

Préparée à Université de Lorraine

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

Soutenue par

**Quang Huy NGUYEN**

Le xx mois 202x

École doctorale n°xxx

**Titre de l'école doctorale**

Spécialité

**Automatique**

Préparée au

Centre de Recherche Nancy

Composition

Prénom Nom 1

Affiliation

Prenom Nom 2

Affiliation

Prenom Nom 3

Affiliation

Prenom Nom 4

Affiliation

Prenom Nom 5

Affiliation

Prenom Nom 6

DR Affiliation

du

jury

*Président du jury*

*Examinateur*

*Rapporteur*

*Rapporteur*

*Rapporteur*

*Rapporteur*

*Directeur de thèse*





# Chapitre 1

## Learning with Unknown Input Observers for Robust Nonlinear Estimation

### Objectifs —

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.

## 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 :

$$\begin{aligned}\dot{x} &= \varphi(x, u) + B\mu_x \\ y &= Cx,\end{aligned}\tag{1.1}$$

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  $C \in \mathbb{R}^{n_y \times n_x}$  and  $B \in \mathbb{R}^{n_x \times n_\mu}$  are constant and known matrices. The function  $\varphi : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}^{n_x}$  represents the known nonlinear component of the system.

**Remark 1.** 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  $\eta$ .

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

**Assumption 1.** The output vector  $y$  has dimension greater than or equal to the number of unknown terms  $\mu_x$ ; equivalently,  $n_\mu \leq n_y$ .

**Assumption 2.** The nonlinear function  $\varphi(\cdot)$  is  $\gamma_\varphi$ -Lipschitz continuous on  $\mathcal{X}$ , uniformly on  $u \in \mathcal{U}$ .

**Assumption 3.** The unknown nonlinearity  $\mu_x(\cdot)$  is Lebesgue measurable and  $\mathcal{L}_2$ -bounded.

**Remark 2.** Assumption 3 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,  $\mu_x$  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  $\mu_x$ . More specifically, we propose to estimate  $\mu_x$  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,  $\mu_x$ , if the system fails to satisfy the following main necessary condition :

$$\text{rank}(CB) = \text{rank}(B). \quad (1.2)$$

In addition, to be able to directly construct a UIO for system (1.1) to estimate simultaneously  $x$  and  $\mu_x$ , the constraint (1.2) is not sufficient. For this, we should be able to write the system under the form :

$$\begin{aligned} \mathbb{E}\dot{\zeta} &= \varphi_\zeta(\zeta, u) \\ y &= \mathcal{H}\zeta \end{aligned} \quad (1.3)$$

with

$$\text{rank} \left( \begin{bmatrix} \mathbb{E} \\ \mathcal{H} \end{bmatrix} \right) = n_x + n_\mu. \quad (1.4)$$

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  $y_{\text{new}}$ , which enables the simultaneous estimation of  $x$  and  $\mu_x$ . 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 (1.2) 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 (1.2). This approach introduces a new measurement vector  $y_{\text{new}}$ , where the original output  $y(t)$  is used to estimate the state  $x(t)$ , and the additional outputs  $y_{\text{new}}$  are exploited to estimate the unknown input,  $\mu_x$ . 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, 2024] ; 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  $\mu_x$ , 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,  $\mu_x$ .

The strategy consists in rewriting system (1.1) without modifying it. Then, the UIO we will propose will be based on reliable approximation of (1.1).

$$\begin{aligned} \dot{x} + E_{\delta_1} \dot{\mu}_x &= \varphi(x, u) + B\mu_x + E\omega_{\delta_1} \\ y &= Cx + D_{\delta_2}\mu_x + D\omega_{\delta_2} \end{aligned} \tag{1.5}$$

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  $\omega_{\delta_j}(t) = \delta_j \dot{\mu}_x(t)$ ,  $\omega_{\delta_2}(t) = \delta_2 \mu_x(t)$ , 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 (1.4), 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 4. A simple procedure is to (i) choose  $D$  so that  $\text{rank}(D) = n_\mu$  (typically by selecting  $n_\mu$  independent measured outputs), and then (ii) choose  $E$  (possibly also sparse/binary) to complete the rank condition (1.4) with minimal additional coupling.
- **Interpretation of  $\delta_1, \delta_2$**  : the scalars  $\delta_1$  and  $\delta_2$  quantify the trade-off between *rank recovery* (feasibility of Assumption 4) and *disturbance amplification* through the induced terms  $\omega_{\delta_1}$  and  $\omega_{\delta_2}$ .

The next section presents LMI conditions that guarantee accurate estimation of  $x$  and  $\mu_x$  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  $\mu_x$  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

$$\mathcal{H}^1(\Omega) := \left\{ \psi \in \mathcal{L}^2(\Omega) \mid \frac{d\psi}{dt} \in \mathcal{L}^2(\Omega) \right\}$$

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

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

First, system (1.5) can be rewritten under the compact form :

$$\begin{aligned} \mathbb{E}\dot{\zeta} &= \varphi_\zeta(\zeta, u) + \bar{E}\omega_\delta \\ y &= \mathcal{H}\zeta + \bar{D}\omega_\delta \end{aligned} \tag{1.6}$$

where

$$\mathbb{E} := \begin{bmatrix} \mathbb{I}_{n_x} & E_{\delta_1} \end{bmatrix}, \quad \mathcal{H} := \begin{bmatrix} C & D_{\delta_2} \end{bmatrix}, \tag{1.7a}$$

$$\bar{E} := \begin{bmatrix} E & 0_{n_x \times n_\mu} \end{bmatrix}, \quad \bar{D} := \begin{bmatrix} 0_{n_y \times n_\mu} & D \end{bmatrix}, \tag{1.7b}$$

$$\zeta := \begin{bmatrix} x \\ \mu_x \end{bmatrix}, \quad \omega_\delta := \begin{bmatrix} \omega_{\delta_1} \\ \omega_{\delta_2} \end{bmatrix}, \tag{1.7c}$$

$$\varphi_\zeta(\zeta, u) := \varphi(x, u) + B\mu_x. \tag{1.7d}$$

Now, let us introduce the following necessary assumption :

**Assumption 4.** *The matrices  $E$  and  $D$  are chosen such that  $\mathbb{E}$  and  $\mathcal{H}$  satisfy the rank condition (1.4).*

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

$$\|\tilde{\zeta}\|_{\mathcal{H}^1} \leq \lambda_\delta \|\mu_x\|_{\mathcal{H}^1} + \beta \|\xi_0\| \tag{1.8}$$

where  $\tilde{\zeta} := \zeta - \hat{\zeta}$ ,  $\lambda_\delta$  is the disturbance attenuation level, explicitly depending on  $\delta_1$  and  $\delta_2$ ,  $\beta > 0$  is a weighting real constant, and  $\xi_0$  is a vector depending on  $\tilde{\zeta}(0)$  and  $\mu_x(0)$ .

