Robust and Sample-Efficient Estimation of Vehicle Lateral Velocity Using Neural Networks With Explainable Structure Informed by Kinematic Principles

Mauro Da Lio, *Member, IEEE*, Mattia Piccinini, *Member, IEEE*, and Francesco Biral

***Abstract*— This paper presents kinematics-structured neural networks (KS-NN) for the lateral speed estimation of vehicles. The internal structure of the networks is designed to incorporate the kinematic principles, enhancing the physical explainability and generalization capacity. Both the internal structure and training method are devised for better generalization performance. Various linear and nonlinear variants of our estimator are assessed for accuracy and robustness. The approach is validated using an openly accessible dataset with two race cars. The performance of the novel networks is evaluated against Luenberger, neural network, factor graph and Kalman filter estimators from the literature. In comparison with a Luenberger kinematic observer, our networks improve noise rejection, and overcome the well-known observability problem for low yaw rates. Compared to existing neural network estimators, our networks exhibit better generalization capacity, are more sample-efficient, require fewer learnable parameters, and their structure is physically explainable.**

***Index Terms*— Lateral speed estimation, explainable neural network, neural observer, sample-efficient neural network.**

I. Introduction

THE estimation of lateral velocity is essential in advanced vehicle control and stability systems [1], [2], [3], [4], [5], [6], [7], [8].

Since the direct measurement of the lateral velocity is feasible only with expensive optical sensors or dual-antenna Global Positioning Systems (GPS) —which are not affordable for most commercial vehicles— many estimators have flourished.

Among the estimation techniques, model-based approaches take a prominent role. The literature (see Section I-B) presents two main types of model-based observers: a) those using a model of the vehicle dynamics and b) those based on universal kinematic principles. The latter type does not depend on dynamical characteristics that may vary among vehicles and environmental conditions. However, the kinematic estimation suffers from observability problems at low yaw rates (e.g., on straights) and significant noise sensitivity.

Data-driven methods, especially neural networks, are alternatives to model-based approaches. They are trained on specific vehicles and environments for which they generally perform well. However, they are unexplainable black boxes. It is difficult to assess how much vehicle- and environment-specific they are, what they have learned, and whether they learned dynamical characteristics that may vary in operation.

A. Contribution of This Paper

The paper presents a specialized neural architecture for the physical problem at hand, offering several advantages:

- *Sample-efficiency*, by requiring fewer parameters to model the physical phenomenon accurately;
- *Ability to generalize*, resulting in robustness;
- *Explainability*, as it is an interpretable linear parameter varying discrete-time model;
- Being a linear parameter varying model, it can be *implemented easily* with automotive-grade hardware.

The study evaluates 28 variants of our specialized model in 4 groups, 10 variants of benchmark general-purpose recurrent neural networks in 2 groups (GRU and LSTM, from [9] and [10]), the benchmark Luenberger kinematic observer [11], and the benchmarks factor graph and Kalman filter [12].

The evaluation of the above candidates for accuracy and robustness uses an openly accessible dataset of telemetries from two real race cars, with challenging top speeds of 240 km/h and lateral accelerations up to m/s^2.

B. Related Work

The design of lateral speed estimators has been an active research topic for more than 30 years. The authors of [13] review the methods published until 2018, finding two broad categories: one based on vehicle models and the other purely data-driven. The former class can be split into three subgroups: dynamical vehicle models, kinematic models, or combinations of the two.

1) Dynamic Model-Based Observers: Observers adopting dynamical models typically depend on several vehicle parameters —including tire submodels—, which makes them vehicle and context specific. For example, [14] and [15] used

Manuscript received 27 October 2021; revised 15 April 2023; accepted 3 August 2023. Date of publication 21 August 2023; date of current version 29 November 2023. The Associate Editor for this article was A. Amditis. *(Corresponding author: Mattia Piccinini.)*

The authors are with the Department of Industrial Engineering, University of Trento, 38123 Trento, Italy (e-mail: mauro.dalio@unitn.it; mattia.piccinini@unitn.it; francesco.biral@unitn.it).

