma_{i,l}^{\text{local}}(t) = \exp\left(-E_{i,l}(t)\right), \tag{4.27}$$

where $\gamma_{i,l}^{\text{local}}(t) \in (0, 1]$, with lower values indicating poor consistency.

3.3.3 Self-consistency with host's local measurement:

The host also checks whether its *own* global estimates produced by the distributed observer are consistent with its local relative measurements. Specifically, the host's global estimate should satisfy, ideally,

$$H(\hat{x}_i^{(j)}(t) - \hat{x}_i^{(i)}(t)) \approx y_{i,j}(t), \quad j \in \mathcal{N}_i.$$

Define the self-consistency error

$$\varepsilon_{i,\text{self}}^{(j)}(t) = H(\hat{x}_i^{(j)}(t) - \hat{x}_i^{(i)}(t)) - y_{i,j}(t), \tag{4.28}$$

and the corresponding aggregate score

$$E_{i,\text{self}}(t) = \sum_{j \in \mathcal{N}_i} (\varepsilon_{i,\text{self}}^{(j)}(t))^\top \Sigma_{\text{local}}^{-1} \varepsilon_{i,\text{self}}^{(j)}(t). \tag{4.29}$$

Then the self-consistency factor is defined as

$$\gamma_{i,l}^{\text{self}}(t) = \exp\left(-E_{i,\text{self}}(t)\right), \tag{4.30}$$

which is shared across all targets l (it depends only on the host's internal consistency at time t).

Remark 9. *We can estimated covariance Σ_{host} and Σ_{local} by collecting the differences in vehicle i from a window of recent received data from l in normal condition. This avoids the need for each node to transmit its own covariance.*

3.4 Target trust and temporal evolution

We can combine the global trust and the local trust to get one single trust of the target l by following :

$$\mathcal{T}_{i,l}(t) = \mathcal{LT}_{i,l} \cdot \mathcal{DT}_{i,l}. \tag{4.31}$$

$\mathcal{T}_{i,l}(t)$ indicates the quality of target l data at the time t , that will facilitate the decision-making process for the host vehicle.

Because this value can vary abruptly, a temporal-evolution mechanism is introduced to obtain a smoother, interpretable estimate.

3.4.1 Discretization into trust quality:

We define a five-level trust quality among $q = 5$ categories: *Unreliable*, *Poor*, *Acceptable*, *Good*, and *Excellent*. When the host vehicle i computes a trust value $\mathcal{T}_{i,l}(t) \in [0, 1]$ for a neighboring vehicle l, the result is discretized into a one-hot vector $r_{i,l}(t) \in \mathbb{R}^q$. For example, if $\mathcal{T}_{i,l}(t) = 0.72$, it falls into the "Good" category (4th level), so $r_{i,l}(t) = [0, 0, 0, 1, 0]^\top$.

3.4.2 Accumulated trust history:

Over time, the accumulated trust vector $R_{i,l}(t)$ for target vehicle l is updated using a weighted aggregation of past interactions:

$$R_{i,l}(t+1) = (1 - \mathcal{T}_{i,l}(t) \cdot \beta) \cdot R_{i,l}(t) + r_{i,l}(t), \tag{4.32}$$

where $\beta \in [0, 1]$ as a tunable weight controls how quickly past experiences decay.

Remark 10. *The choice of β is crucial for effectively capturing the dynamics of trust over time. A higher β places more emphasis on recent observations, allowing the trust score to adapt quickly to changes in behavior. Conversely, a lower β gives more weight to historical data, resulting in a smoother trust evolution that is less sensitive to transient fluctuations.*

3.4.3 Normalized trust distribution:

Finally, at the current step t (in the following, we omit t for simplifying), a normalized trust distribution vector $S_{i,l}(k)$ is computed by incorporating an a priori distribution α and a confidence parameter C:

$$S_{i,l}(k) = \frac{R_{i,l}(k) + C \cdot \alpha(k)}{C + \sum_{j=1}^{q} R_{i,l}(j)}, \tag{4.33}$$

which provides a probabilistic interpretation of the trustworthiness of the target vehicle l, blending historical observations with prior beliefs to produce a robust and adaptive trust estimate.

3.4.4 Expected trust level:

The target vehicle trust score $\mathcal{VT}_{i,l}$ is then calculated as:

$$\mathcal{VT}_{i,l} = \sum_{j=1}^{q} \frac{j-1}{q-1} \cdot S_{i,l}(j). \tag{4.34}$$

While $\mathcal{VT}_{i,l}$ provides a stable, pairwise evaluation between directly connected vehicles, a more global view is required for the observer to weigh all available information. This motivates the definition of a generalized trust vector, introduced next.

3.5 Generalized trust vector

The target trust value $VT_{i,l}$ obtained in the previous subsection quantifies how much the host vehicle i trusts a specific neighbor l. However, for the distributed observer to operate reliably, each vehicle must maintain an opinion about all other vehicles in the platoon, including those beyond its direct communication range. To achieve this, we define a generalized trust vector O_i that aggregates and propagates local trust information in a robust one-hop manner. For vehicle i, the vector is defined as

$$O_i = [O_i(1), O_i(2), \ldots, O_i(N)], \tag{4.35}$$

where O_i represents the current opinion of vehicle i regarding all platoon members, while $O_i(j) \in [0, 1]$ specifies vehicle i's current trust opinion toward vehicle j.

1) Self-Trust. Each vehicle assumes full confidence in its own state:

$$O_i(i) = 1, \tag{4.36}$$

2) Direct-Neighbor Trust. For every neighbor $j \in \mathcal{N}_i$ with which vehicle i exchanges data directly, the trust opinion is set as :

$$O_i(j) = VT_{i,j}, \tag{4.37}$$

where $VT_{i,j}$ is the smoothed target-trust value derived in the previous subsection.

3) One-Hop Trust Propagation. For any non-neighbor vehicle $j \notin \mathcal{N}_i \cup \{i\}$, the host vehicle i estimates

$O_i(j)$ using only information from its direct neighbors $\mathcal{N}_i$.

$$S_{i,j} = \Big\{ O_k(j) \mid k \in \mathcal{N}_i,\ VT_{i,k} > \theta_{\min},\ \delta_{i,k}(t) = 1 \Big\} \tag{4.38}$$

Here, $\theta_{\min}$ is the minimum acceptable trust threshold,

Each valid neighbor k is associated with a *credibility weight*

$$w_{i,k} = \frac{VT_{i,k}}{\sum_{p \in S_{i,j}} VT_{i,p}} \tag{4.39}$$

If the credible set $S_{i,j}$ is non-empty, the propagated opinion is computed using a *weighted median* operator:

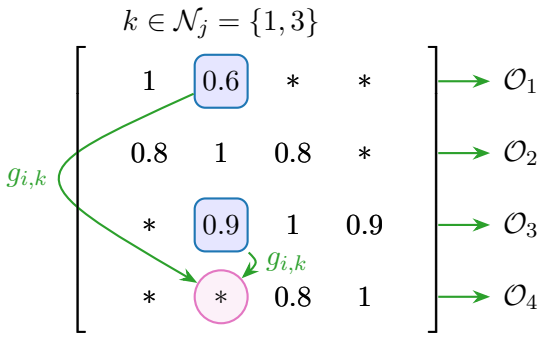
$$O_i(j) = \mathrm{wMed}\big(\{O_k(j)\}_{k \in S_{i,j}}, \{w_{i,k}\}_{k \in S_{i,j}}\big), \tag{4.40}$$

which limits the influence of outliers or manipulated values.

4) Index Distance-Based Fallback. If the credible set $S_{i,j}$ is empty, vehicle i resorts to a *host-centric distance weighting* of all its neighbors:

$$O_i(j) = \frac{\sum_{k \in \mathcal{N}_i} g_{i,k}\, O_k(j)}{\sum_{k \in \mathcal{N}_i} g_{i,k}}, \qquad g_{i,k} = \frac{1}{1 + |i - k|}. \tag{4.41}$$

where $|i - k|$ is the index distance between vehicles i and k. The closer a neighbor k is to host i, the larger its influence $g_{i,k}$. This fallback guarantees that the trust vector remains fully populated even when no verified propagated opinion exists.



Vehicle $i = 4$ doesn't have the trust for the vehicle $j = 2$

Figure 4.3: Weight based distance for $\mathcal{O}_{i,j}^{\text{dis}}$.

6) Final Expression. Combining all cases, the complete definition of the generalized trust vector is

$$O_i(j) = \begin{cases} 1, & j = i, \\ VT_{i,j}, & j \in \mathcal{N}_i, \\ \mathrm{wMed}(\{O_k(j)\}, \{w_{i,k}\}), & S_{i,j} \neq \varnothing, \\ \dfrac{\sum_{k\in\mathcal{N}_i} g_{i,k}\, O_k(j)}{\sum_{k\in\mathcal{N}_i} g_{i,k}}, & S_{i,j} = \varnothing. \end{cases} \tag{4.42}$$

This generalized representation allows each vehicle to maintain a resilient, full-platoon trust profile that supports the adaptive observer weight design described in the following section.

4 Weight Design for observer based on the trust

The trust framework provides each vehicle with a set of quantified opinions $\mathcal{O}_{i,j}$ representing the reliability of information from other vehicles. This section shows how these values are used to define adaptive observer weights $w_{il}^{j}(t)$ and establishes conditions for stability of the estimation error dynamics.

4.1 Weight based on the trust

Based on the virtual communication graph $\bar{\mathcal{G}}^{(j)}$ as that includes the local observer node 0, each vehicle i identifies its legitimate neighbors at time t:

$$\begin{aligned} \mathcal{LN}_i^{(j)}(t) = &\left\{ l \in N_i \mid O_i(l,t) \geq \theta_{\min} \right\} \\ &\cup \left(\{0\} \text{ if } \mathcal{LT}_{i,j}(t) \geq \theta_{\min} \right) \end{aligned} \tag{4.43}$$

To limit the influence of any single neighbor, a normalization factor is introduced.

$$n_{il}^{(j)}(t) = \max\left\{ \kappa, \left| \mathcal{LN}_i^{(j)}(t) \right| + 1 \right\} \geq 1, \forall l \in \mathcal{LN}_i^{(j)} \tag{4.44}$$

where $\kappa > 0$ is the parameter to limit the maximum influence from the neighbors of ith vehicle. Then, the observer's weight gain can be chosen as

$$w_{il}^{(j)}(t) = \begin{cases} \dfrac{1}{n_{il}^{(j)}(t)}, & l \in \mathcal{LN}_i^{(j)}(t) \\ 0, & \text{otherwise.} \end{cases} \tag{4.45}$$

According to the weight gain, the communication graph will be switched depending on the time. The following assumption is required.

Assumption 10 (*Jointly connected graph). There exists a scalar constant $T > 0$ such that for all $t \geq 0$, the union graph $\cup_{[t,t+T]} \bar{\mathcal{G}}^{(j)}(t)$ contains a spanning tree.*

Assumption 10 means that for all $t \geq 0$, every node $i \in \mathcal{V}$ is reachable from node "0 in the union graph $\cup_{[t,t+T]}\bar{\mathcal{G}}^{(j)}(t)$. This ensures the connectivity of the proposed switching communication topology, which is required for our observer design method. For more details on this assumption, we refer the reader to [**hao2024eventtriggered**].