### 3.2 UIO structure and error dynamics

From Assumption 4, there exist two matrices  $P_\zeta$  and  $Q_\zeta$  such that

$$P_\zeta \mathbb{E} + Q_\zeta \mathcal{H} = \mathbb{I}_{n_\zeta}. \quad (1.9)$$

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

$$\begin{cases} \dot{\eta} &= P_\zeta \varphi_\zeta(\hat{\zeta}, u) + K(y - \mathcal{H}\hat{\zeta}) \\ \hat{\zeta} &= \eta + Q_\zeta y \end{cases} \quad (1.10)$$

where  $\hat{\zeta}$  is the estimate of  $\zeta$  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 (1.8) with appropriate  $\lambda_\delta, \beta$ , and  $\xi_0$ .

We have

$$\begin{aligned} \tilde{\zeta} &= \zeta - \eta - Q_\zeta y \\ &= (\mathbb{I}_{n_\zeta} - Q_\zeta \mathcal{H}) \zeta - \eta - Q_\zeta \bar{D} \omega_\delta \\ &\stackrel{(1.9)}{\equiv} P_\zeta \mathbb{E} \zeta - \eta - Q_\zeta \bar{D} \omega_\delta \end{aligned} \quad (1.11)$$

which gives equivalently

$$\overbrace{\tilde{\zeta} + Q_\zeta \bar{D} \omega_\delta}^{\xi(t)} = P_\zeta \mathbb{E} \zeta(t) - \eta(t). \quad (1.12)$$

It is worth noting that the reformulation (1.12) conveniently avoids the derivative of  $\omega_\delta$  in the derivation of the error dynamics.

By using the dynamics (1.6) and (1.10), we obtain

$$\begin{aligned} \dot{\xi} &= P_\zeta [\varphi_\zeta(\zeta, u) - \varphi_\zeta(\hat{\zeta}, u)] - K \mathcal{H} \tilde{\zeta} \\ &\quad + (P_\zeta \bar{E} - K \bar{D}) \omega_\delta \\ &= (P_\zeta \nabla_\zeta^{\varphi_\zeta}(\mathbf{z}, u) - K \mathcal{H}) \xi \\ &\quad + (\mathcal{E}(\mathbf{z}, u) - K \mathcal{D}) \omega_\delta \end{aligned} \quad (1.13)$$

where

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

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

and from the differential mean value theorem [**Hasni\_LCSS\_24**], we have

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

for a given  $\mathbf{z}$ .

From this point onward, the analysis will be carried out with the vector  $\xi$  instead of  $\tilde{\zeta}$ , and we will return to  $\tilde{\zeta}$  at the end to derive the final  $\mathcal{H}^1$  bound (1.8).

### 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  $\vartheta(\cdot)$  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 :*

$$\dot{\vartheta}(z) + \|z\|^2 + \dot{z}^\top \mathcal{S} \dot{z} - \lambda \|w\|^2 \leq 0, \forall z \in \mathbb{R}^{n_\zeta}, w \in \mathbb{R}^{2n_\mu} \quad (1.14)$$

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

$$\|z\|_{\mathcal{H}^1} \leq \sqrt{\frac{\lambda}{\min(1, \lambda_{\min}(\mathcal{S}))}} \|w\|_{\mathcal{L}^2} + \sqrt{\frac{\vartheta_{\max}}{\min(1, \lambda_{\min}(\mathcal{S}))}} \|z_0\|. \quad (1.15)$$

Moreover, if (1.14) is satisfied with  $z := \xi$  and  $w := \omega_\delta$ , where  $\xi$  and  $\omega_\delta$  are defined by (1.12) and (1.7c), respectively, then the estimation error,  $\tilde{\zeta}$ , satisfies the  $\mathcal{H}^1$  bound (1.8) with  $\lambda_\delta$ ,  $\beta$ , and  $\xi_0$  given as follows :

$$\lambda_\delta := \delta_2 \sigma_{\max}(Q_\zeta D) + \max(\delta_1, \delta_2) \sqrt{\frac{\lambda}{\min(1, \lambda_{\min}(\mathcal{S}))}} \quad (1.16a)$$

$$\beta := \sqrt{\frac{\vartheta_{\max}}{\min(1, \lambda_{\min}(\mathcal{S}))}} \quad (1.16b)$$

$$\xi_0 := \tilde{\zeta}(0) + Q_\zeta D \mu_x(0). \quad (1.16c)$$

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

*Proof.* The first part of the proof is relatively straightforward. Before integrating (1.14) to get the norms  $\|\cdot\|_{\mathcal{H}^1}$  and  $\|\cdot\|_{\mathcal{L}^2}$ , take in mind that

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

Hence, since

$$\int_0^{+\infty} (\|z(s)\|^2 + \|\dot{z}(s)\|^2) ds = \|z\|_{\mathcal{H}^1}^2,$$

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

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

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

we can readily conclude (1.15).

As for the second part of the proof, we will exploit the structure of  $\xi$  and  $\omega_\delta$ . First,  $\xi$  and  $\omega_\delta$  satisfy (1.15) proved in the first part. In addition, we have

$$\begin{aligned}\|\tilde{\zeta}\|_{\mathcal{H}^1} &= \left\| \xi - Q_\zeta \bar{D} \omega_\delta \right\|_{\mathcal{H}^1} \\ &\leq \|\xi\|_{\mathcal{H}^1} + \left\| Q_\zeta \bar{D} \omega_\delta \right\|_{\mathcal{H}^1} \\ &= \|\xi\|_{\mathcal{H}^1} + \left\| \begin{bmatrix} 0 & Q_\zeta D \end{bmatrix} \omega_\delta \right\|_{\mathcal{H}^1} \\ &= \|\xi\|_{\mathcal{H}^1} + \delta_2 \|Q_\zeta D \mu_x\|_{\mathcal{H}^1} \\ &\leq \|\xi\|_{\mathcal{H}^1} + \delta_2 \sigma_{\max}(Q_\zeta D) \|\mu_x\|_{\mathcal{H}^1}\end{aligned}\tag{1.17}$$

and

$$\begin{aligned}\|\omega_\delta\|_{\mathcal{L}^2} &= \sqrt{\delta_1^2 \|\dot{\mu}_x\|_{\mathcal{L}^2}^2 + \delta_2^2 \|\mu_x\|_{\mathcal{L}^2}^2} \\ &\leq \max(\delta_1, \delta_2) \sqrt{\|\dot{\mu}_x\|_{\mathcal{L}^2}^2 + \|\mu_x\|_{\mathcal{L}^2}^2} \\ &= \max(\delta_1, \delta_2) \|\mu_x\|_{\mathcal{H}^1}.\end{aligned}\tag{1.18}$$

Hence, by substituting (1.17) and (1.18) in (1.15), the bound (1.8) is inferred with the parameters given in (1.16a)–(1.16c).  $\square$

Before stating the main theorem, notice that from Assumption 2, there exist constant matrices  $\mathcal{A}_j \in \mathbb{R}^{n_\zeta \times n_\zeta}$  and functions  $\alpha_j(\mathbf{z})$ ,  $j = 1, \dots, \bar{n}_\zeta$  such that the Jacobian  $\nabla_\zeta^{\varphi_\zeta}(\mathbf{z}, u)$  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.

$$\mathcal{H}_\varphi \triangleq \left\{ \sum_{j=1}^{\bar{n}_\zeta} \alpha_j(\mathbf{z}) \mathcal{A}_j, \sum_{j=1}^{\bar{n}_\zeta} \alpha_j(\mathbf{z}) = 1, \alpha_j(\mathbf{z}) \geq 0 \right\}\tag{1.19}$$

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 (1.8).