Digital Object Identifier 10.1109/TITS.2023.3303776

Kalman filters (KF) with single-track vehicle models. Using a dynamical vehicle model, a sliding mode estimator was adopted in [16], in combination with a KF. Linear, nonlinear, and sliding mode observers were compared in [17], using dynamic vehicle models restricted to moderate accelerations. The authors of [12] presented a factor graph approach based on a vehicle dynamic model.

However, the use of dynamic vehicle models usually results in a high sensitivity to variations in the environment, operative conditions and/or vehicle parameters.

2) Kinematic Model-Based Observers: Kinematic observers exploit the principles of kinematics to estimate the vehicle velocity components, using the measured accelerations and angular rate. One seminal paper is [11], which developed a parameter-varying Luenberger observer such that the estimation error vanishes for a non-zero yaw rate. The approach of [11] is quite robust to variations in the physical parameters of the vehicle, which do not appear in the formulation. The authors of [18] extended the Luenberger estimator with an adaptive dead zone in the feedback terms, which permits larger gains to compensate for lateral acceleration biases. Starting from [11], a heuristic correction term was added in [19] to bring the estimated lateral speed to zero when the vehicle is driving straight. The lateral speed estimation in [20] and [21] was improved by utilizing the measurement bias estimates of an inertial measurement unit (IMU). Extended KFs were implemented in [22], [23], and [24] using the two- and three-dimensional kinematic models.

However, using basic kinematic estimators, the lateral speed becomes unobservable for low yaw rates. In addition, the estimation degrades in the presence of biases on the measured accelerations (e.g., due to gravity when the chassis rolls or pitches and when the road bank and grade angles are not negligible).

3) Mixed Kinematic-Dynamic Model-Based Observers: Mixed kinematic-dynamic methods were introduced in [25], [26], [27], and [28] to overcome the limitations of purely kinematic estimators. In these papers, lateral vehicle dynamic models improve the lateral speed observability for low yaw rates. The authors of [29] proposed a fuzzified weighted mean of the estimates obtained with a kinematic observer and with a dynamic model-based KF.

Compared with pure kinematic estimators, mixed kinematic-dynamic methods inevitably increase the sensitivity to changes in operative conditions and vehicle parameters.

4) Data-Driven Observers: Some authors used neural networks (NNs) in pure data-driven black-box state estimation methods. For example, NNs using gated recurrent units (GRUs) were employed in [9] and [30] to estimate the lateral speed with end-to-end approaches. Similarly, [10] used a long short-term memory (LSTM) network. A simple single-hidden-layer NN was presented in [31] to estimate the side slip using only simulated tests. A similar NN was adopted in [32] to predict the lateral tire forces, which were then fed to an extended KF for state estimation. The authors in [33] designed a single-hidden-layer neural observer using the set membership theory. To deal with variable road friction, [34] combined three nonlinear auto-regressive networks (ARX), to learn the side slip in three different adherence conditions.

However, pure data-driven neural network estimators tend to suffer from overfitting issues, since they implicitly learn vehicle dynamic characteristics that depend on the operating conditions used during training.

To the best of the authors' knowledge, the lateral speed estimators in the literature are limited by at least one of the following aspects:

1) The neural networks in data-driven estimation (e.g., GRU, LSTM) [9], [10], [30] usually overfit the vehicle and environment conditions used for training.
2) The neural networks' generic internal structures do not follow kinematic laws, and the learnable parameters cannot be physically explained.
3) All the neural and many model-based kinematic estimators rely on measuring the steering wheel angle. However, this can decrease the robustness of the estimator: the relation between the steering angle and the lateral velocity depends on vehicle characteristics (e.g., tires and steering system), which can vary with different tires, road surfaces, and vehicle types.
4) Some kinematic estimators require an IMU sensor, e.g., to estimate the roll angle [19]. However, an IMU is not affordable for most commercial cars.