4.2 Stability of the error

In this section, we explore the conditions that guarantee the stability of the estimation error dynamics of the proposed observer given in (4.5). The error dynamics are given as follows:

$$e_0^{(j)}(t+1) = (A_\mathrm{b} - F_j C_\mathrm{b})\, e_0^{(j)}(t) + d_0^j, \tag{4.46}$$

$$\begin{aligned} e_{\mathcal{DO}}^{(j)}(t+1) = &\left(w_0^{(j)} \otimes A_\mathrm{b}\right) e_0^{(j)}(t) \\ &+ \left(\left(I_N - \bar{\mathcal{L}}^{(j)}\right) \otimes A_\mathrm{b}\right) e_{\mathcal{DO}}^{(j)}(t) + d_{\mathcal{DO}}^j, \end{aligned} \tag{4.47}$$

with

$$\bar{\mathcal{L}}^{(j)} = \begin{bmatrix} \sum_{l\in\{\mathcal{N}_1\cup\{0\}\}} w_{1l}^{(j)} & -w_{12}^{(j)} & \cdots & -w_{1N}^{(j)} \\ -w_{21}^{(j)} & \sum_{l\in\mathcal{N}_2} w_{2l}^{(j)} & \cdots & -w_{1N}^{(j)} \\ \vdots & \vdots & \ddots & \vdots \\ -w_{N1}^{(j)} & -w_{N2}^{(j)} & \cdots & \sum_{l\in\mathcal{N}_N} w_{Nl}^{(j)} \end{bmatrix}$$

$$\begin{aligned} d_0^j &= -\Delta_j, \\ d_{DO}^j &= (I_N \otimes B_b)(\hat{u}_j - u_j) - (1_N \otimes \Delta_j) \\ e_{\mathcal{DO}}^{(j)}(t) &= \mathrm{col}\left(e_1^{(j)}(t), \cdots, e_N^{(j)}(t)\right) \end{aligned}$$

where $w_0^{(j)} = \mathrm{col}\left(w_{10}^{(j)}, \cdots, w_{N0}^{(j)}\right)$ is a vector describing the connection between the local observer $(\mathcal{LO}_j)$ and other distributed observers $(\mathcal{DO}_i)$, $i \in \mathcal{V}$; $\bar{\mathcal{L}}^{(j)}$ is the virtual weighted Laplacian matrix of the virtual graph $\bar{\mathcal{G}}^{(j)}$.

Before stating the main theorem of this section, we need to make the following assumption which is necessary to develop an ISS bound on the estimation errors.

Assumption 11. *The disturbance vectors $d_0^j(t)$ and $d_{\mathrm{DO}}^j(t)$ are bounded. That is, there exist $\bar{d}_0 \geq 0$ and $\bar{d}_{\mathrm{DO}} \geq 0$ such that $|d_0^j(t)| \leq \bar{d}_0$ and $|d_{\mathrm{DO}}^j(t)| \leq \bar{d}_{\mathrm{DO}}$, for all $t \geq 0$.*

Now, we are ready to state the following theorem, which provides the main sufficient conditions ensuring the stabilization of the error system (4.46)–(4.47).

Theorem 4. *Consider a vehicle platoon (4.4) over a directed weighted communication network $\bar{\mathcal{G}}^{(j)}$. Let (4.5) be the corresponding distributed observer. Assume that the following conditions hold true:*

1. *For all $j \in \mathcal{V}$, there exists F_j such that the matrix $(A_b - F_j C_b)$ is Schur stable.*
2. *For all $i \in \mathcal{V}$, $w_{i0}^{(j)} \geq 0$ and $w_{il}^{(j)} \geq 0$ satisfy*

$$\sum_{l\in\bar{\mathcal{G}}^{(j)}} w_{il}^{(j)} \leq 1. \tag{4.48}$$

3. *There exists* $\alpha \in (0,1)$ *and a matrix norm* $\|\cdot\|_*$ *such that, uniformly over all admissible switches,*

$$\| \left(I_N - \bar{\mathcal{L}}^{(j)} \right) \otimes A_b \|_* \leq \alpha < 1. \tag{4.49}$$

Then, the estimation error (4.46)–(4.47) is ISS. In particular, there exist constants $c_0, c_1 > 0$ *and* $\lambda \in (0,1)$ *such that*

$$\|e_0^{(j)}(t)\| \leq \lambda^t \|e_0^{(j)}(0)\| + c_0 \bar{d}_0, \tag{4.50}$$

$$\|e_{DO}^{(j)}(t)\| \leq \lambda^t \|e_{DO}^{(j)}(0)\| + c_1 \Big(\|e_0^{(j)}(0)\| + \bar{d}_0 + \bar{d}_{DO} \Big). \tag{4.51}$$

Intuition (cascade + switching). Intuitively, the local observer $\mathcal{LO}_j$ stabilizes the *self-estimation* error $e_0^{(j)}$ because $A_{\ell,j} \triangleq A_\mathrm{b} - F_j C_\mathrm{b}$ is Schur (Condition 1), so $e_0^{(j)}$ is ISS with respect to the bounded disturbance d_0^j. The distributed layer then behaves as a switched consensus-like filter driven by $e_0^{(j)}$ and d_{DO}^j:

$$e_{\mathrm{DO}}^{(j)}(t+1) = A_{\mathrm{DO}}^{(j)}(t) e_{\mathrm{DO}}^{(j)}(t) + B_0^{(j)}(t) e_0^{(j)}(t) + d_{\mathrm{DO}}^j(t),$$

with $A_{\mathrm{DO}}^{(j)}(t) = (I_N - \bar{\mathcal{L}}^{(j)}(t)) \otimes A_\mathrm{b}$ and $B_0^{(j)}(t) = w_0^{(j)}(t) \otimes A_\mathrm{b}$. The trust mechanism enforces a uniformly jointly connected virtual graph (Assumption 10) and bounded nonnegative weights (Condition 2); combined with the uniform contraction bound in Condition 3 ($\|A_{\mathrm{DO}}^{(j)}(t)\|_* \leq \alpha < 1$ for all admissible switches), this prevents switching from destroying stability and yields an ISS cascade.

A sketch of the proof. The argument relies on Assumptions 10 and 11 and on the cascade structure of (4.46)–(4.47).

Step 1 (local loop). Fix $j \in \mathcal{V}$ and set $A_{\ell,j} \triangleq A_\mathrm{b} - F_j C_\mathrm{b}$. Under Condition (1), $A_{\ell,j}$ is Schur, so there exist $c_{\ell,j} > 0$ and $\lambda_{\ell,j} \in (0,1)$ such that $\|A_{\ell,j}^t\| \leq c_{\ell,j} \lambda_{\ell,j}^t$ for all $t \geq 0$. Unrolling (4.46) yields

$$e_0^{(j)}(t) = A_{\ell,j}^t e_0^{(j)}(0) + \sum_{k=0}^{t-1} A_{\ell,j}^{t-1-k} d_0^j(k),$$

and by Assumption 11 we obtain an ISS estimate of the form

$$\|e_0^{(j)}(t)\| \leq c_{\ell,j} \lambda_{\ell,j}^t \|e_0^{(j)}(0)\| + \frac{c_{\ell,j}}{1 - \lambda_{\ell,j}} \bar{d}_0.$$

Step 2 (distributed loop under switching). Rewrite (4.47) as

$$e_{\mathrm{DO}}^{(j)}(t+1) = A_{\mathrm{DO}}^{(j)}(t) e_{\mathrm{DO}}^{(j)}(t) + B_0^{(j)}(t) e_0^{(j)}(t) + d_{\mathrm{DO}}^j(t),$$

with $A_{\mathrm{DO}}^{(j)}(t) = (I_N - \bar{\mathcal{L}}^{(j)}(t)) \otimes A_\mathrm{b}$ and $B_0^{(j)}(t) = w_0^{(j)}(t) \otimes A_\mathrm{b}$. Condition (3) provides a uniform contraction $\|A_{\mathrm{DO}}^{(j)}(t)\|_* \leq \alpha < 1$ for all admissible switches. Using the induced norm $\|\cdot\|_*$ and unrolling

the recursion gives

$$\|e_{\mathrm{DO}}^{(j)}(t)\|_* \le \alpha^t \|e_{\mathrm{DO}}^{(j)}(0)\|_* + \sum_{k=0}^{t-1} \alpha^{t-1-k} \Big(\|B_0^{(j)}(k)\|_* \|e_0^{(j)}(k)\|_* + \|d_{\mathrm{DO}}^{j}(k)\|_* \Big).$$

Condition (2) (nonnegative weights with row-sum bounded by 1) implies $\|w_0^{(j)}(t)\|$ is uniformly bounded, hence $\|B_0^{(j)}(t)\|_*$ is uniformly bounded as well. Combining this inequality with the bound from Step 1 and Assumption 11 yields the ISS bounds stated in the theorem. □

A sketch of the proof. The proof is straightforward and relies on Assumptions 10 and 11. Its simplicity stems from the fact that the coupled system (4.46)–(4.47) has a cascade structure. Condition (1) guarantees that (4.46) is ISS. Condition (2), ensured by Assumption 10, makes it possible to verify Condition (3) by applying the Gershgorin theorem. Finally, Condition (3), together with (2), ensures that (4.47) is ISS. □

Remark 11. *The conditions of Theorem 4 can be verified using a Lyapunov stability approach. This analysis yields explicit expressions for the matrix F_j and the constants c_0, c_1, α, and λ, depending on the selected Lyapunov functions. Since the overall system exhibits a cascade structure, system (4.46) can be analyzed independently from system (4.47), which effectively leads to a separation principle. Due to the page limit constraint, these detailed results are not presented as a separate theorem.*

Remark 12. *(Limitations of the Stability Guarantees). The ISS guarantees of Theorem 1 rely critically on (i) bounded influence of each neighbor ensured by trust-based weight normalization, and (ii) uniform joint connectivity of the virtual graph, which guarantees repeated anchoring through the local observer node.*

These conditions may be violated under the following scenarios:

- *Stealthy Byzantine attacks, where adversarial vehicles generate physically consistent but biased data, maintaining high trust and inducing convergence to an incorrect equilibrium.*
- *Colluding attacks, in which multiple compromised vehicles mutually reinforce false estimates, undermining trust propagation mechanisms.*
- *Persistent DoS or network partition attacks, which permanently disconnect vehicles from anchoring local observers, rendering absolute state reconstruction impossible.*

These scenarios correspond to fundamental identifiability limits of distributed estimation and cannot be addressed without additional assumptions such as cryptographic authentication, majority-honest agents, or infrastructure-based anchoring.

5 Simulation Results

This section evaluates the performance of the proposed trust-aware distributed observer for a connected vehicle platoon under various cyberattack scenarios. The main objectives of the simulation are to:

- Assess the observer resilience to falsified and delayed information.
- Analyze the evolution of trust scores for detecting and isolating malicious nodes.
- Quantify estimation performance degradation under different types of attacks.

5.1 Simulation setup

The simulation considers a platoon of four fully connected vehicles, where each vehicle exchanges information with all others through vehicle-to-vehicle (V2V) communication. All vehicles share similar nominal parameters, with small modeling variations treated as uncertainties.The proposed trust-aware distributed observer runs onboard each vehicle to estimate states and evaluate the reliability of shared data. Table 4.2 summarizes the main parameters used for the vehicle model and the trust framework.

Table 4.2: Main simulation parameters.

Parameter	**Value**	**Description**
T_s	0.1 s	Sampling time
τ	1.5 s	Engine lag constant
w_v, w_d, w_a	1.2, 2.0, 0.9	Local trust weights
$\beta_{\text{trust}}, C, k$	0.5, 0.2, 5	Trust decay and regularization
d_{norm}	15 m	Normalization constants
$\theta_{\min}$	0.5	Minimum trust threshold
N	4	Number of vehicles

5.2 Attack scenarios

To assess the robustness of the vehicle platooning system, six attack cases are defined, as summarized in Table 4.3. All attacks are launched by the leader vehicle and broadcast to every follower in the platoon. Each attack is active during the interval $[10, 15]$ s. For reproducibility , the observer is evaluated under the assumption that the follower controllers are known, thereby isolating and emphasizing the observer capability to detect and mitigate malicious data. The parameter p denotes the probability of the attack or fault occurrence.

Table 4.3: Six attack cases in the scenario.

Case	**Type**	**Target**	**Parameters**
1	Bias	Position	bias = -5 m
2	Faulty	Position	int. = 10, $p = 0.3$
3	Bias	Velocity	bias = -2 m/s
4	Faulty	Velocity	int. = 2.5, $p = 0.3$
5	Faulty	Acceleration	int. = 1.0, $p = 0.3$
6	Drop	All	$p_{\text{drop}} = 0.5$

5.3 Results

Figure 4.4 presents a heatmap of the normalized impact scores for each vehicle across all six attack cases. The normalized impact score is obtained by dividing the raw combined error by the maximum observed

combined error across all vehicles and attack scenarios. This normalization enables a clear comparison of the relative severity of each attack on individual vehicles.

To quantitatively assess the influence of different cyberattack scenarios, both raw combined errors and normalized impact scores were analyzed. The raw combined error provides a direct measure of deviation in distance, velocity, and acceleration estimation, whereas the normalized impact score offers a more balanced indicator of overall system degradation relative to nominal operating performance.

Across all cases, the DoS attack (Case 6) exhibited the highest normalized impact score (0.872), confirming it as the most severe attack scenario. This high value reflects a significant degradation in estimation accuracy and control performance during the 5-second attack window. In contrast, the velocity bias attack (Case 3) with a bias of -2 m/s produced the lowest normalized impact score (0.240), indicating minimal influence on system stability and tracking capability.