**Theorem 1.** Suppose that Assumptions 1, 2, and 3 hold. Let the matrices  $E$  and  $D$ , together with the scalars  $\delta_1$  and  $\delta_2$ , be chosen such that the Assumption 4 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 (1.26) are fulfilled. Then, the estimation error  $\tilde{\zeta}$ , with  $K = \mathcal{P}^{-1} \mathcal{Q}^\top$ , satisfies the  $\mathcal{H}^1$ -optimality bound (1.8) with the parameters given in (1.16a)–(1.16c), 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 (1.14). 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 (1.19), yielding vertex-wise LMIs.

*Proof.* Consider the quadratic Lyapunov function  $\vartheta(\xi) := \xi^\top \mathcal{P} \xi$ . Then, the derivative of  $\vartheta(\cdot)$  along the trajectories of (1.13) is given as follows :

$$\begin{aligned}\dot{\vartheta}(\xi) = \xi^\top & \left[ \left( P_\zeta \nabla_{\zeta}^{\varphi_\zeta}(z, u) - K\mathcal{H} \right)^\top \mathcal{P} + \mathcal{P} \left( P_\zeta \nabla_{\zeta}^{\varphi_\zeta}(z, u) - K\mathcal{H} \right) \right] \xi \\ & + \xi^\top \mathcal{P} (\mathcal{E}(z, u) - K\mathcal{D}) \omega_\delta + \omega_\delta^\top (\mathcal{E}(z, u) - K\mathcal{D})^\top \mathcal{P} \xi.\end{aligned}\quad (1.20)$$

and

$$\begin{aligned}\dot{\xi}^\top \mathcal{S} \dot{\xi} = \xi^\top & \left( P_\zeta \nabla_{\zeta}^{\varphi_\zeta}(z, u) - K\mathcal{H} \right)^\top \mathcal{S} \times \left( P_\zeta \nabla_{\zeta}^{\varphi_\zeta}(z, u) - K\mathcal{H} \right) \xi \\ & + \omega_\delta^\top (\mathcal{E}(z, u) - K\mathcal{D})^\top \mathcal{S} (\mathcal{E}(z, u) - K\mathcal{D}) \omega_\delta \\ & + 2\xi^\top \left( P_\zeta \nabla_{\zeta}^{\varphi_\zeta}(z, u) - K\mathcal{H} \right)^\top \mathcal{S} (\mathcal{E}(z, u) - K\mathcal{D}) \omega_\delta.\end{aligned}\quad (1.21)$$

Then, we have

$$\dot{\vartheta}(\xi) + \|\xi\|^2 + \dot{\xi}^\top \mathcal{S} \dot{\xi} - \lambda \|\omega_\delta\|^2 = \begin{bmatrix} \xi \\ \omega_\delta \end{bmatrix}^\top \mathbb{M}(z, u) \begin{bmatrix} \xi \\ \omega_\delta \end{bmatrix}\quad (1.22)$$

where  $\mathbb{M}(z, u)$  is defined in (1.24). By applying Schur lemma, we deduce that  $\mathbb{M}(z, u) < 0$  whenever inequality (1.25) 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 (1.26) hold. This, in turn, yields

$$\dot{\vartheta}(\xi) + \|\xi\|^2 + \dot{\xi}^\top \mathcal{S} \dot{\xi} - \lambda \|\omega_\delta\|^2 \leq 0.\quad (1.23)$$

Hence, Proposition 1 applies and the desired result follows. □

**Remark 3.** 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  $\epsilon$  is included as a feasibility factor to improve the feasibility of the LMI conditions (1.26), and it should be chosen a priori.

$$\begin{bmatrix} \left( P_\zeta \nabla_\zeta^{\varphi\zeta}(\mathbf{z}, u) - K\mathcal{H} \right)^\top \mathcal{P} + \mathcal{P} \left( P_\zeta \nabla_\zeta^{\varphi\zeta}(\mathbf{z}, u) - K\mathcal{H} \right) & \left[ \mathcal{P} + \left( P_\zeta \nabla_\zeta^{\varphi\zeta}(\mathbf{z}, u) - K\mathcal{H} \right)^\top \mathcal{S} \right] \left( \mathcal{E}(\mathbf{z}, u) - K\mathcal{D} \right) \\ \left( \mathcal{E}(\mathbf{z}, u) - K\mathcal{D} \right)^\top \left[ \mathcal{P} + \mathcal{S} \left( P_\zeta \nabla_\zeta^{\varphi\zeta}(\mathbf{z}, u) - K\mathcal{H} \right) \right] & -\lambda \mathbb{I}_{2n_\mu} + \left( \mathcal{E}(\mathbf{z}, u) - K\mathcal{D} \right)^\top \mathcal{S} \left( \mathcal{E}(\mathbf{z}, u) - K\mathcal{D} \right) \end{bmatrix} \quad (1.24)$$

$$\begin{bmatrix} \left( P_\zeta \nabla_\zeta^{\varphi\zeta}(\mathbf{z}, u) - K\mathcal{H} \right)^\top \mathcal{P} + \mathcal{P} \left( P_\zeta \nabla_\zeta^{\varphi\zeta}(\mathbf{z}, u) - K\mathcal{H} \right) & \mathcal{P} \left( \mathcal{E}(\mathbf{z}, u) - K\mathcal{D} \right) & \left( P_\zeta \nabla_\zeta^{\varphi\zeta}(\mathbf{z}, u) - K\mathcal{H} \right)^\top \mathcal{S} \\ \left( \mathcal{E}(\mathbf{z}, u) - K\mathcal{D} \right)^\top \mathcal{P} & -\lambda \mathbb{I}_{2n_\mu} & \left( \mathcal{E}(\mathbf{z}, u) - K\mathcal{D} \right)^\top \mathcal{S} \\ \mathcal{S} \left( P_\zeta \nabla_\zeta^{\varphi\zeta}(\mathbf{z}, u) - K\mathcal{H} \right) & \mathcal{S} \left( \mathcal{E}(\mathbf{z}, u) - K\mathcal{D} \right) & -\mathcal{S} \end{bmatrix} < 0 \quad (1.25)$$