C. Structure of the Paper

Section II discusses the principles of traditional kinematic estimation and its limitations. Section III presents the kinematics-driven internal structure of the proposed neural networks. The interpretation of the structure of the neural network is explained in more detail in Section IV. Section V discusses the literature benchmarks used for comparison. Section VI introduces the open experimental race car dataset used to validate the approach, and the training method. Section VII analyzes the measurement noise and dynamic characteristics of the two race cars used in the dataset. Section VIII discusses the main findings, highlighting the improved robustness of the new networks. Finally, Section IX concludes the study and suggests areas for future research.

II. Principles and Limitations of Traditional Kinematic Estimation

This section provides an overview of the principles of traditional kinematic lateral speed estimation, along with a discussion of their limitations. Understanding these principles is crucial to comprehend the proposed neural network architecture, and its advantages over conventional methods.

A. Measurement Principle

The direct measurement of a vehicle's lateral velocity (v) is not easily affordable, but estimators can give indirect measurements. Kinematic estimators exploit measurements of the angular velocity ($\boldsymbol{\omega}$), acceleration ($\boldsymbol{a}$), and forward speed (u) with the following principle. Let:

$$\boldsymbol{v} = u\boldsymbol{i} + v\boldsymbol{j} + w\boldsymbol{k} \tag{1}$$

be the velocity vector expressed in a moving reference frame $\{\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}\}$, where u, v, w are the components of $\boldsymbol{v}$ in that frame.

The time derivative of (1) reads as:

$$\boldsymbol{a} = \dot{u}\boldsymbol{i} + \dot{v}\boldsymbol{j} + \dot{w}\boldsymbol{k} + u(\boldsymbol{\omega} \times \boldsymbol{i}) + v(\boldsymbol{\omega} \times \boldsymbol{j}) + w(\boldsymbol{\omega} \times \boldsymbol{k}) \quad (2)$$

where the Poisson's formulas are used to express the derivatives of $\{\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}\}$ (e.g., $\frac{d\boldsymbol{i}}{dt} = \boldsymbol{\omega} \times \boldsymbol{i}$). After computing the cross products, the projection of (2) on the axes $\{\boldsymbol{i}, \boldsymbol{j}\}$ is:

$$\begin{cases} a_y = \dot{v} + u\omega_z - w\omega_x & (3a) \\ a_x = \dot{u} - v\omega_z + w\omega_y & (3b) \end{cases}$$

where $\{\omega_x, \omega_y, \omega_z\}$ are the components of $\boldsymbol{\omega}$ in $\{\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}\}$.

If the three-dimensional motion of the vehicle is approximated in two dimensions ($\omega_x = \omega_y = 0$, $w = 0$ and $\omega_z = \omega$), with $\boldsymbol{i}$ being the vehicle longitudinal axis on a flat ground and $\boldsymbol{j}$ pointing leftwards ($\boldsymbol{k}$ upwards), the planar kinematic model is finally obtained:

$$\begin{cases} \dot{v} = a_y - u\omega & (4a) \\ \dot{u} = a_x + v\omega & (4b) \end{cases}$$

Since a_y, u and ω can be measured by on-board sensors, (4a) provides a means to compute $\dot{v}$, and hence, in principle, an estimation $\hat{v}$ of v could be obtained via integration. However, besides the approximation from (3a) to (4a), a_y, u and ω are noisy,[1] and their integration according to (4a) will diverge following a partially random walk.

The drift can be attenuated by using both equations (4a) and (4b): the velocity $\hat{v}$ integrated by the former is used in the latter, to compute the expected longitudinal velocity $\hat{u}$. Since the actual longitudinal velocity can be measured (u_{m}), the quantity $\hat{u} - u_{\mathrm{m}}$ becomes an indicator of the accumulated drift, which can be injected in (4a) as follows:

$$\begin{cases} \dot{\hat{v}} = a_y - \hat{u}\omega - \bar{f}_1(\omega, \hat{u} - u_{\mathrm{m}}) & (5a) \\ \dot{\hat{u}} = a_x + \hat{v}\omega - \bar{f}_2(\omega, \hat{u} - u_{\mathrm{m}}) & (5b) \end{cases}$$

where $\bar{f}_1(\omega, \hat{u} - u_{\mathrm{m}})$ and $\bar{f}_2(\omega, \hat{u} - u_{\mathrm{m}})$ are suitable feedback corrections, which can be designed as functions of the yaw rate ω and of $\hat{u} - u_{\mathrm{m}}$.