From the vehicle perspective, V4 was identified as the most affected vehicle, reaching the maximum normalized impact score (0.872), whereas vehicle V2 demonstrated the highest resilience, with the lowest average impact score (0.362). This suggests that V4 local dynamics or communication dependencies make it more sensitive to external disruptions, while V2 maintains more stable performance under varying attack conditions.

Overall, the relatively low mean normalized impact scores across all cases indicate that the proposed trust-aware distributed observer effectively mitigates diverse types of cyberattacks and maintains accurate state estimation for each vehicle in the platoon.

In Figure 4.5, the trust score evolution for the DoS attack (Case 6) is illustrated. During the attack period $[10, 15]$ s, these 3 vehicles (V2, V3, V4) exhibit a noticeable drop in trust values below the threshold of 0.5, indicating successful detection of the communication disruption. Once the attack ceases, the trust scores gradually recover, demonstrating the system ability to restore confidence and resume normal operation. This behavior confirms the effectiveness of the proposed trust mechanism in identifying and isolating malicious data in real time.

5.4 Distributed State Estimation for Platoon Control

In this section, we present a platoon control framework that leverages distributed state estimation to enhance longitudinal control performance. The approach adopts a constant time headway spacing (CTHS) policy to maintain small inter-vehicle distances, improving traffic flow and safety. Two controllers are employed: an inner controller, based on the Intelligent Driver Model (IDM), for local longitudinal control using estimates from the local observer, and an outer controller, the Cooperative Adaptive Cruise Control (CACC), for distributed coordination using estimates from the distributed observer. These controllers are combined to form a robust final control strategy, balancing local responsiveness and platoon-wide consensus.

5.4.1 Notation

5.4.2 Inner/Local Controller: Intelligent Driver Model (IDM)

The IDM is a car-following model that computes a vehicle's desired acceleration based on its own state and the relative state to its predecessor. Here, it serves as the inner controller for local longitudinal

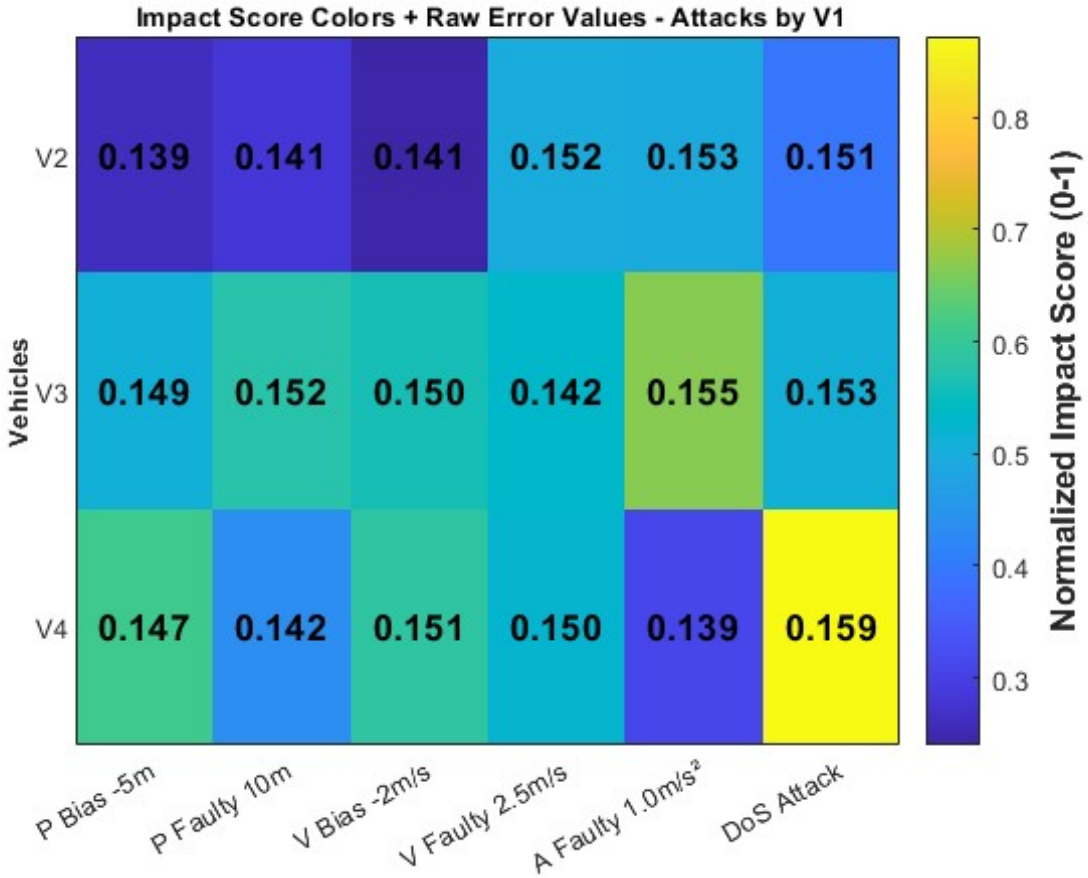


Figure 4.4: Heatmap of normalized impact scores for all vehicles and attack cases.

control, utilizing the local observer's estimate of the vehicle's own state, $\hat{x}_0^{(i)}$. The IDM acceleration for vehicle i is given by:

$$a_{\text{IDM},i} = \alpha \left[1 - \left(\frac{\hat{v}_0^{(i)}}{v_0} \right)^{\delta} - \left(\frac{s^*(\hat{v}_0^{(i)}, \Delta v_{i-1,i})}{s_{i-1,i}} \right)^2 \right], \tag{4.52}$$

where:

- $\hat{v}_0^{(i)}$ is the estimated velocity of vehicle i from $\mathcal{LO}_i$.
- $s_{i-1,i} = s_{i-1} - s_i - L$ is the actual relative distance to the predecessor, with L as the vehicle length (assuming direct sensor measurement for simplicity, as the query specifies IDM uses only local observer estimates, but relative distance typically requires predecessor data).
- $\Delta v_{i-1,i} = v_{i-1} - \hat{v}_0^{(i)}$ is the actual relative velocity, using the predecessor's true velocity v_{i-1} from sensors.
- $s^*(\hat{v}_0^{(i)}, \Delta v_{i-1,i}) = s_0 + T\hat{v}_0^{(i)} + \frac{\hat{v}_0^{(i)} \Delta v_{i-1,i}}{2\sqrt{\alpha\beta}}$ is the desired minimum gap.
- α, β, δ, v_0, s_0, and T are model parameters: maximum acceleration, comfortable deceleration, free-flow exponent, desired velocity, minimum gap, and time headway, respectively.

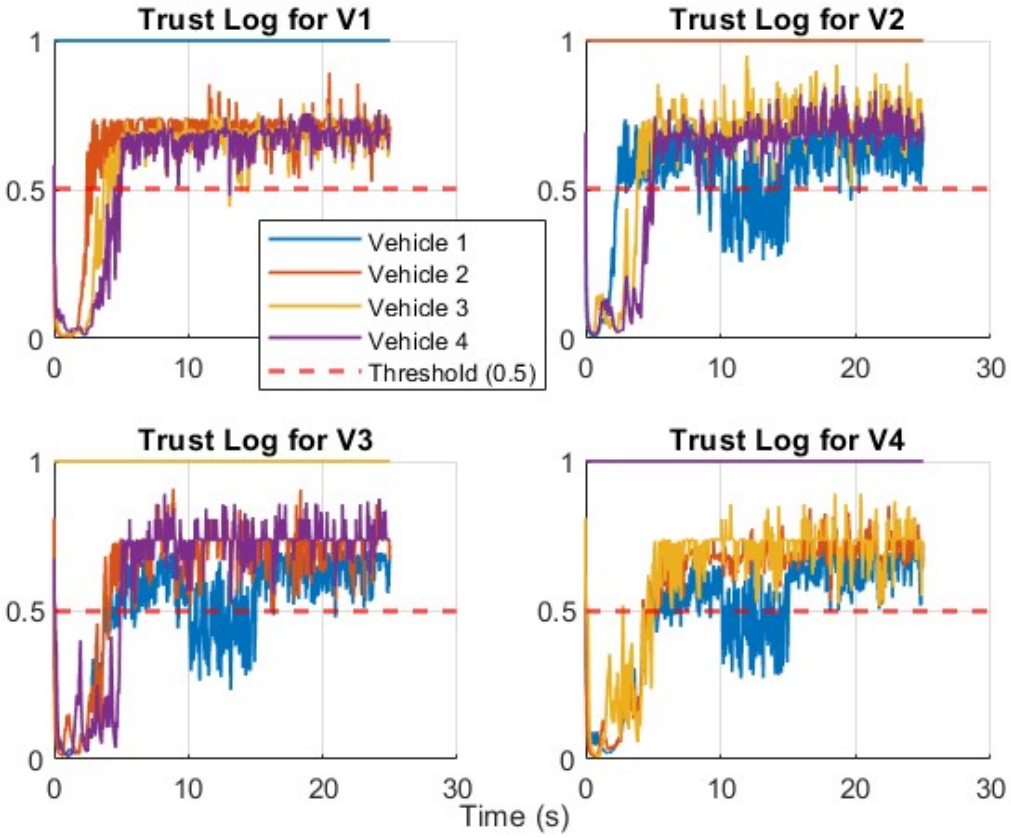


Figure 4.5: Trust score evolution during DoS attack (***Case 6***).

Since $\mathcal{LO}_i$ provides only $\hat{x}_0^{(i)}$ and not the predecessor's state, we assume vehicle i uses sensor data (e.g., radar) for $s_{i-1,i}$ and $\Delta v_{i-1,i}$, consistent with standard IDM implementations, while adhering to the query's directive to use local observer estimates for the vehicle's own state.

5.4.3 Cooperative Controller: Cooperative Adaptive Cruise Control (CACC)

The CACC enhances platoon coordination by leveraging information from multiple preceding vehicles via the distributed observer $\mathcal{DO}_i$. For vehicle i, the CACC control input is:

$$u_{\text{CACC},i}(k) = \sum_{j=1}^{i-1} [\kappa_s \left(\hat{s}_i^{(j)}(k) - \hat{s}_0^{(i)}(k) - d_{i,j}(k) \right) + \kappa_v \left(\hat{v}_i^{(j)}(k) - \hat{v}_0^{(i)}(k) \right) + \kappa_a \left(\hat{a}_i^{(j)}(k) - \hat{a}_0^{(i)}(k) \right)] \tag{4.53}$$

where:

- $\hat{s}_i^{(j)}(k)$, $\hat{v}_i^{(j)}(k)$, $\hat{a}_i^{(j)}(k)$ are the estimated position, velocity, and acceleration of vehicle j from $\mathcal{DO}_i^{(j)}$.
- $\hat{s}_0^{(i)}(k)$, $\hat{v}_0^{(i)}(k)$, $\hat{a}_0^{(i)}(k)$ are the estimated position, velocity, and acceleration of vehicle i from $\mathcal{LO}_i$ (included for consistency, though the query specifies distributed observer estimates for CACC).

- $d_{i,j}(k) = d + h\hat{v}_0^{(i)}(k)$ is the desired spacing between vehicles i and j, with d as a constant gap and h as the time headway (for $j = i - 1$, this aligns with the CTHS policy).
- κ_s, κ_v, κ_a are control gains for position, velocity, and acceleration errors, respectively.

This formulation ensures that vehicle i adjusts its behavior based on the estimated states of all preceding vehicles, promoting consensus and stability across the platoon.

5.4.4 Final Controller

The final control input integrates the IDM and CACC controllers to balance local and cooperative objectives. The target acceleration is:

$$a_{\text{target},i} = (1 - \gamma(t))a_{\text{IDM},i} + \gamma(t)u_{\text{CACC},i}, \tag{4.54}$$

where $\gamma(t) \in [0, 1]$ is a tuning parameter that is influenced by the opinion score mentioned in Section 3. For this context, we choose $\gamma(t) = \min(\mathcal{O}_i)$. To prevent abrupt changes of the mixing 2 type controller, a first-order filter is applied:

$$u_i(t) = u_i(t-1) + \tau_f(a_{\text{target},i} - u_i(t-1)), \tag{4.55}$$

where τ_f is the filter time constant.