$$\begin{bmatrix} \left( P_\zeta \mathcal{A}_j \right)^\top \mathcal{P} + \mathcal{P} \left( P_\zeta \mathcal{A}_j \right) - \mathcal{H}^\top \mathcal{Q} - \mathcal{Q}^\top \mathcal{H} & \mathcal{P} \mathcal{E}_j - \mathcal{Q}^\top \mathcal{D} & \left( P_\zeta \mathcal{A}_j \right)^\top \mathcal{P} - \mathcal{H}^\top \mathcal{Q} \\ \mathcal{E}_j^\top \mathcal{P} - \mathcal{D}^\top \mathcal{Q} & -\lambda \mathbb{I}_{2n_\mu} & \mathcal{E}_j^\top \mathcal{P} - \mathcal{D}^\top \mathcal{Q} \\ \mathcal{P} \left( P_\zeta \mathcal{A}_j \right) - \mathcal{Q}^\top \mathcal{H} & \mathcal{P} \mathcal{E}_j - \mathcal{Q}^\top \mathcal{D} & -\frac{1}{\epsilon} \mathcal{P} \end{bmatrix} < 0 \quad (1.26)$$

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 (1.2) 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 :

$$\begin{cases} \dot{\chi}_1 = \chi_2 \\ \dot{\chi}_2 = C\bar{\varphi}_x(x, u) + C\frac{\partial\varphi}{\partial x}(x, u)B\mu_x + CB\dot{\mu}_x \\ \dot{x} = \varphi(x, u) + B\mu_x \\ y_\chi = \chi_1 + C_\chi\chi_2 + Cx \end{cases} \quad (1.27)$$

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 (1.2) is satisfied, then (1.27) can be rewritten in the form of (1.6), for which Assumption 4 holds. In this case, an exponential UIO for (1.27) can be directly constructed if we add  $\chi_2$  as output measurement. However, if (1.2) 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 (1.27). 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 (1.2) is satisfied and the derivative  $\dot{y}$  is used as a measurement.

### 4.2 Regularization of the generalized model

System (1.27) can be regularized as follows

$$\begin{cases} \dot{\chi}_1 = \chi_2 \\ \dot{\chi}_2 + (E_{\delta_1} - CB)\dot{\mu}_x = C\bar{\varphi}_x(x, u) + E\omega_{\delta_1} + C\frac{\partial\varphi}{\partial x}(x, u)B\mu_x \\ \dot{x} = \varphi(x, u) + B\mu_x \\ y_\chi = \chi_1 + C_\chi\chi_2 + Cx + D_{\delta_2}\mu_x + D\omega_{\delta_2} \end{cases} \quad (1.28)$$

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  $\delta_1$  and  $\delta_2$ . 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  $\text{rank}(D) = n_\mu$  (select  $n_\mu$  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 (1.5) can be rewritten under the compact form :

$$\begin{aligned}\mathbb{E}\dot{\zeta} &= \varphi_\zeta(\zeta, u) + \bar{E}\omega_\delta \\ y_\chi &= \mathcal{H}\zeta + \bar{D}\omega_\delta\end{aligned}\tag{1.29}$$

where

$$\mathbb{E} := \begin{bmatrix} \mathbb{I}_{n_y} & 0 & 0 & 0 \\ 0 & \mathbb{I}_{n_y} & 0 & -CB \\ 0 & 0 & \mathbb{I}_{n_x} & 0 \end{bmatrix}, \quad \mathcal{H} := \begin{bmatrix} \mathbb{I}_{n_y} & C_\chi & C & D_{\delta_2} \end{bmatrix}, \tag{1.30a}$$

$$\bar{E} := \begin{bmatrix} 0 & 0_{n_y \times n_\mu} \\ E & 0_{n_y \times n_\mu} \\ 0 & 0_{n_x \times n_\mu} \end{bmatrix}, \quad \bar{D} := \begin{bmatrix} 0_{n_y \times n_\mu} & D \end{bmatrix}, \tag{1.30b}$$

$$\zeta := \begin{bmatrix} \chi_1 \\ \chi_2 \\ x \\ \mu_x \end{bmatrix}, \quad \omega_\delta := \begin{bmatrix} \omega_{\delta_1} \\ \omega_{\delta_2} \end{bmatrix}, \tag{1.30c}$$

$$\varphi_\zeta(\zeta, u) := \begin{bmatrix} \chi_2 \\ C\bar{\varphi}_x(x, u) + C\frac{\partial\varphi}{\partial x}(x, u)B\mu_x \\ \varphi(x, u) + B\mu_x. \end{bmatrix} \tag{1.30d}$$

with  $E_{\delta_1}, D_{\delta_2}, \omega_{\delta_1}$ , and  $\omega_{\delta_2}$  defined as in (1.5).

### 4.3 Generalized UIO

Here we provide the generalized UIO corresponding to the generalized and regularized system (1.29).

**Assumption 5.** *The matrices  $E$ ,  $D$ , and  $C_\chi$ , and the scalars  $\delta_1$  and  $\delta_2$  are chosen such that  $\mathbb{E}$  and  $\mathcal{H}$  satisfy the rank condition (1.4).*

**Assumption 6.** *The function  $\varphi_\zeta(., u)$  defined in (1.30d) is globally Lipschitz, uniformly on  $u$ .*

As in the previous section, from Assumption 5, there exist two matrices  $P_\zeta$  and  $Q_\zeta$  such that

$$P_\zeta \mathbb{E} + Q_\zeta \mathcal{H} = \mathbb{I}_{n_\zeta}. \tag{1.31}$$

where  $n_\zeta := 2n_y + n_x + n_\mu$ , and then an observer is constructed as in (1.10) as follows :

$$\begin{cases} \dot{\eta} &= P_\zeta \varphi_\zeta(\hat{\zeta}, u) + K(y_\chi - \mathcal{H}\hat{\zeta}) \\ \hat{\zeta} &= \eta + Q_\zeta y_\chi \end{cases} \quad (1.32)$$

To avoid repetitions, under Assumption 6, assume that the Jacobian  $\nabla_\zeta^{\varphi_\zeta}(\mathbf{z}, u)$  belongs to the convex polytopic set  $\mathcal{H}_\varphi$  defined in (1.19).

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 1 apply directly under Assumptions 5 and 6. Specifically, the observer (1.32) with  $K = \mathcal{P}^{-1}\mathcal{Q}^\top$  satisfies the  $\mathcal{H}^1$ -optimality bound (1.8), with parameters given in (1.16a)–(1.16c), 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  $\epsilon$  are defined in Theorem 1.