Equations (5a) and (5b) are the basic prototype of kinematic observers.

B. Unobservability Issues of Pure Kinematic Estimation

Using the basic kinematic model (4a,4b) and measuring the forward speed u_{m}, the lateral speed $\hat{v}$ becomes unobservable as the yaw rate ω approaches zero (a proof can be found in [11]). As a consequence, the estimators based on pure kinematic models, like the Luenberger observer of [11], need to change their estimation model or be reset for low yaw rate values.

The structure of the novel neural model of this paper is (among the others) designed to overcome the observability issue.

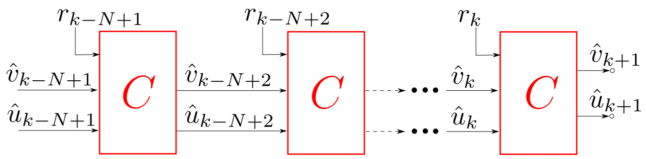


Fig. 1. The proposed KS-NN model is a memoryless recurrent neural network (RNN). The input of KS-NN is a window of N past measurements $\{\boldsymbol{r}_{k-N+1}, \ldots, \boldsymbol{r}_k\}$, with $\boldsymbol{r} = \{\omega, u_{\mathrm{m}}, a_y, a_x\}$. The recurrence to compute $\{\hat{v}_{k+1}, \hat{u}_{k+1}\}$ starting from $\{\hat{v}_{k-N+1}, \hat{u}_{k-N+1}\} = \{0, u_{\mathrm{m}_{k-N+1}}\}$ is unrolled, for an easier graphical representation. The inner structure of the C blocks is depicted in Fig. 2(a).

III. Neural Network Estimator With Kinematics-Driven Structure

This section outlines the design of the proposed lateral velocity neural network estimator. Our estimator is labeled KS-NN (*kinematics-structured neural network*), and its internal structure is designed to generalize the kinematic laws.

Let us begin with an overview of the design and operating principle of KS-NN.

We collect in the vector $\boldsymbol{r} = \{\omega, u_{\mathrm{m}}, a_y, a_x\}$ the input measurements required by KS-NN, namely the yaw rate ω, the forward speed u_{m}, and the accelerations $\{a_y, a_x\}$.

KS-NN operates as a memoryless recurrent neural network (RNN), i.e., as a finite impulse response (FIR) system. As shown in Fig. 1, at each discrete time step k, a window of N past measurements $\{\boldsymbol{r}_{k-N+1}, \ldots, \boldsymbol{r}_{k-1}, \boldsymbol{r}_k\}$ is used to estimate the states $\{\hat{v}_{k+1}, \hat{u}_{k+1}\}$, starting from a standard initial state $\{\hat{v}_{k-N+1}, \hat{u}_{k-N+1}\} = \{0, u_{\mathrm{m}_{k-N+1}}\}$. In the following implementations, the sampling time of the KS-NN model is $\tau = 0.05$ s, and $N = 30$, so that the input windows of past measurements span 1.5 s ($= N\tau$), which means that we ask the estimator to converge within 1.5 s.

Fig. 2(a) shows the internal architecture of the recurrent network (the C block appearing in Fig. 1). It is a bank of Luenberger-like modules (M_p) that are combined with the channel coding technique [36, Section III.A-1]. Each module M_p is a local model, activated by a function $\phi_p(|\omega|)$ with local support. Hence, each module learns to estimate $\hat{v}$ in a specific range of yaw rates (ω). There can be many local models M_p, $p \in \{1, \ldots, \mathcal{P}\}$ to improve the descriptive capacity of the network at varying yaw rates.

The structure of each module is similar to the Luenberger estimator [11], but with several learnable weights and a possibly nonlinear feedback function, with $\mathcal{Q}$ hidden neurons.