5.4.5 Expected Distance (Spacing)

To validate the control strategy, we derive the expected steady-state spacing:

- **IDM Steady-State Spacing**: When $a_{\text{IDM},i} = 0$ and $\Delta v_{i-1,i} = 0$:

$$1 - \left(\frac{\hat{v}_0^{(i)}}{v_0}\right)^{\delta} = \left(\frac{s_0 + T\hat{v}_0^{(i)}}{s_{i-1,i}}\right)^2,$$

  yielding:

$$s_{i-1,i} = \frac{s_0 + T\hat{v}_0^{(i)}}{\sqrt{1 - \left(\frac{\hat{v}_0^{(i)}}{v_0}\right)^{\delta}}}.$$

- **CACC Steady-State Spacing**: For the CTHS policy, when the platoon reaches consensus (all velocities equal), the spacing between vehicle i and $i - 1$ is:

$$\boxed{s_i = d + \frac{h\,\hat{v}_0^{(i)}}{i-1}\,.}$$

These expressions allow comparison with simulation results, assessing the controllers' ability to maintain desired spacing under various conditions, including cyber-attacks.

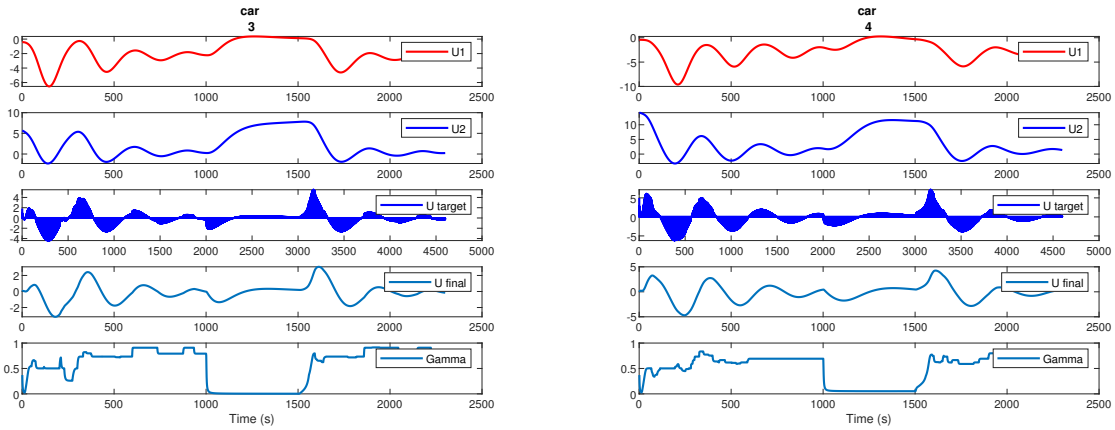


Figure 4.6: Controller of vehicle 2 and 3.

In figure 4.6, we can see the control input of vehicle 2 and 3, which is the acceleration of the vehicle. Smoothly switched between 2 controllers, and no abrupt change in the control input.

In figure 4.7, we can see the relative state of the vehicle in platoon, which is the difference between the position of the vehicle and the position of its predecessor. That prouve even in attack scenario, the vehicle can keep good distance No accident or crash happen in the platoon, which is a good sign of the robustness of the platoon control framework. Also that show the expected distance between the vehicle and its predecessor is maintained, which is the desired spacing in the platoon.

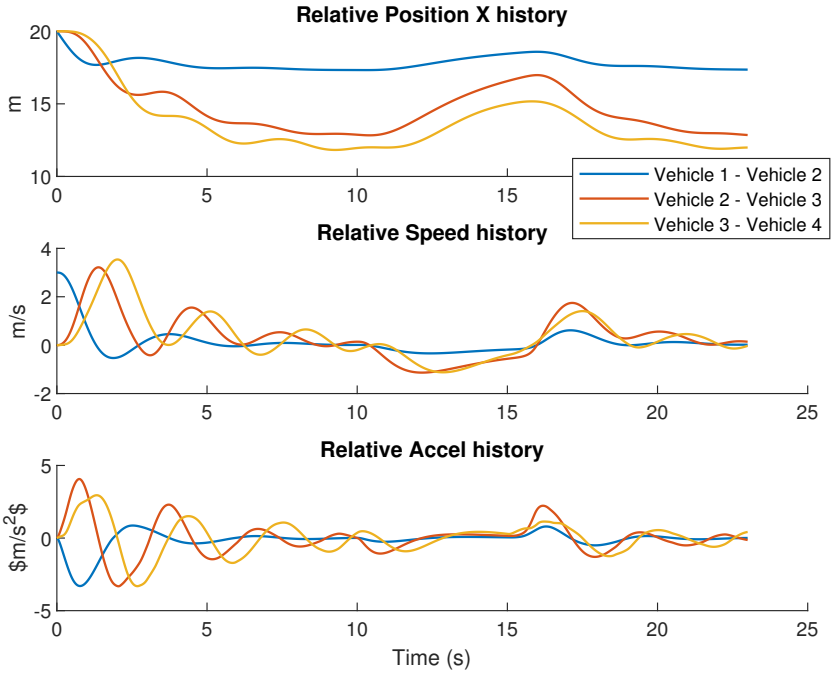


Figure 4.7: Relative state of the vehicle in platoon

6 Conclusion and Future Work

This paper presented a distributed observer-based platoon control framework capable of maintaining stability and safety under cyberattacks. The approach combines local and distributed observers with a trust evaluation mechanism to assess the reliability of inter-vehicle data, ensuring robust operation even with compromised nodes.

6.1 Open Problems and Future Directions

The analysis in this chapter focuses on stability and boundedness (ISS) under bounded disturbances and a trust-driven switching topology. Several open problems remain before such architectures can be made both sharper (less conservative) and more resilient in practice:

- **Matrix-valued (channel-wise) trust weights.** In (4.5)–(4.45), the coupling weights $w_{il}^{(j)}(t)$ are scalar and therefore apply the same attenuation to all state channels. A promising extension is to replace them with *matrix* gains $W_{il}^{(j)}(t) \in \mathbb{R}^{n_x \times n_x}$ (e.g., diagonal or block-diagonal), allowing the trust mechanism to down-weight only the corrupted channels (e.g., acceleration) while preserving reliable channels (e.g., position). This leads to a matrix-weighted virtual Laplacian and stability conditions of the form $\|\mathcal{A}_{\mathrm{DO}}^{(j)}(t)\|_* \leq \alpha < 1$ for a suitably defined induced norm, or equivalently LMI-type contraction constraints on the lifted (block) error dynamics.

- **Trust/detection delay and rollback mechanisms.** In realistic networks, trust values are computed from time windows and may be delayed; consequently, false data can enter the distributed observer before a node is flagged, degrading the estimate. Moreover, abruptly cutting a node can temporarily reduce information flow and worsen estimation during the transition. A natural direction is to

equip each host with a *rollback buffer*: store a fixed window of past estimates and received packets. When the trust of a node drops below threshold, roll back to time $t - T_{\mathrm{w}}$ (window length) and re-propagate the observer using (i) only trusted neighbors and (ii) model-based prediction/local-observer anchoring to bridge the missing information. This resembles fixed-lag smoothing with trust-aware data rejection and could mitigate transient corruption while preserving stability, provided the reset/rollback map is bounded and updates satisfy a dwell-time or bounded-variation condition.

Contamination rollback (compact recipe). If a neighbor $l^\star$ is flagged at time t, undo the last K steps by keeping a buffer of the *innovation terms* appearing in (4.5b). For each step k, store $\hat{x}_i^{(j)}(k)$ and

$$\eta_{il}^{(j)}(k) \triangleq w_{il}^{(j)}(k)\big(\hat{x}_l^{(j)}(k) - \hat{x}_i^{(j)}(k)\big), \quad l \in \mathcal{N}_i,$$
$$\eta_{i0}^{(j)}(k) \triangleq w_{i0}^{(j)}(k)\big(\hat{x}_0^{(j)}(k) - \hat{x}_i^{(j)}(k)\big),$$

as well as the input term $\hat{u}_j(k)$. When $l^\star$ is flagged, set $x \leftarrow \hat{x}_i^{(j)}(t-K)$ and replay for $k = t-K, \dots, t-1$ using the same observer update but excluding $l^\star$ (or a malicious set $\mathcal{M}$):

$$x \leftarrow A_{\mathrm{b}}\Big(x + \sum_{l \in \mathcal{N}_i \setminus \mathcal{M}} \eta_{il}^{(j)}(k) + \eta_{i0}^{(j)}(k)\Big) + B_{\mathrm{b}}\hat{u}_j(k),$$

and finally set $\hat{x}_i^{(j)}(t) \leftarrow x$.

- **Data-driven trust and learned uncertainty (with model-based safety layer).** Machine learning can be used to *learn the mapping from signals to reliability*, while keeping the stability guarantee in the model-based layer by enforcing bounded weights. Concretely: (i) learn a calibrated trust score or anomaly probability from a feature vector built from innovation/residual signals (e.g., $\hat{x}_l^{(j)} - \hat{x}_i^{(j)}$, $\varepsilon_{i,l}^{(j)}$, packet age/loss, and consistency indicators), using change-point detection, self-supervised prediction, or lightweight sequence models; (ii) learn state-dependent covariance/uncertainty (heteroscedastic models) to adapt Σ_{local} and thresholding; and (iii) implement the learned output only through a saturation/projection step that maps it to admissible weights satisfying Condition (2) and preserving contraction in Condition (3).

Future work will also focus on enhancing the trust mechanism by coupling it more closely with the controller, enabling trust evaluation for non-neighboring vehicles, and improving adaptability under dynamic communication topologies. We will also investigate machine-learning based trust estimation.

Chapter 5

Vehicle experiments and Board electronic

Objectives

This chapter reports the experimental validation workflow and the implementation results obtained on the Quanser QCar2/QLabs platform. It first presents the virtual-to-real experimental methodology, the platform architecture (sensing, perception, V2V communication, software stack), and the longitudinal/lateral control loops used to reproduce cooperative driving. Then, it details the observer and trust-aware distributed estimation pipeline, and summarizes the main results under nominal conditions and under representative cyber/communication attack scenarios. Finally, this chapter introduces the custom embedded electronic board designed to host the proposed estimation and trust mechanisms on a real vehicle, and discusses the firmware validation and the integration roadmap.

Contents

1 Vehicle Experiments

1.1 Experimental methodology: from virtual validation to real trials

The experimental validation strategy follows a modular pipeline centered on (i) platoon modeling, (ii) cooperative control, (iii) distributed observation, and (iv) incremental validation on a realistic platform. In practice, the workflow is decomposed into five phases:

- **Phase 1 – Environment setup:** configuration of the virtual environment under Ubuntu and deployment of the Quanser Virtual Environment container to connect QLabs and QCar2.
- **Phase 2 – V2V communication and software architecture:** implementation of inter-vehicle communication via UDP and development of a modular Python architecture.
- **Phase 3 – Longitudinal and lateral control:** two complementary control loops are implemented: (i) longitudinal control to regulate speed and inter-vehicle distance, and (ii) lateral control to track the road/trajectory.
- **Phase 4 – Observer and trust system:** development of a distributed observer to estimate the fleet states even in presence of delays and information losses; integration of a trust system that weights the reliability of received data and dynamically adapts the fusion mechanism.
- **Phase 5 – Full integration and real-time validation:** complete integration and real-time testing in QLabs; the transfer to physical scenarios on real QCar2 is prepared as a next step.

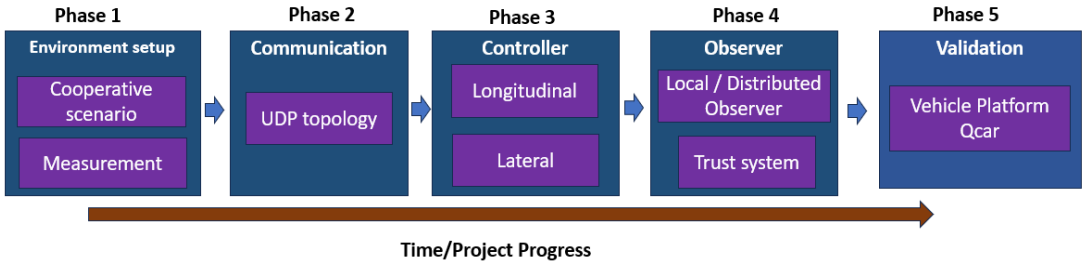


Figure 5.1: Modular Experimental Phases.