## 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 $\text{rank}(CB) = \text{rank}(B)$

We show that in this case, exponential convergence of the estimation error to zero can be guaranteed. Indeed, if  $\text{rank}(CB) = \text{rank}(B)$ , then to satisfy Assumption 5, it suffices to choose an appropriate matrix  $C_\chi$ , independently of  $\delta_1$  and  $\delta_2$ . We can then set  $\delta_1 = \delta_2 = 0$ , which implies  $\lambda_\delta = 0$  in (1.16a) and  $\omega_\delta(t) \equiv 0$  in (1.23). Under these conditions, the bound (1.8) reduces to

$$\|\tilde{\zeta}\|_{\mathcal{H}^1} \leq \beta \|\xi_0\|, \quad (1.33)$$

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

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

In other words, we have  $\xi(t) = \tilde{\zeta}(t)$ , and the LMIs (1.26) ensure (1.23), 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  $\text{rank}(CB) = \text{rank}(B)$ , the only way to satisfy Assumption 5 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 $\text{rank}(CB) < \text{rank}(B)$

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  $\delta_1$  and  $\delta_2$  such that Assumption 5 is satisfied, the  $\mathcal{H}^1$  bound (1.8) can be effectively adjusted by tuning the parameters  $\delta_1$  and  $\delta_2$ .

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

The generalized UIO (1.32) 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  $\chi_2$ . In particular, when  $\text{rank}(CB) = \text{rank}(B)$ , the estimate  $\hat{\chi}_2$  converges exponentially to  $\chi_2$ , 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 1.1 – Summary of specific cases : assumptions, design choices, and convergence properties.

Case / setting	Design choice to satisfy Assumption 5	Main guarantee (what you get)	Use of $\dot{y}$
$\text{rank}(CB) = \text{rank}(B)$ (classical matching holds)	Pick $C_\chi \neq 0$ ; can set $\delta_1 = \delta_2 = 0$ (no regularization, $\omega_\delta \equiv 0$ )	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
$\text{rank}(CB) < \text{rank}(B)$ (classical UIO fails)	Regularize : choose $E$ , $D$ , and $\delta_1, \delta_2 > 0$ so that the augmented pair $(\mathbb{E}, \mathcal{H})$ satisfies (1.4)	$\mathcal{H}^1$ bound with tunable attenuation level $\lambda_\delta$ (via $\delta_1, \delta_2$ ); exponential convergence not guaranteed	Typically used if $C_\chi \neq 0$ ; otherwise not needed

## 5 UIO-Based Neural Online Approximation of Unmodeled Nonlinearity $\mu_x$

As discussed in the previous section, the UIO provides an estimation of the unmodeled nonlinearity  $\mu_x$ . 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  $\mu_x$

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

$$u := \kappa(x_{ref}, u_{ref}, \hat{x}) \quad (1.34)$$

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 (1.1).

## 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  $\hat{x}_{UIO}$ , and the unknown input estimate,  $\hat{\mu}_x$  :

$$\hat{\zeta} = \begin{bmatrix} \hat{\chi}_1 \\ \hat{\chi}_2 \\ \hat{x}_{UIO} \\ \hat{\mu}_x \end{bmatrix} \in \mathbb{R}^{2n_y + n_x + n_\mu}. \quad (1.35)$$

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  $\mu_\theta(\hat{x}_{nn})$  denote the neural network approximation of  $\hat{\mu}_x$ , parameterized by  $\theta$  :

$$\mu_\theta(\hat{x}_{nn}) = \mathcal{N}_\theta(\hat{x}_{nn}, u) \quad (1.36)$$

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

$$\begin{cases} \dot{\hat{x}}_{nn} = \varphi(\hat{x}_{nn}, u) + B\mu_\theta(\hat{x}_{nn}) + L_{nn}(y - y_{nn}) \\ y_{nn} = C\hat{x}_{nn} \end{cases} \quad (1.37)$$

where  $\hat{x}_{nn} \in \mathbb{R}^{n_x}$  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 1.1. The physics-based UIO uses the

measured output  $y$  to provide robust initial estimates  $\hat{x}_{UIO}$  and  $\hat{\mu}_x$ . Simultaneously, the Neural Observer utilizes the learned model  $\mu_\theta$  to produce a refined state estimate  $\hat{x}_{nn}$ . The neural network parameters  $\theta$  are updated online by minimizing a composite loss function that enforces consistency between the data-driven model and the physics-based UIO estimates.

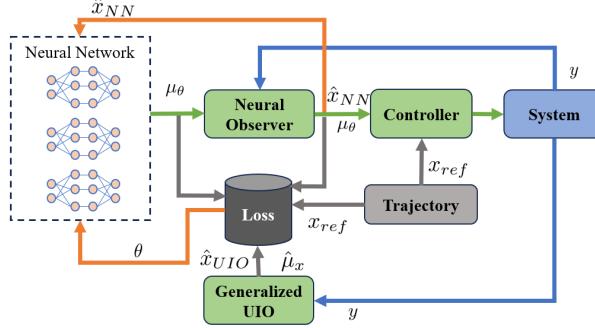


FIGURE 1.1 – Diagram of the proposed hybrid estimation strategy.

The dynamics of the original system (1.1) can be rewritten as :

$$\dot{x} = \varphi(x, u) + B\mu_\theta(\hat{x}_{nn}) + B\Delta_x(t) \quad (1.38)$$

where  $\Delta_x(t) \triangleq \mu_x(t) - \mu_\theta(\hat{x}_{nn}(t))$  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  $\epsilon_{nn} \triangleq x - \hat{x}_{nn}$  satisfies an  $\mathcal{L}^2$  performance criterion :

$$\|\epsilon_{nn}\|_{\mathcal{L}^2} \leq \lambda_{nn} \|\Delta_x\|_{\mathcal{L}^2} + \beta_{nn} \|\epsilon_{nn}(0)\| \quad (1.39)$$

where  $\lambda_{nn}$  and  $\beta_{nn}$  are positive scalars. We consider the quadratic Lyapunov function  $V(\epsilon_{nn}) = \epsilon_{nn}^\top \mathcal{P} \epsilon_{nn}$  with  $\mathcal{P} > 0$ . To satisfy (1.39), it suffices to ensure :

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

The error dynamics are given by :

$$\dot{\epsilon}_{nn} = \left( \nabla_x^\varphi(z_x, u) - L_{nn}C \right) \epsilon_{nn} + B\Delta_x(t) \quad (1.40)$$

where  $\nabla_x^\varphi(z_x, u)$  is the Jacobian of  $\varphi$  evaluated at some  $z_x$ . Under Assumption 2, 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 2.** *Under Assumption 2, if there exist a symmetric positive definite matrix  $\mathcal{P} \in \mathbb{R}^{nx \times nx}$ , a matrix  $\mathcal{Q} \in \mathbb{R}^{ny \times nx}$ , and a positive scalar  $\sigma$  satisfying the following LMI conditions*

for all vertices  $j = 1, \dots, \bar{n}_x$  :

$$\begin{bmatrix} (\mathcal{A}_j^x)^\top \mathcal{P} + \mathcal{P} \mathcal{A}_j^x - C^\top \mathcal{Q} - \mathcal{Q}^\top C & \mathcal{P} B \\ B^\top \mathcal{P} & -\sigma \mathbb{I}_{n_\mu} \end{bmatrix} < 0 \quad (1.41)$$

then the observer gain  $L_{nn} = \mathcal{P}^{-1} \mathcal{Q}^\top$  ensures that the estimation error  $\epsilon_{nn}$  satisfies the  $\mathcal{L}^2$  criterion (1.39) 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.  $\square$