Let us see more in detail the structure of the models and how they are combined with channel coding.

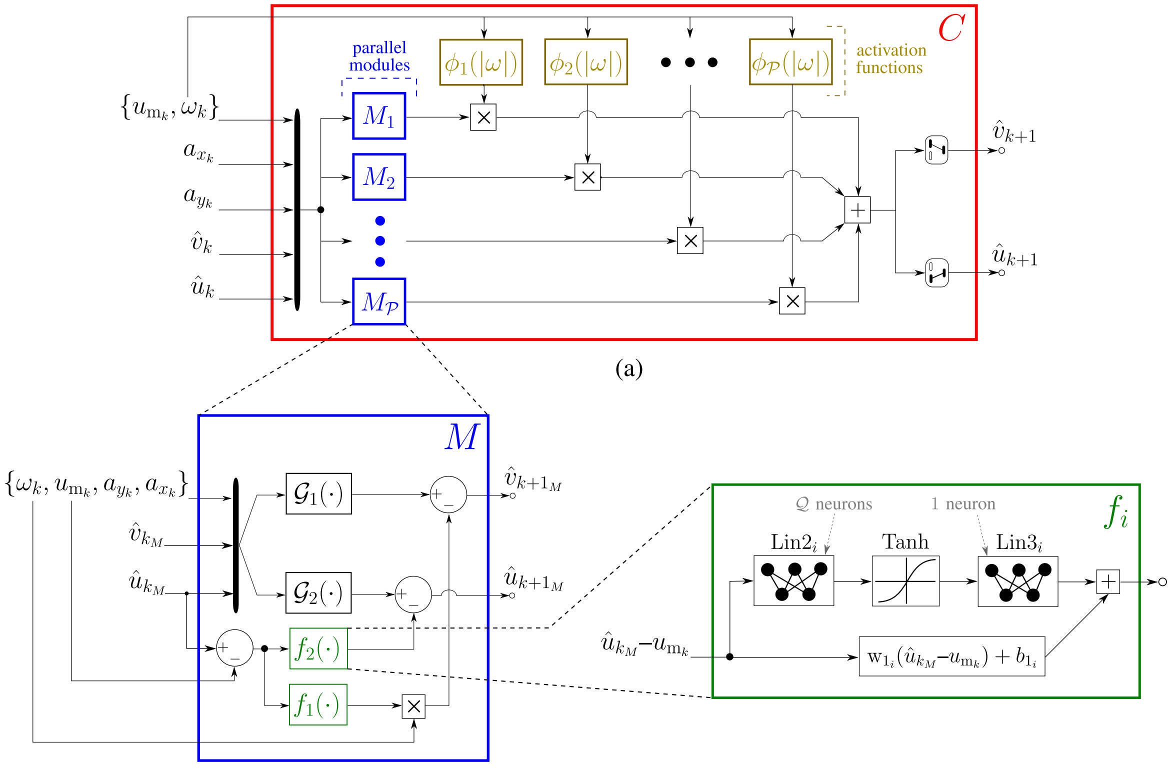
A. Custom Neural Network Module M

The design of the local models $\{M_1, \ldots, M_{\mathcal{P}}\}$, depicted in Fig. 2(b), is based on a neural module, named M, which we now illustrate. Let us start from the explicit Euler discretization of the basic kinematic observer (5a,5b):

$$\begin{cases} \hat{v}_{k+1} = \hat{v}_k + \tau\left(a_{y_k} - \hat{u}_k\omega_k - \bar{f}_1(\omega_k, \hat{u}_k - u_{\mathrm{m}_k})\right) & (6a) \\ \hat{u}_{k+1} = \hat{u}_k + \tau\left(a_{x_k} + \hat{v}_k\omega_k - \bar{f}_2(\omega_k, \hat{u}_k - u_{\mathrm{m}_k})\right) & (6b) \end{cases}$$

where $\{\hat{v}_k, \hat{u}_k\}$ are the estimates of $\{v, u\}$ at the time step k, and τ is the integration time step. We now cast the basic kinematic

[1] The acceleration measurements $\{a_y, a_x\}$ contain biases if the chassis rolls or pitches, or if the road bank and grade angles are non zero. The estimation of such angles [35] can help to compensate the biases, to some extent.