The objective of the real-trial preparation is to design an *embedded intelligent soft sensor* able to estimate non-measured quantities and detect disturbances related to cyberattacks and/or sensor faults. This embedded soft sensor associates perception measurements to surrounding vehicles, reconstructs critical variables, and distinguishes physical anomalies from malicious perturbations.

Beyond the modular phases described above, the experiments rely on a coordinated workflow that links code development, simulation, and real-world testing. A containerized development environment is used to write and package the Python control and estimation code. Once a new controller or observer is ready, the container sends the code to a Graphical Ground Station. This application acts as the central hub for the experiments: it runs the cooperative control and planning algorithms, visualizes trajectories and state estimates in real time, and forwards commands to the test platform. When working with the QLabs simulator, the ground station interfaces with the QLabs Virtual Environment, which is a digital

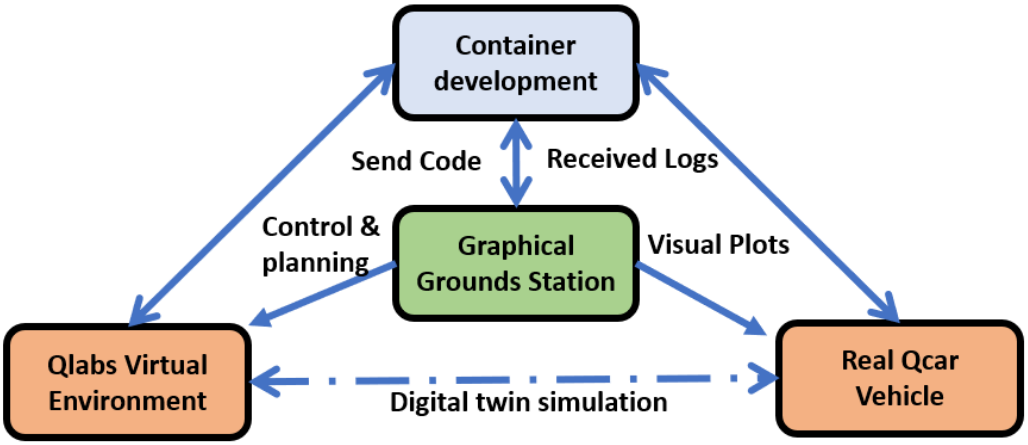


Figure 5.2: Experimental workflow for the virtual validation on QLabs/QCar2 and preparation of real trials.

twin of the physical QCar platform. The virtual QCar behaves the same way as the real hardware and exposes the same sensors and actuators, so control software can be measured and tested in simulation before deployment.

During a simulation trial, the ground station sends high-level control commands (e.g., desired speed and spacing) to the virtual environment and receives simulated sensor data and vehicle states. This allows the researcher to verify the platoon behaviour, tune the cooperative adaptive cruise control and look-ahead steering loops, and adjust observer parameters. Because the digital twin mirrors the physical self-driving car studio, the same code can then be executed on the real QCar with minimal changes. In physical tests, the ground station sends visual plots and user commands to the vehicle while logging the on-board measurements, estimated distances and trust indicators. The logs collected during real trials are sent back to the containerized development environment, where they are analysed to refine the models and software. This feedback loop—code development $\rightarrow$ ground-station execution $\rightarrow$ virtual trials $\rightarrow$ physical trials $\rightarrow$ log analysis—enables safe, incremental validation: controllers and observers are first validated in a realistic digital twin, then applied to the real vehicle. Bridging and blending code between the virtual and physical platforms allows researchers to explore new scenarios and behaviours while ensuring that the resulting algorithms are robust when transferred to hardware.

2 Quanser QCar2/QLabs experimental platform

2.1 Virtual environment and scenario definition

The QLabs environment is configured with two connected vehicles: a *leader* vehicle tracking a predefined closed-loop trajectory and a *follower* vehicle reproducing cooperative platoon behavior. The leader trajectory is defined via a set of waypoints that form a closed loop. In the reported experiments, a desired speed of $0.3\,\mathrm{m/s}$ is selected to generate a smooth and repeatable reference motion for the follower. The scenario includes typical road elements (ground, walls, pedestrian crossing), making it closer to real driving conditions.

Placeholder for the QLabs environment configuration and the selected waypoints ("Figures 5–6" in the provided report).

Figure 5.3: QLabs scenario and example of selected waypoints for the leader vehicle.

2.2 Measurements and perception pipeline

Designing trust-aware distributed estimation requires a clear definition of the measurements and metadata available on the QCar2 platform. In this work, the data sources include: onboard sensors, perception modules (LiDAR-camera fusion), and V2V exchanged states. Table 5.1 summarizes the measurements used in the reported experiments.

Table 5.1: Summary of measurements and metadata available on QCar2/QLabs used for control, estimation, and trust evaluation.

Category	**Measurement / Metadata**	**Source**	**Usage**
Onboard sensors (raw)	Position (x, y) and timestamp	QLabs virtual GPS (GPSSync API)	Time alignment and trajectory registration
Onboard sensors (raw)	Acceleration and angular rate	IMU (accelerometers, gyros)	Instantaneous vehicle dynamics
Onboard sensors (raw)	2D point cloud	LiDAR (RPLiDAR A2, $12\,\mathrm{m}$ range)	360° obstacle sensing
Onboard sensors (raw)	RGB / depth images	Cameras (Intel RealSense D435 and CSI 360°)	Visual perception
Onboard sensors (raw)	Wheel speed and steering angle	Motor encoder and steering servo	Inputs for longitudinal/lateral control
Derived kinematics	Position, orientation (ψ)	QLabs global state	Absolute vehicle state in the map
Derived kinematics	Longitudinal speed (v)	Encoder + Kalman filter	Smoothed speed estimate
Derived kinematics	Longitudinal acceleration (a)	Derivative of v (Kalman) or IMU	Longitudinal acceleration estimate
Cooperative variables	Inter-vehicle distance (d)	LiDAR-camera fusion pipeline	Reliable distance for cooperative control
Cooperative variables	Relative speed (Δv)	Temporal derivative of d	Rate of change of inter-vehicle gap
V2V metadata	Neighbor states (p, v, a)	Periodic V2V messages	Cooperation / distributed estimation
V2V metadata	Link quality	Packet loss rate, delays	Communication reliability for trust

Two critical quantities for cooperative control and trust assessment are the inter-vehicle distance d and the relative speed Δv. They are obtained via a LiDAR-camera fusion pipeline, which can be summarized as follows:

- Image segmentation: YOLOv8-seg is used to extract object masks (vehicles, pedestrians).
- Post-processing: mask erosion reduces false positives.
- LiDAR projection: 3D LiDAR points are projected into the image plane using intrinsic/extrinsic calibration matrices.
- Data association: projected points are associated to segmented objects, yielding a 2D-3D coupling.
- Spatial clustering: DBSCAN filters and groups fused points.
- 3D estimation: PCA computes 3D bounding boxes and relative positions.
- Final quantities: distance d is deduced from relative position; relative speed is obtained from time-derivative of d.

2.3 V2V communication and modular software architecture and Attack design

2.3.1 Design of V2V communication

Beyond control and estimation algorithms, the overall system requires a robust communication layer. The V2V stack is organized into three functional layers, as illustrated in Figure 5.4:

1. **Application layer**: high-level cooperative functions (platoon controller, distributed observer, trust manager) consume and produce vehicle state messages.
2. **Session layer**: handles message reliability through acknowledgments (ACK/NACK), periodic heartbeats for neighbor-presence detection, and a virtual-GPS-based time synchronization service that timestamps every packet.
3. **Transport layer**: UDP sockets transmit and receive datagrams at a fixed period (typically $T_s = 100\,\mathrm{ms}$). Separate threads handle transmission and reception to avoid blocking.

At each communication period, a vehicle broadcasts a state packet containing:

- vehicle identifier (id);
- position (x, y) and heading ψ;
- longitudinal speed v and acceleration a;
- synchronized timestamp t.

When a packet is received, an ACK is returned to the sender; if no ACK arrives within a timeout, the message is flagged as lost. In parallel, a heartbeat mechanism periodically checks for missing neighbors: if no message (nor heartbeat) is received from a neighbor for a configurable interval, that neighbor is

marked as unavailable and excluded from the cooperative estimation until communication resumes. These mechanisms ensure temporal consistency and data integrity, which are prerequisites for the distributed observer and trust framework.

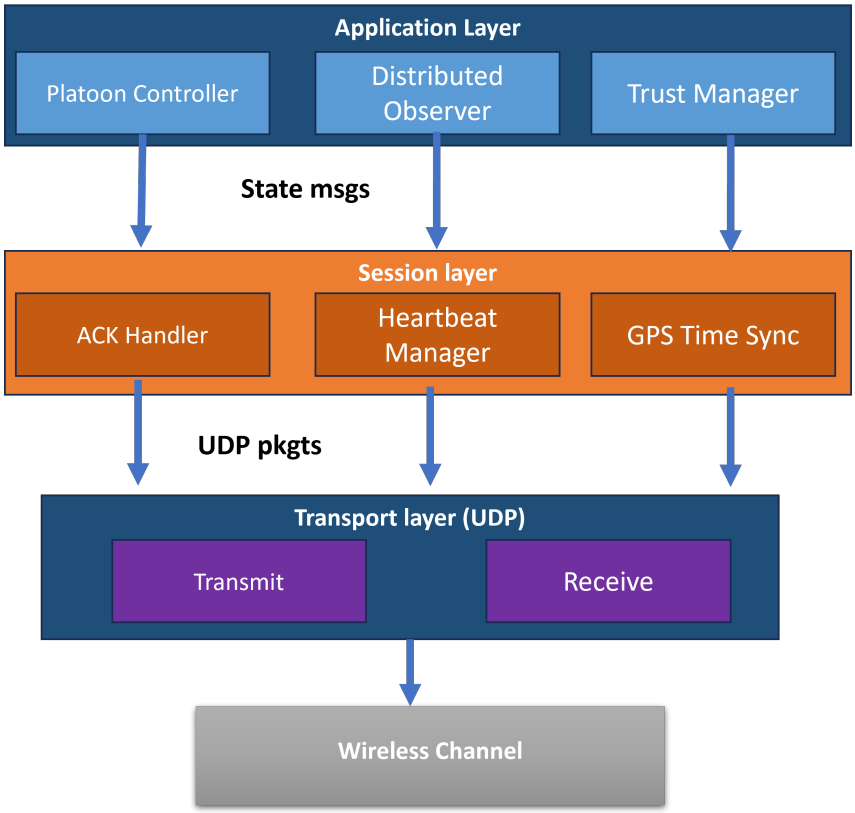


Figure 5.4: Layered architecture of the V2V communication stack used in the QLabs experiments.

2.3.2 Attack injection mechanism

To evaluate the robustness of the distributed observer and trust framework under adversarial conditions, a configurable *attack injector* is integrated into the communication layer. The injector intercepts outgoing or incoming V2V packets and applies one of several perturbation models before the data reach the application layer. Figure 5.5 illustrates the injection pipeline.

Injection modes. Three attack families are supported:

1. **Bias injection**: a constant offset δ is added to one or more state variables (e.g., $\tilde{v} = v + \delta_v$).
2. **Random fault**: with probability p, the transmitted value is replaced by a random sample drawn from a uniform distribution of configurable intensity.
3. **Data drop (DoS)**: with probability p_{drop}, the packet is discarded entirely, simulating a denial-of-service or severe packet-loss scenario.

Injection parameters. Each attack scenario is defined by:

- *attacker ID*: the vehicle whose outgoing messages are corrupted;
- *target variables*: which state components are affected (position, velocity, acceleration, or all);
- *attack window*: the time interval $[t_{start}, t_{end}]$ during which the attack is active;
- *attack parameters*: bias magnitude δ, fault intensity, or drop probability p_{drop}.

The injector can operate transparently (victim vehicles are unaware) or be logged for post-experiment analysis. This flexibility enables systematic benchmarking of the trust-aware estimation framework across a wide range of adversarial conditions (see Table 5.2 for the specific scenarios used in this work).