### 5.1.2 Optimization of the Learning Parameter $\theta$

To approximate the unknown dynamics  $\mu_x(t)$ , the neural network parameters  $\theta$  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 :

$$\begin{aligned} \mathcal{L}(\theta) = & \underbrace{(\hat{x}_{nn} - x_{ref})^\top \mathcal{T}_{ref} (\hat{x}_{nn} - x_{ref})}_{\text{Tracking Error}} + \underbrace{(y - y_{nn})^\top \mathcal{T}_y (y - y_{nn})}_{\text{Output Error}} \\ & + \underbrace{(\hat{x}_{nn} - \hat{x}_{uio})^\top \mathcal{T}_{uio} (\hat{x}_{nn} - \hat{x}_{uio})}_{\text{UIO Consistency}} + \underbrace{\lambda \|\mu_\theta(\hat{x}_{nn}) - \hat{\mu}_x\|^2}_{\text{Regularization}} \end{aligned} \quad (1.42)$$

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

The terms in (1.42) are motivated as follows :

- **Tracking Error** : Penalizes deviations of the neural observer state  $\hat{x}_{nn}$  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  $\hat{x}_{nn}$  to remain close to the robust UIO estimate  $\hat{x}_{uio}$ , preventing divergence during the early phases of learning.
- **Regularization** : Directly guides the neural network output  $\mu_\theta$  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  $\theta$ . 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 :

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

Applying a forward Euler discretization with sampling period  $\Delta t$ , we obtain the discrete-time update law :

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

This can be written compactly as :

$$\hat{x}_{k+1} = f_{\text{obs}}(\hat{x}_k, \theta, k). \quad (1.43)$$

### 5.2.2 Gradient Computation via Sensitivity Analysis

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

$$\begin{aligned} \min_{\theta} \quad & J(\theta) = \sum_{k=0}^K \mathcal{L}_k(\hat{x}_k(\theta)) \\ \text{s.t.} \quad & \hat{x}_{k+1} = f_{\text{obs}}(\hat{x}_k, \theta, k). \end{aligned} \quad (1.44)$$

Since the state  $\hat{x}_k$  depends implicitly on  $\theta$  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  $\theta$  is given by the chain rule :

$$\nabla_\theta \mathcal{L}_k = \frac{\partial \mathcal{L}_k}{\partial \hat{x}_k} S_k + \frac{\partial \mathcal{L}_k}{\partial \mu_k} \left( \frac{\partial \mu_\theta}{\partial \theta} + \frac{\partial \mu_\theta}{\partial \hat{x}_k} S_k \right) \quad (1.45)$$

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 (1.42) are :

$$\frac{\partial \mathcal{L}_k}{\partial \hat{x}_k} = 2(\hat{x}_k - x_{\text{ref},k})^\top \mathcal{T}_{\text{ref}} - 2(y_k - C \hat{x}_k)^\top \mathcal{T}_y C + 2(\hat{x}_k - \hat{x}_{\text{UIO},k})^\top \mathcal{T}_{\text{ui}} \quad (1.46)$$

$$\frac{\partial \mathcal{L}_k}{\partial \mu_k} = 2\lambda(\mu_\theta(\hat{x}_k) - \hat{\mu}_{x,k})^\top. \quad (1.47)$$

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

$$S_{k+1} = S_k + \Delta t (A_k S_k + B_k) \quad (1.48)$$

where

$$A_k = \frac{\partial f_{\text{obs}}}{\partial \hat{x}_k} = \frac{\partial \varphi}{\partial \hat{x}_k} + B \frac{\partial \mu_\theta}{\partial \hat{x}_k} - L_{\text{nn}} C \quad (1.49)$$

$$B_k = \frac{\partial f_{\text{obs}}}{\partial \theta} = B \frac{\partial \mu_\theta}{\partial \theta}. \quad (1.50)$$

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 4** (Learning stability). Due to the nonlinear parameterization of deep neural networks,

*global convergence of the learning parameters  $\theta$  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  $\Delta_x$ ; in particular, if  $\Delta_x$  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  $\mu_x$  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. 1.2. 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,  $\hat{f}_{nn}^{(i+1)}$ , is a weighted ensemble of the models in the queue :

$$\hat{f}_{nn}^{(i+1)} = M_\psi \left( \hat{f}_{nn}^i, e^{i+1-p} f_\theta^{(i+1)} \right) \quad (1.51)$$

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

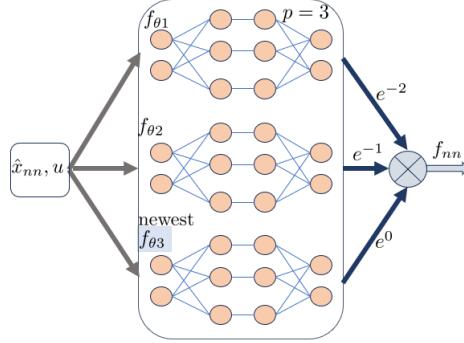


FIGURE 1.2 – Schematic of the continuous learning with forgetting mechanism.

## 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] :

$$\dot{v}_x = a - \frac{F_{yf} \sin(\delta)}{m} - \mu g + \dot{\psi} v_y \quad (1.52a)$$

$$\dot{v}_y = \frac{F_{yr}}{m} + \frac{F_{yf} \cos(\delta)}{m} - \dot{\psi} v_x \quad (1.52b)$$

$$\dot{\psi} = r \quad (1.52c)$$

$$\dot{r} = \frac{l_f F_{yf} \cos(\delta)}{I_z} - \frac{l_r F_{yr}}{I_z} \quad (1.52d)$$

where  $v_x$  and  $v_y$  are the longitudinal and lateral velocities at the center of gravity (CG),  $\psi$  is the yaw angle, and  $r = \dot{\psi}$  is the yaw rate. The control inputs are the longitudinal acceleration  $a$  and the steering angle  $\delta$ . 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) :

$$F_{yf} = C_f \alpha_f + f_{yf}(\alpha_f), \quad F_{yr} = C_r \alpha_r + f_{yr}(\alpha_r) \quad (1.53)$$

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

$$\alpha_f = \delta - \frac{v_y + l_f r}{v_x}, \quad \alpha_r = -\frac{v_y - l_r r}{v_x}. \quad (1.54)$$

Defining the state vector  $x = [v_x, v_y, \psi, r]^\top$  and assuming small steering angles ( $\cos(\delta) \approx 1, \sin(\delta) \approx \delta$ ), the system can be written in the following quasi-LPV form with additive unknown tire-force nonlinearities.