(a)

(b)

(c)

Fig. 2. Internal architecture of the recurrent block C of Fig. 1: (a) combination of $\mathcal{P}$ parallel modules of type M, to define the overall neural model C (9); (b) internal structure of the neural network module M (7a,7b); (c) internal structure of the $\{f_1(\cdot), f_2(\cdot)\}$ feedback sub-networks (8) composing the module M.

model (6a,6b) into a neural network module M, using the following recursion scheme:

$$M: \begin{cases} \hat{v}_{k+1_M} = \gamma_1 \hat{v}_{k_M} + \beta_1 \tau \cdot (a_{y_k} - u_{\mathrm{m}_k} \omega_k) + \\ \qquad\qquad - \omega_k \cdot f_1(\hat{u}_{k_M} - u_{\mathrm{m}_k}) \\ \qquad = \mathcal{G}_1(\hat{v}_{k_M}, a_{y_k}, u_{\mathrm{m}_k}, \omega_k) + \\ \qquad\qquad - \omega_k \cdot f_1(\hat{u}_{k_M} - u_{\mathrm{m}_k}) & \text{(7a)} \\ \hat{u}_{k+1_M} = \gamma_2 \hat{u}_{k_M} + \beta_2 \tau \cdot (a_{x_k} + \hat{v}_{k_M} \omega_k) + \\ \qquad\qquad - f_2(\hat{u}_{k_M} - u_{\mathrm{m}_k}) \\ \qquad = \mathcal{G}_2(\hat{v}_{k_M}, \hat{u}_{k_M}, a_{x_k}, \omega_k) + \\ \qquad\qquad - f_2(\hat{u}_{k_M} - u_{\mathrm{m}_k}) & \text{(7b)} \end{cases}$$

In (7a,7b), $\{\hat{v}_{k_M}, \hat{u}_{k_M}\}$ are the estimates obtained with the module M at the time step k. In comparison with the basic observation model (6a,6b), the proposed module (7a,7b) has additional learnable quantities, namely the parameters $\{\gamma_i, \beta_i\}$, $i \in \{1, 2\}$, and the weights and biases contained in the neural feedback functions $\{f_1(\cdot), f_2(\cdot)\}$, which will be described next. Equations (7a,7b) can therefore be seen as a generalized explicit Euler discretization of (5a,5b), in which the time step is τ. The $\{\mathcal{G}_1(\cdot), \mathcal{G}_2(\cdot)\}$ functions in (7a,7b) are used to facilitate a graphical interpretation of the module M, provided in Fig. 2(b).

As will be described in Section IV, all the learnable parameters in (7a,7b) have a physically explainable role, improve the noise rejection capabilities of the observer, and overcome the unobservability problem of the basic kinematic model during straight driving.

The learnable feedback functions $f_i(\cdot)$ of (7a,7b), $i \in \{1, 2\}$, are designed with a two-branch shallow neural architecture:

$$f_i(\hat{u}_{k_M} - u_{\mathrm{m}_k}) = \mathrm{w}_{1_i}(\hat{u}_{k_M} - u_{\mathrm{m}_k}) + b_{1_i} + \sum_{j=1}^{\mathcal{Q}} \mathrm{w}_{3_{ji}} \mathrm{Tanh}\Big(\mathrm{w}_{2_{ji}}(\hat{u}_{k_M} - u_{\mathrm{m}_k}) + b_{2_{ji}}\Big) \tag{8}$$