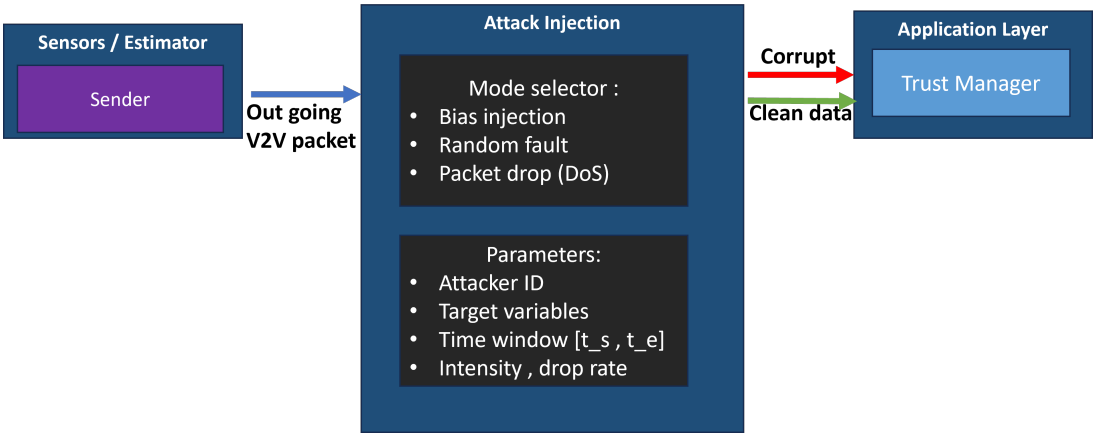


Figure 5.5: Attack injection mechanism integrated into the V2V communication layer.

3 Control architecture

3.1 Longitudinal control: cooperative cruise control

The longitudinal control loop is based on cooperative adaptive cruise control (CACC) to regulate both velocity and inter-vehicle spacing. The control input combines locally measured quantities (e.g., speed/acceleration) with cooperative information received via V2V (e.g., predecessor acceleration), and relies on the perception-derived distance d and relative speed Δv. This chapter focuses on the experimental integration and validation aspects; the detailed CACC modeling and resilient observer-based compensation are developed in earlier chapters.

3.2 Lateral control: extended look-ahead to reduce corner cutting

In curves, follower vehicles tend to exhibit *corner cutting* (reduced turning radius compared to the leader), which can induce lateral offsets and affect the effective inter-vehicle spacing. To mitigate this effect, an Extended Look-Ahead Controller (ELC) is implemented. The idea is to define a virtual pursuit point P^* located ahead of the preceding vehicle by a look-ahead distance d_{LA} and laterally shifted by d_{lat}. The follower then computes its steering command using this virtual target rather than the preceding vehicle's

center of gravity. This improves curve tracking and helps maintain the spacing policy (e.g., constant time-gap) in both straight and curved roads.

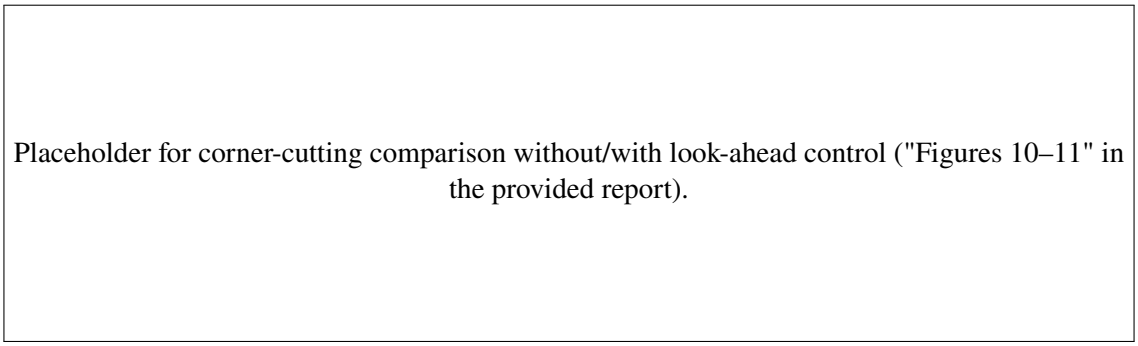


Figure 5.6: Effect of extended look-ahead on corner cutting in curves.

3.3 Attack scenarios and quantitative summary

In the reported virtual experiments, the leader vehicle (ID 1) behaves as the attacker between $t = 5\,\mathrm{s}$ and $t = 10\,\mathrm{s}$. Several scenarios are considered, including bias injection, random faults, and DoS-like packet drops on position/velocity/acceleration.

Table 5.2: Attack scenarios used to evaluate the trust-aware estimation framework (leader ID 1 attacks other vehicles during $t \in [5, 10]\,\mathrm{s}$).

Case	Attack method	Parameter(s)	Target (data)
1	Bias	$-5\,\mathrm{m}$	Position
2	Random fault	intensity $= 10$, $p = 0.3$	Position
3	Bias	$-2\,\mathrm{m/s}$	Velocity
4	Random fault	intensity $= 2.5$, $p = 0.3$	Velocity
5	Random fault	intensity $= 1.0$, $p = 0.3$	Acceleration
6	Data drop (DoS)	$p_{drop} = 0.5$	Position, velocity, acceleration

Two indicators are used to quantify the effects of the attacks: (i) a combined raw error capturing deviations on distance, speed, and acceleration estimates; and (ii) a normalized impact score obtained by dividing the combined error by the maximum error observed across scenarios. Among the considered scenarios, the DoS attack (Case 6) yields the highest normalized impact score (0.872), confirming it as the most severe perturbation. Conversely, the velocity bias attack (Case 3, bias $= -2\,\mathrm{m/s}$) yields the lowest reported impact score (0.240), indicating a limited effect on the overall platoon stability. In terms of vehicle sensitivity, the reported results show that vehicle $V4$ is the most affected (max impact 0.872), whereas vehicle $V2$ is the most resilient with the lowest average impact score (0.362). Finally, the trust score evolution under DoS shows a clear drop below the 0.5 threshold during the perturbation interval, followed by progressive recovery after the attack ends.

4 Custom embedded electronic board for future onboard deployment

4.1 Motivation and design objectives

To move from simulation-based validation / small-scale experiments to real trials, the project includes the design of a dedicated embedded electronic board. The objective is not only to "run the code on hardware", but to provide an integration-ready platform where sensing, communication, timing, and basic cyber-resilience can be validated under realistic constraints. In this thesis, the board is envisioned as an embedded *intelligent soft sensor*: it reconstructs non-measured variables, monitors the consistency of cooperative data, and exports reliable state and trust indicators to higher-level decision and control.

From a system engineering viewpoint, the design targets four key objectives:

- **Real-time execution:** guarantee deterministic execution of periodic estimation and control-related computations (e.g., $\approx 100\,\mathrm{Hz}$ loops as used in the experimental software stack), with bounded jitter.
- **Multi-interface sensing and synchronization:** acquire heterogeneous data streams (inertial, positioning/time reference, and auxiliary sensors) and ensure time alignment through timestamping.
- **V2V-ready communication:** support low-latency exchange of compact state messages and metadata (packet loss, delay indicators) required by the distributed observer and trust evaluation.
- **Robustness and diagnosability:** provide a hardware and firmware structure that supports fault detection, health monitoring, and graceful degradation when sensors or communications are unavailable.

Compared with a purely PC-based ground-station deployment, the embedded approach also addresses practical constraints that become critical on a vehicle: compactness and wiring reduction, controlled power distribution, electromagnetic compatibility (EMC), and reliable boot/execution behavior. The overall role of the board is therefore to acquire onboard sensing data, exchange information with neighboring vehicles, fuse local and remote information, and publish consolidated variables (including trust indicators) to the vehicle-level decision and control system.

4.2 Architecture and PCB design

The development process is structured into four phases: (i) functional needs analysis and block-architecture definition, (ii) electronic CAD design, (iii) prototyping and embedded programming, and (iv) experimental verification with rigorous test protocols. The design identifies the required sensors, the target microcontroller family, and the relevant communication protocols, including **SPI**, **I^{2}C**, **UART**, and **Wi-Fi 6**. The electronic design is implemented in KiCad and routed on a **4-layer PCB**, which offers two main benefits: (i) dedicated reference planes to control return paths and reduce EMI, and (ii) improved power distribution and decoupling compared with 2-layer prototypes.

Functional blocks. At a high level, the board architecture (Figure 5.7) can be described as a composition of:

- a **real-time processing unit** (microcontroller + memory resources) responsible for scheduling tasks, running the embedded observer/trust logic, and managing peripherals;
- a **communication subsystem** supporting short-range V2V exchange (Wi-Fi) and wired diagnostic/telemetry links (UART);
- a **sensing and timing subsystem** interfacing inertial and positioning/time-reference signals through SPI/I^{2}C/UART, and providing timestamping for data consistency;
- a **power subsystem** that converts the vehicle supply to regulated rails for digital logic and RF modules, with attention to inrush current, ripple, and protection.

Electrical and layout considerations. Beyond functional connectivity, the PCB is designed to be compatible with an onboard environment where noisy actuators, switching regulators, and RF emissions may coexist. The main layout guidelines are summarized below:

- **Power integrity:** short decoupling loops close to IC supply pins, separation of noisy switching nodes from sensitive analog/RF areas, and adequate bulk capacitance for transient loads.
- **Grounding and return paths:** continuous ground reference planes and stitching vias to reduce loop areas, ensuring that high-frequency return currents follow controlled paths.
- **High-speed / RF routing:** controlled-impedance routing and keep-out zones for the Wi-Fi/RF section, plus careful placement of the GNSS/RF components to minimize coupling.
- **Robust I/O:** ESD-aware connector placement and protection where needed (especially for external cables), as well as clear separation between internal buses (SPI/I^{2}C) and external-facing ports.

Particular care is therefore devoted to power integrity, impedance constraints, electromagnetic compatibility, and high-frequency signal management (Wi-Fi and GNSS).

Table 5.3: Embedded board design targets for thesis-to-vehicle transfer (high-level, implementation-dependent).

extbfDesign aspect	**Target / rationale**
Real-time behavior	Deterministic scheduling for periodic estimation tasks (e.g., $\sim 100\,\mathrm{Hz}$) and bounded latency for event-driven message handling.
Interfaces	Support SPI/I^{2}C/UART for sensors and debugging; support Wi-Fi for V2V exchange of compact state packets and link-quality metadata.
Time consistency	Timestamping and time-reference handling to align sensor samples and received packets before fusion and trust evaluation.
Robustness	Health monitoring (watchdog, peripheral status), logging/telemetry hooks, and safe fallback behavior when links or sensors degrade.
PCB constraints	4-layer stack-up to improve grounding and EMC; separation of RF/power/sensitive areas; routing practices compatible with on-board EMI constraints.

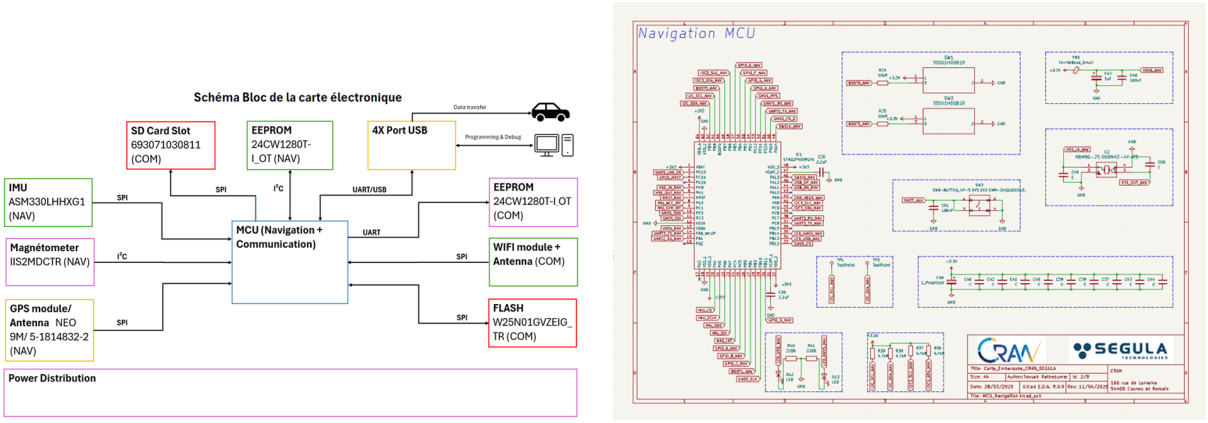
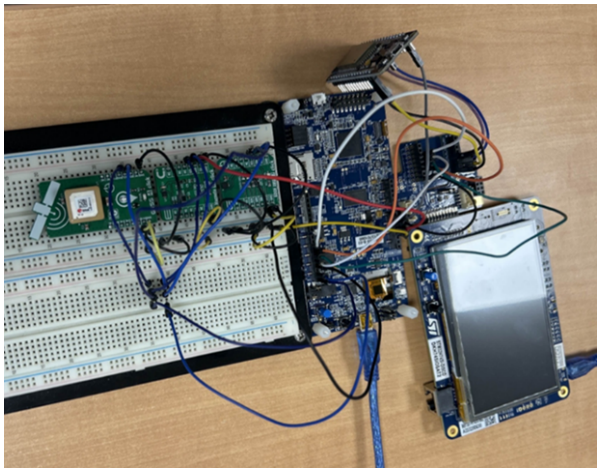
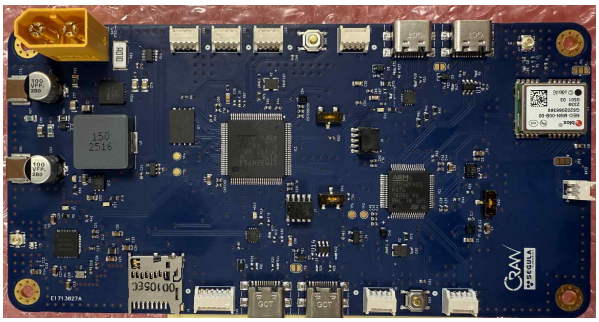


Figure 5.7: Embedded board design: block architecture and KiCad PCB implementation.