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.

$$\begin{cases} \dot{x} = A(v_x)x + Bu + E\mu_x(x, u) \\ y = Cx \end{cases} \quad (1.55)$$

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

$$\mu_x(x, u) := \begin{bmatrix} f_{yf}(\alpha_f) \\ f_{yr}(\alpha_r) \end{bmatrix}. \quad (1.56)$$

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

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

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

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.

## 7.2 Observer Design and Implementation

### 7.2.1 LPV Scheduling and Polytopic Embedding

The matrix  $A(v_x)$  in (1.55) 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

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

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

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

where the vertices  $A_j$  are obtained by evaluating  $A(v_x)$  at the corners of  $(\rho_1, \rho_2)$  (typically  $\bar{n} = 4$ ). This polytopic description is used to synthesize constant observer gains via LMIs.

### 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  $\mu_x$ . To satisfy the structural rank condition (1.4) for the augmented variable  $\zeta := [x^\top, \mu_x^\top]^\top$ , we use the regularized output

$$y = Cx + D_{\delta_2}\mu_x + D\omega_{\delta_2}, \quad (1.57)$$

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  $\text{rank}(D) = n_\mu$  (hence  $\text{rank}([\mathbb{E}^\top, \mathcal{H}^\top]^\top) = n_x + n_\mu$ ). A convenient choice for  $n_y = 3$  and  $n_\mu = 2$  is, for example,

$$D = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (1.58)$$

With this choice, the UIO can be written in the standard form (1.10) :

$$\begin{cases} \dot{\eta} &= P_\zeta \varphi_\zeta(\hat{\zeta}, u) + K(y - \mathcal{H}\hat{\zeta}) \\ \hat{\zeta} &= \eta + Q_\zeta y \end{cases} \quad (1.59)$$

where  $P_\zeta$  and  $Q_\zeta$  satisfy (1.9). For the LPV vehicle model, the Jacobian of  $\varphi_\zeta$  is polytopic ; its vertices can be constructed as

$$\mathcal{A}_{\zeta,j} := \begin{bmatrix} A_j & E \\ 0 & 0 \end{bmatrix}, \quad j = 1, \dots, \bar{n}. \quad (1.60)$$

Then, the UIO gain is obtained from the LMI synthesis in Theorem 1 : solve the LMIs (1.26) for all vertices  $j$ , recover the decision variables  $(\mathcal{P}, \mathcal{Q})$ , and compute

$$K = \mathcal{P}^{-1} \mathcal{Q}^\top. \quad (1.61)$$

### 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 :

$$\dot{\hat{x}}_{NN} = A(v_x)\hat{x}_{NN} + Bu + E\mu_\theta(\hat{x}_{NN}) + L_{NN}(y - C\hat{x}_{NN}). \quad (1.62)$$

The gain  $L_{NN}$  is designed by applying Theorem 2 to the polytopic LPV set  $\{A_j\}_{j=1}^{\bar{n}}$  (i.e., take  $\mathcal{A}_j^x := A_j$  in (1.41)), 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  $\Delta t$ ) 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

$$S_{k+1} = S_k + \Delta t (A_k S_k + B_k), \quad S_k := \frac{\partial \hat{x}_k}{\partial \theta}, \quad (1.63)$$

with the vehicle-dependent Jacobians

$$A_k = \frac{\partial f_{\text{obs}}}{\partial \hat{x}_k} = A(\rho_k) + E \frac{\partial \mu_\theta}{\partial \hat{x}_k} - L_{NN}C, \quad (1.64)$$

$$B_k = \frac{\partial f_{\text{obs}}}{\partial \theta} = E \frac{\partial \mu_\theta}{\partial \theta}. \quad (1.65)$$

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

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 1.3 and Figure 1.4 illustrate the state estimation results. While the longitudinal velocity  $v_x$  and yaw angle  $\psi$  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. 1.3) 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. 1.4), resulting in smooth and accurate tracking.

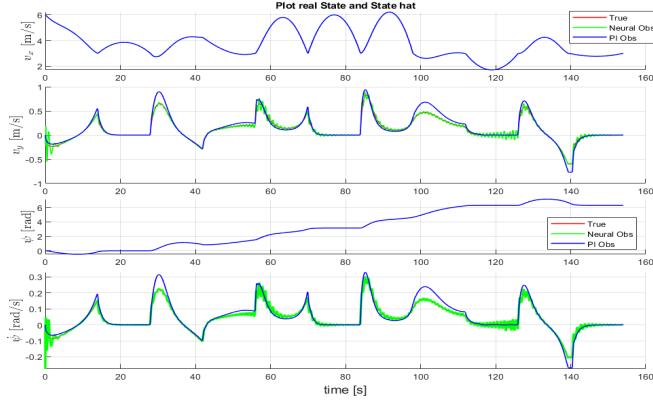


FIGURE 1.3 – State Estimation : Comparison between True State, Neural Observer, and PI Observer (Without Loss Accumulation).

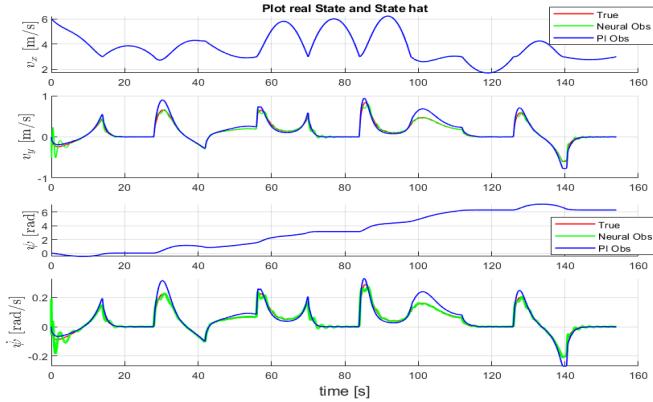


FIGURE 1.4 – State Estimation : Comparison with Loss Accumulation Strategy. Note the reduction in estimation noise and spikes.

### 7.3.2 Uncertainty Estimation

Figure 1.5 shows the estimation of the unknown dynamics  $\mu_x$ . 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.

### 7.3.3 Convergence Analysis

The evolution of the loss function during the simulation is depicted in Figure 1.6. 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.