(a) Prototype validation using development kits (firmware bring-up and I/O tests).



(b) Manufactured embedded board intended for onboard deployment.

Figure 5.8: Prototype validation and manufactured embedded board.

4.3 Firmware structure and first validation tests

In order to validate the hardware early and de-risk the integration, initial firmware developments are performed using development modules before deployment on the final board. The embedded software is prepared using STM32CubeMX and STM32CubeIDE, and integrates a real-time operating system (RTOS). The firmware is designed as a modular set of drivers and tasks that mirrors the software decomposition used in the QLabs experiments (sensing, communication, estimation, and trust update), while respecting embedded constraints (bounded memory, fixed timing, and robust I/O).

RTOS organization. From a scheduling point of view, two categories of tasks are distinguished:

- **Periodic tasks** (time-triggered), responsible for sensor acquisition, filtering, and state propagation at a fixed rate.
- **Event-driven tasks** (message-triggered), responsible for reception/transmission handling and for

updating distributed estimates when new V2V data arrive.

This separation is aligned with the needs of the trust-aware distributed observer: local estimation must remain stable even when communication becomes intermittent, while V2V updates can be processed opportunistically and explicitly timestamped.

Data flow and software services. At a functional level, the firmware implements the following services:

- acquire environment data from onboard sensors;
- structure data in a dynamic list adapted to real-time processing;
- communicate with neighbor vehicles to exchange environmental and state data;
- fuse received information with local measurements for a coherent view;
- transmit consolidated variables to the vehicle system for decision making.

The sensing layer provides unified data structures that include both numerical values and metadata (timestamps, validity flags, and optional quality indicators). The communication layer provides packet framing, serialization/deserialization, and hooks for monitoring link quality (e.g., packet counters, estimated loss, and delay statistics). At the application level, the embedded "observer/soft-sensor" logic consumes these time-stamped inputs and outputs a consistent state estimate together with trust-related indicators that can be logged or transmitted to the vehicle control stack.

Verification philosophy. The first firmware iterations focus on hardware bring-up and repeatable tests that validate each subsystem independently before enabling closed-loop operation. This staged validation reduces integration risk: peripherals are tested with controlled stimuli, and each interface is instrumented with debug traces before moving to multi-task real-time execution. First validation tests reported in the provided document confirm:

- sensor data readout and transmission via UART;
- stable RTOS scheduler operation at $100\,\mathrm{Hz}$;
- correct interrupt and asynchronous task handling;
- compatibility between configured peripherals and the target hardware.

In practice, these checks provide evidence that the embedded platform can sustain the required sampling and communication rates, and that timing-critical sections (interrupt handlers, DMA transfers if used, and inter-task synchronization) behave reliably under continuous operation.

4.4 Integration roadmap with thesis algorithms

The embedded platform is designed to support a progressive transfer of the estimation and resilience mechanisms developed throughout the thesis:

- implementing the local observer as a periodic real-time task (sensor acquisition, filtering, state update);
- implementing the distributed observer and trust update as event-driven tasks synchronized with V2V message reception;
- maintaining time consistency via timestamping (synchronized clock) and heartbeat/ACK mechanisms;
- ensuring safe degradation modes under detected faults/attacks by down-weighting or isolating untrusted neighbors.

The proposed integration is intentionally incremental. First, the embedded platform is used as a data concentrator and logger to validate sensing and timestamping on the vehicle. Second, the local observer is activated onboard, producing filtered kinematic estimates and residuals for fault detection. Third, V2V exchange is enabled and the distributed observer is executed, initially in a monitoring mode (estimating and logging without affecting control), then progressively coupled to the decision/control layer once stability and robustness are confirmed.

Two practical aspects are key to a successful transfer:

- **Computational budgeting:** profiling of task execution times and memory usage to ensure that worst-case execution times remain compatible with the selected sampling periods.
- **Numerical robustness:** careful management of numeric scaling, saturation, and data validation (outlier rejection, missing-data handling) to avoid observer divergence when inputs are corrupted or delayed.

At the hardware level, the reported design identifies future optimization axes, including RF matching network tuning, via-loss reduction, improved high-frequency routing, and more refined power management. From an experimental perspective, these optimizations will be coupled with extended stress tests (continuous operation, communication congestion, and perturbation injection) to verify that the embedded implementation preserves the key properties demonstrated in simulation: stable local estimation, timely trust degradation under attacks, and recovery once the perturbation ends.

5 Conclusion

This chapter presented the experimental workflow and results obtained on the Quanser QCar2/QLabs platform. The full software stack (communication, perception, control, estimation) was integrated and validated in real time. Nominal experiments confirm accurate local state reconstruction and coherent distributed estimation of the whole fleet. Under representative attack scenarios (biases, random faults, and DoS drops), the trust-aware fusion mechanism detects inconsistencies and mitigates their impact, with the DoS case being the most severe according to the reported normalized impact score. Finally, a custom embedded electronic board was designed and prototyped as a hardware foundation for future real-vehicle deployment of the proposed observers and cyber-resilience mechanisms.

Conclusion and Perspectives

1 General Conclusion

This thesis addressed the problem of resilience in autonomous and connected vehicles through the development of advanced state estimation algorithms. As autonomous driving systems increasingly rely on interconnected sensors, embedded computation, and wireless communications, their vulnerability to sensor faults, modeling uncertainties, and cyber-attacks has become a major safety concern. In particular, false data injection attacks targeting vehicle-to-vehicle communication channels pose a serious threat to cooperative driving functionalities such as Adaptive Cruise Control (ACC), Cooperative Adaptive Cruise Control (CACC), and vehicle platooning.

The main objective of this research was to design estimation frameworks capable of maintaining reliable state information under adverse conditions, thereby ensuring safe and robust control performance. To this end, several complementary observer-based approaches were developed, analyzed, and validated within the context of autonomous and connected vehicle systems.

First, a discrete-time unknown input observer was proposed to estimate both vehicle states and malicious disturbances affecting communication signals. By employing an advanced discretization strategy and appropriate state transformations, the observer overcomes classical existence constraints and significantly reduces estimation delay. This contribution enables fast and accurate reconstruction of cyber-attack signals, allowing the controller to compensate for their effects in real time rather than merely detecting their presence.

Second, to address nonlinearities and modeling uncertainties inherent to vehicle dynamics, a neural network-based observer was introduced. This observer integrates learning capabilities into a model-based estimation framework, allowing online adaptation to unknown dynamics and external perturbations. The proposed architecture preserves stability while improving estimation accuracy in scenarios where classical models alone are insufficient, such as aggressive maneuvers or varying road conditions.

Third, the estimation framework was extended to cooperative and distributed scenarios involving vehicle platoons and mixed traffic environments. A distributed observer architecture combined with a trust management mechanism was developed to evaluate the reliability of information received through vehicle-to-vehicle communication. This approach limits the influence of compromised or unreliable data and enhances the collective resilience of the platoon, preventing the propagation of malicious effects from a single vehicle to the entire formation.

The proposed methods were extensively validated through high-fidelity simulations using MAT-

LAB/Simulink, CARLA, and Quanser QLabs environments. Various scenarios were considered, including nominal operation, sensor faults, communication disturbances, and cyber-attacks. The results demonstrate that the developed estimation algorithms significantly improve robustness, estimation accuracy, and safety compared to conventional approaches. Furthermore, the preparation for embedded implementation confirms the feasibility of deploying the proposed observers in real-time automotive systems.

Overall, this thesis contributes novel theoretical developments and practical solutions for resilient autonomous and connected vehicles. By combining model-based estimation, learning techniques, and cooperative trust mechanisms, it provides a unified framework capable of addressing both physical and cyber-related uncertainties in modern intelligent transportation systems.

2 Outlook and Perspectives

Although the results obtained in this thesis represent a significant step toward resilient autonomous driving, several promising research directions remain open and deserve further investigation.

A first perspective concerns large-scale experimental validation. While the proposed algorithms were tested in realistic simulation environments and prepared for embedded implementation, full-scale experiments on real vehicles in diverse traffic conditions would provide valuable insights into their real-world performance. Such experiments could include highway platooning, urban driving, and interaction with non-cooperative road users.

A complementary perspective is the **full onboard implementation** of the proposed resilient estimation and trust mechanisms on an embedded platform. In particular, the custom electronic board introduced in Chapter 6 was designed as a future hosting target for the observers developed in the thesis (unknown-input observers, neuro-adaptive observers, and trust-aware distributed observers). Future work will consist in porting the estimation stack to the embedded RTOS environment (task scheduling at $100\,\mathrm{Hz}$, timestamped V2V messaging, and robust I/O handling), and validating the resulting closed-loop performance on the QCar2 platform and, ultimately, on full-scale vehicles. This step will enable a rigorous assessment of real-time constraints (latency, packet losses, sensor noise), computational load, and energy consumption, while providing a practical path from simulation-grade validation to deployable automotive-grade solutions.

A second perspective involves extending the estimation framework to cooperative perception systems. Future autonomous vehicles will increasingly rely on shared perception data, such as camera images, LiDAR point clouds, and radar detections exchanged through V2V and V2I communications. Integrating resilient estimation algorithms with cooperative perception would enable robust fusion of heterogeneous data sources while mitigating the impact of corrupted or delayed information.

Another important direction concerns the interaction between estimation, control, and decision-making layers. In this thesis, the focus was placed on state estimation and its role in ensuring reliable control. Future work could explore tighter integration between resilient observers and higher-level planning or decision-making modules, allowing autonomous vehicles to adapt their behavior dynamically in response to detected threats or uncertainty levels.

From a methodological standpoint, further research could investigate advanced learning techniques,

such as adaptive neural ordinary differential equations or hybrid physics-informed learning models, to enhance estimation accuracy while maintaining stability guarantees. Additionally, incorporating uncertainty quantification into the estimation process could provide confidence measures that inform both control actions and trust management strategies.

Finally, extending the proposed framework to multi-agent traffic networks and infrastructure-assisted systems represents a long-term perspective. In such systems, vehicles, roadside units, and cloud services collaborate to optimize traffic flow and safety. Developing scalable, resilient estimation algorithms capable of operating across multiple layers of this ecosystem remains a challenging and impactful research direction.

In conclusion, the work presented in this thesis lays a solid foundation for resilient estimation in autonomous and connected vehicles. It opens the door to numerous future developments aimed at enhancing safety, reliability, and trust in next-generation intelligent transportation systems.

List of Publications

Peer-Reviewed Journal Articles

Q. H. Nguyen, O. Sadki, H. Rafaralahy, M. Haddad, A. Zemouche. "*Cyberattack Detection by Using a Discrete-Time Model-Based Unknown Input Observer*". IEEE Control Systems Letters 8, 856-861, 2024.

International Conference Papers

Q. H. Nguyen, S. Meng, M. Haddad, H. Rafaralahy, A. Zemouche. "*Neural Observer-Based Learning for Robust State Estimation*". The International Conference on Electrical and Computer Engineering (ICEECE), Madagasca, 2025.

S. Bougherara, **Q. H. Nguyen**, M. Haddad, H. Rafaralahy, A. Zemouche. "*Application of Discrete-Time Unknown Input Observers for Cyberattack Detection in Connected CACC Vehicles*". (ICEECE), Madagasca, 2025.

S. Meng, **Q. H. Nguyen**, A. Zemouche, F. Meng, F. Zhang. "*Distributed High-Gain Observer for Nonlinear Connected Autonomous Vehicle*". 2025 American Control Conference (ACC).

O. Sadki, **Q. H. Nguyen**, A. Zemouche, H. Rafaralahy, M. Haddad. "*Cyberattack Estimation in Intelligent Transportation System by Using an Adaptive High Order Sliding Mode Differentiator*". 2025 33rd Mediterranean Conference on Control and Automation (MED).

Q. H. Nguyen, O. Sadki, H. Rafaralahy, M. Haddad, A. Zemouche. "*Cyberattack Detection by Using a Discrete-Time Model-Based Unknown Input Observer*". CDC, 2024.

Article Submitted for Publication

Q. H. Nguyen, S. Meng, M. Haddad, H. Rafaralahy, A. Zemouche. "*Resilient Trust–Aware Distributed Observer Design for Connected Vehicle Platoons.*". 23rd IFAC World Congress, 2026 (Submitted).

Q. H. Nguyen, S. Meng, M. Haddad, H. Rafaralahy, A. Zemouche. "*Control-Oriented Trust-Aware Observer Design for Connected Vehicle Platoons.*". IEEE Transactions on Intelligent Transportation Systems (Preparation).

Poster Presentations

Q. H. Nguyen, O. Sadki, H. Rafaralahy, M. Haddad, A. Zemouche. "*Cyberattack Detection by Using a Discrete-Time Model-Based Unknown Input Observer*". (ACC), 2024.

Bibliography

[Benallouch, 2017] M. Benallouch, R. Outbib, M. Boutayeb, and E. Laroche. "Robust Observers for a Class of Nonlinear Systems Using PEM Fuel Cells as a Simulated Case Study". *IEEE Transactions on Control Systems Technology. DOI: 10.1109/TCST.2017.2658181* (2017) (cit. on p. 36).

[Bu, 2012] Fanping Bu and Ching-Yao Chan. "Adaptive and Cooperative Cruise Control". *Handbook of Intelligent Vehicles*. Ed. by Azim Eskandarian. London: Springer London, 2012, pp. 191–207 (cit. on p. 15).

[Cabelin, 2021] Joe Diether Cabelin, Paul Vincent Alpano, and Jhoanna Rhodette Pedrasa. "SVM-based Detection of False Data Injection in Intelligent Transportation System". *2021 International Conference on Information Networking (ICOIN)*. 2021, pp. 279–284 (cit. on p. 15).

[Chaouche, 2022] A. Chaouche, A. Zemouche, M. Ramdani, K. Chaib Draa, and D. Delattre. "Unknown input estimation algorithms for a class of LPV/nonlinear systems with application to wastewater treatment process". *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* 236.7 (2022), pp. 1372–1385 (cit. on pp. 18, 36).

[Charandabi, 2014] B.A. Charandabi and H.J. Marquez. "A novel approach to unknown input filter design for discrete-time linear systems". *Automatica* 50.2 (2014), pp. 2835–2839 (cit. on p. 36).

[Cheng, 2023] P. Cheng, J. Pan, and Y. Zhang. "Adaptive unknown input observer-based detection and identification method for intelligent transportation under malicious attack". *Measurement and Control* 56.7-8 (2023), pp. 1377–1386 (cit. on p. 15).

[Dabroom, 1997] A. Dabroom and H.K. Khalil. "Numerical differentiation using high-gain observers". *Proceedings of the 36th IEEE Conference on Decision and Control*. Vol. 5. 1997, 4790–4795 vol.5 (cit. on p. 19).

[Hassan, 2013] L. Hassan, A. Zemouche, and M. Boutayeb. "Robust Unknown Input Observers For Nonlinear Time-Delay Systems". *SIAM Journal on Control and Optimization* 51.4 (2013), pp. 2735–2752 (cit. on p. 36).

[Hou, 1992] M. Hou and P. C. Muller. "Design of observers for linear systems with unknown inputs". *IEEE Transactions on Automatic Control* 37.6 (1992), pp. 871–875 (cit. on p. 36).

[Huang, 2022] X. Huang and X. Wang. "Detection and isolation of false data injection attack in intelligent transportation system via robust state observer". *Processes* 10.7 (2022), p. 1299 (cit. on p. 15).

[Jeon, 2020] W. Jeon, Z. Xie, A. Zemouche, and R. Rajamani. "Simultaneous cyber-attack detection and radar sensor health monitoring in connected ACC vehicles". *IEEE Sensors Journal* 21.14 (2020), pp. 15741–15752 (cit. on pp. 15, 37).

[Khalil, 2020] A. Khalil, M. Al Janaideh, K.F. Aljanaideh, and D. Kundur. "Fault detection, localization, and mitigation of a network of connected autonomous vehicles using transmissibility identification". *2020 American Control Conference (ACC)*. IEEE. 2020, pp. 386–391 (cit. on p. 15).

[Mousavinejad, 2019] E. Mousavinejad, F. Yang, Q.L. Han, X. Ge, and L. Vlacic. "Distributed cyber attacks detection and recovery mechanism for vehicle platooning". *IEEE Transactions on Intelligent Transportation Systems* 21.9 (2019), pp. 3821–3834 (cit. on p. 15).

[Nguyen, 2024a] Q. H. Nguyen, O. Sadki, H. Rafaralahy, M. Haddad, and A. Zemouche. "Cyberattack Detection by Using a Discrete-Time Model-Based Unknown Input Observer". *IEEE Control Systems Letters* 8 (2024), pp. 856–861 (cit. on pp. 27–30).

[Nguyen, 2024b] Q.H. Nguyen, O. Sadki, H. Rafaralahy, M. Haddad, and A. Zemouche. "Cyberattack Detection by Using a Discrete-Time Model-Based Unknown Input Observer". *IEEE Control Systems Letters* 8 (2024), pp. 856–861 (cit. on p. 37).

[Pan, 2023a] Dengfeng Pan, Xiaohua Ge, Derui Ding, and Qing-Long Han. "Observer-Based Resilient Control of CACC Vehicle Platoon Against DoS Attack". *Automot. Innov. 6, 176-189 (2023)* (2023) (cit. on p. 15).

[Pan, 2023b] Dengfeng Pan, Xiaohua Ge, Derui Ding, and Qing-Long Han. "Simultaneous Cyber Attack Estimation and Radar Spoofing Attack Detection for Connected Automated Vehicles". *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society* (2023), pp. 1–6 (cit. on p. 15).

[Phanomchoeng, 2014] G. Phanomchoeng and R. Rajamani. "Real-Time Estimation of Rollover Index for Tripped Rollovers With a Novel Unknown Input Nonlinear Observer". *IEEE/ASME Transactions on Mechatronics* 19.2 (2014), pp. 743–754 (cit. on p. 36).

[Rajamani, 2002] R. Rajamani and C. Zhu. "Semi-autonomous adaptive cruise control systems". *IEEE Transactions on Vehicular Technology* 51.5 (2002), pp. 1186–1192 (cit. on pp. 16, 17, 21).

[Trinh, 2011] H. Trinh and T. Fernando. *Functional observers for dynamical systems*. Vol. 420. Springer Science & Business Media, 2011 (cit. on pp. 18, 36).

[Trinh, 2008] H. Trinh, T.D. Tran, and T. Fernando. "Disturbance decoupled observers for systems with unknown inputs". *IEEE Transactions on Automatic Control* 53.10 (2008), pp. 2397–2402 (cit. on p. 36).

[Yamamoto, 2021] Y. Yamamoto, N. Kuze, and T. Ushio. "Attack detection and defense system using an unknown input observer for cooperative adaptive cruise control systems". *IEEE Access* 9 (2021), pp. 148810–148820 (cit. on p. 15).

[Zemouche, 2009] Ali Zemouche and Mohamed Boutayeb. "Nonlinear-observer-based $\mathcal{H}_\infty$ synchronization and unknown input recovery". *IEEE Trans. Circuits Syst. I. Regul. Pap.* 56.8 (2009), pp. 1720–1731 (cit. on p. 36).

[Zhang, 2021] Mukai Zhang, Badriah Alenezi, Stefen Hui, and Stanislaw H. Żak. "Unknown Input Observers for Discretized Systems With Application to Networked Systems Corrupted by Sparse Malicious Packet Drops". *IEEE Control Systems Letters* 5.4 (2021), pp. 1261–1266 (cit. on pp. 26, 27).

[Zhu, 2012] Fanglai Zhu. "State estimation and unknown input reconstruction via both reduced-order and high-order sliding mode observers". *Journal of Process Control* 22.1 (2012), pp. 296–302 (cit. on p. 19).

RÉSUMÉ

La fiabilité et la sécurité des véhicules connectés et autonomes (VCA) dépendent de manière critique de la disponibilité en temps réel d'informations précises sur l'état dynamique du véhicule. Cependant, les contraintes économiques limitent souvent l'utilisation de capteurs haut de gamme. De plus, les VCA sont vulnérables aux défauts de capteurs et aux cyberattaques ciblant les communications inter-véhicules (V2V/V2I). Pour relever ces défis, cette thèse propose un cadre de « capteur logiciel » résilient basé sur la redondance analytique.

Cette recherche développe une méthodologie de modélisation hybride intégrant des équations physiques et des réseaux de neurones neuro-adaptatifs. Cette approche permet une représentation précise des paramètres complexes et variables dans le temps ainsi que des dynamiques non modélisées. S'appuyant sur ces modèles, des algorithmes d'estimation non linéaires avancés observateurs à entrées inconnues (UIO) et observateurs neuro-adaptatifs sont développés pour garantir la résilience du système.

L'efficacité des stratégies proposées est démontrée à travers trois axes principaux :

- **CACC Cyber-sécurisé :** Des architectures d'observateurs sont conçues pour détecter et atténuer une large gamme de cybermenaces, incluant l'injection de fausses données (FDI), le déni de service (DoS), ainsi que les retards et pertes de paquets. Ces observateurs permettent la reconstruction des signaux d'attaque, assurant un contrôle longitudinal robuste.
- **Peloton et Suivi de Véhicules :** Un cadre d'estimation distribué est proposé pour gérer des modèles dynamiques non linéaires. Couplé à un mécanisme de gestion de la confiance, il permet de filtrer les données non fiables et d'assurer la cohésion du peloton lors de manœuvres complexes.
- **Estimation des Forces et Dynamique du Véhicule :** L'application des observateurs hybrides neuronaux est étendue à l'estimation de forces inconnues et de dynamiques complexes, adressant des problèmes critiques tels que le risque de renversement, où la connaissance précise des forces externes est essentielle pour la sécurité.

MOTS CLÉS

Estimation distribuée, observateur neuronal, Cyberattaque, Gestion de la confiance

ABSTRACT

The reliability and safety of Connected and Autonomous Vehicles (CAVs) critically depend on the real-time availability of precise dynamic state information. However, economic constraints often limit the use of high-end sensors. Furthermore, CAVs are vulnerable to sensor faults and cyber-attacks targeting inter-vehicle communications (V2V/V2I). To address these challenges, this thesis proposes a resilient "soft sensor" framework based on analytical redundancy.

This research develops a hybrid modeling methodology integrating physical equations with neuro-adaptive neural networks. This approach enables the accurate representation of complex, time-varying parameters and unmodeled dynamics. Building on these models, advanced nonlinear estimation algorithms: Unknown Input Observers (UIO) and neuro-adaptive observers are developed to ensure system resilience.

The effectiveness of the proposed strategies is demonstrated through three main axes:

- **Cyber-Secure CACC:** Observer architectures are designed to detect and mitigate a wide range of cyber-threats, including False Data Injection (FDI), Denial of Service (DoS), as well as delays and packet losses. These observers enable the reconstruction of attack signals, ensuring robust longitudinal control.
- **Platooning and Vehicle Tracking:** A distributed estimation framework is proposed to handle nonlinear dynamic models. Coupled with a trust management mechanism, it allows filtering out unreliable data and ensuring platoon cohesion during complex maneuvers.
- **Vehicle Dynamics and Force Estimation:** The application of hybrid neural observers is extended to the estimation of unknown forces and complex dynamics, addressing critical issues such as rollover risk, where precise knowledge of external forces is essential for safety.

KEYWORDS

State Estimation, Neural Observer, Cyber-attack, Trust management,