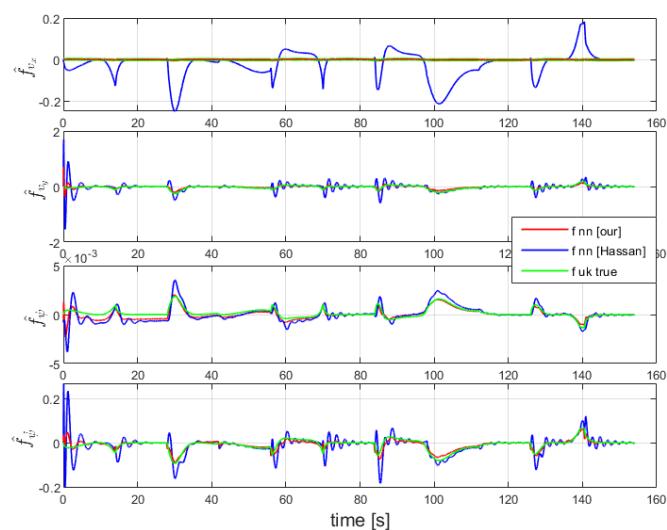


FIGURE 1.5 – Estimation of Unmodeled Dynamics : Comparison of the proposed method against baseline approaches.

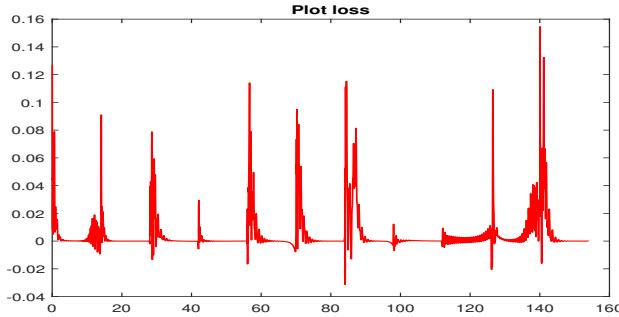


FIGURE 1.6 – Evolution of the Loss Function during the simulation.

## 8 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  $\text{rank}(CB) = \text{rank}(B)$  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. Future work will focus on the experimental validation of this framework using real driving data and its extension to discrete-time formulations for embedded implementation.

# Bibliographie

- [Benallouch, 2017] M. BENALLOUCH, R. OUTBIB, M. BOUTAYEB et 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) (cf. p. 4).
- [Chauouche, 2022] A. CHAOUCHE, A. ZEMOUCHE, M. RAMDANI, K. CHAIB DRAA et 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), p. 1372-1385 (cf. p. 4).
- [Charandabi, 2014] B.A. CHARANDABI et H.J. MARQUEZ. « A novel approach to unknown input filter design for discrete-time linear systems ». *Automatica* 50.2 (2014), p. 2835-2839 (cf. p. 4).
- [Hassan, 2013] L. HASSAN, A. ZEMOUCHE et M. BOUTAYEB. « Robust Unknown Input Observers For Nonlinear Time-Delay Systems ». *SIAM Journal on Control and Optimization* 51.4 (2013), p. 2735-2752 (cf. p. 4).
- [Hou, 1992] M. HOU et P. C. MULLER. « Design of observers for linear systems with unknown inputs ». *IEEE Transactions on Automatic Control* 37.6 (1992), p. 871-875 (cf. p. 4).
- [Jeon, 2020] W. JEON, Z. XIE, A. ZEMOUCHE et R. RAJAMANI. « Simultaneous cyber-attack detection and radar sensor health monitoring in connected ACC vehicles ». *IEEE Sensors Journal* 21.14 (2020), p. 15741-15752 (cf. p. 5).
- [Nguyen, 2024] Q.H. NGUYEN, O. SADKI, H. RAFARALAHY, M. HADDAD et A. ZEMOUCHE. « Cyberattack Detection by Using a Discrete-Time Model-Based Unknown Input Observer ». *IEEE Control Systems Letters* 8 (2024), p. 856-861 (cf. p. 5).
- [Phanomchoeng, 2014] G. PHANOMCHOENG et R. RAJAMANI. « Real-Time Estimation of Rollover Index for Trip-ped Rollovers With a Novel Unknown Input Nonlinear Observer ». *IEEE/ASME Transactions on Mechatronics* 19.2 (2014), p. 743-754 (cf. p. 4).
- [Trinh, 2011] H. TRINH et T. FERNANDO. *Functional observers for dynamical systems*. T. 420. Springer Science & Business Media, 2011 (cf. p. 4).
- [Trinh, 2008] H. TRINH, T.D. TRAN et T. FERNANDO. « Disturbance decoupled observers for systems with unknown inputs ». *IEEE Transactions on Automatic Control* 53.10 (2008), p. 2397-2402 (cf. p. 4).
- [Zemouche, 2009] Ali ZEMOUCHE et Mohamed BOUTAYEB. « Nonlinear-observer-based  $\mathcal{H}_\infty$  synchronization and unknown input recovery ». *IEEE Trans. Circuits Syst. I. Regul. Pap.* 56.8 (2009), p. 1720-1731 (cf. p. 4).



## RÉSUMÉ

---

## MOTS CLÉS

---

Estimation, mot clé 2, mot clé 3, mot clé 4

## ABSTRACT

---

The reliability and safety of Connected and Autonomous Vehicles (CAVs) rely heavily on the real-time availability of precise dynamic variables. However, equipping vehicles with high-end sensors to measure every necessary variable—such as sideslip angles or vehicle trajectories—is often economically unfeasible or physically impossible. Furthermore, CAVs are increasingly vulnerable to sensor faults and cyber-attacks targeting inter-vehicle communications (V2V/V2I), which can lead to critical failures. To address these challenges without incurring the high cost of physical hardware redundancy, this thesis proposes the development of an embedded electronic board functioning as a resilient "soft sensor" based on analytical redundancy.

The research adopts a hybrid modeling methodology that integrates physical differential equations with online learning-based neuro-adaptive neural networks. This approach allows for the accurate representation of complex, time-varying parameters and unknown inputs within the vehicle's dynamics. Based on these models, advanced nonlinear estimation algorithms—specifically Unknown Input Observers (UIO) and neuro-adaptive observers—are developed to reconstruct missing state variables and ensure system resilience against data loss and malicious signal injection.

The proposed estimation strategies are applied to two primary control challenges:

- Resilience to Cyber-Attacks in Adaptive Cruise Control (ACC): The thesis investigates specific architectures for detecting and mitigating cyber-attacks, such as Denial of Service (DoS) and False Data Injection (FDI), within autonomous and cooperative ACC systems.
- Vehicle Tracking and Platooning: Novel estimation algorithms are proposed to handle the nonlinear dynamics and unknown forces inherent in tracking surrounding vehicles, which is essential for decision-making in maneuvers like lane changes and platooning.

The final contribution of this work is the design and fabrication of a prototype embedded electronic board. This hardware integrates the developed algorithms, validating the theoretical contributions and demonstrating a cost-effective, scalable solution for enhancing the autonomy, security, and fault tolerance of next-generation vehicles.

## KEYWORDS

---

State Estimation, keyword 2, keyword 3, keyword 4