where $i \in \{1, 2\}$. As shown in Fig. 2(c), the $f_i(\cdot)$ functions in (8) have a linear and a nonlinear branch. The linear branch computes $\mathrm{w}_{1_i}(\hat{u}_{k_M} - u_{\mathrm{m}_k}) + b_{1_i}$, with the learnable parameters $\{\mathrm{w}_{1_i}, b_{1_i}\}$. The nonlinear branch combines two fully-connected layers: the first layer ("$\mathrm{Lin2}_i$" in Fig. 2(c)) has $\mathcal{Q}$ neurons and a Tanh activation function, while the second layer ("$\mathrm{Lin3}_i$") has 1 neuron and a linear activation function. Each of the $f_i(\cdot)$ functions in (8), $i \in \{1, 2\}$, has a maximum of $3\mathcal{Q} + 2$ learnable parameters, namely the weights $\{\mathrm{w}_{1_i}, \mathrm{w}_{2_{ji}}, \mathrm{w}_{3_{ji}}\}$ and the biases $\{\mathrm{b}_{1_i}, \mathrm{b}_{2_{ji}}\}$, $j \in \{1, \ldots, \mathcal{Q}\}$. The number $\mathcal{Q}$ of neurons might be varied, together with the number of hidden layers, to change the complexity of the $f_i(\cdot)$ networks.

The maximum number of learnable parameters in the neural module M is $N_p = 4 + 2 \cdot (3\mathcal{Q} + 2) = 6\mathcal{Q} + 8$.

B. Combining Multiple Parallel Modules

Within the neural module M (7a,7b), the learnable parameters are $\{\gamma_i, \beta_i\}$ and the weights and biases of the functions $f_i(\cdot)$ in (8), $i \in \{1, 2\}$.

The theory of vehicle dynamics [37] suggests that the behavior of the lateral velocity could vary with the magnitude

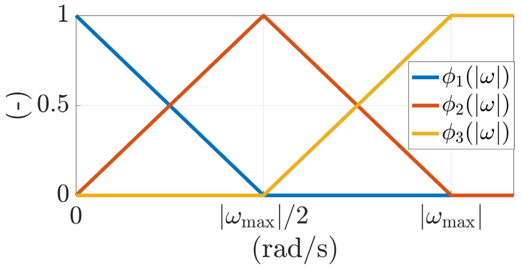


Fig. 3. This figure shows the activation functions $\phi_p(|\omega|)$ of the local neural models M_p, where $p \in \{1,\ldots,\mathcal{P}\}$. The centers of the $\phi_p(|\omega|)$ functions are equally spaced over the range of recorded yaw rate ω values, which is denoted as $\left[0, |\omega_{\max}|\right]$. The plot is an example with three local models, $\mathcal{P} = 3$.

of the yaw rate $|\omega|$, and therefore the optimized parameters of the neural module M should also vary with $|\omega|$. Following this physical consideration, the channel coding technique [36, Section III.A-1] is employed to create $\mathcal{P}$ parallel models of type M, named $\{M_1, \ldots, M_{\mathcal{P}}\}$ and shown in Fig. 2(a). The outputs of the M_p models are weighted by the activation functions $\phi_p(|\omega|)$, $p \in \{1,\ldots,\mathcal{P}\}$. As shown in Fig. 3, the $\phi_p(|\omega|)$ are by design piecewise linear in $|\omega|$, they partition the unity, that is, $\sum_{p=1}^{\mathcal{P}} \phi_p(|\omega|) = 1$, and they have local support: they are non-zero in a neighborhood of their center $|\omega| = \omega_p$. The centers ω_p are defined to partition the range $\left[0, |\omega_{\max}|\right]$ of the recorded yaw rate values (Fig. 3). In this way, a model M_p locally estimates $\{\hat{v}, \hat{u}\}$ in the vicinity of $|\omega| = \omega_p$.

Notably, the functions $\phi_p(|\omega|)$ are also called receptive fields, validity functions, or membership functions [38] in different contexts. Interested readers can find more details about our implementation in [36] and [39, Section III.A-1], where we discuss the relationship with similar approaches in biology and other engineering applications.

The complete local model approximation C is written as follows, and implemented in the network shown in Fig. 2(a):

$$C: \begin{bmatrix} \hat{v}_{k+1} \\ \hat{u}_{k+1} \end{bmatrix} = \sum_{p=1}^{\mathcal{P}} \phi_p(|\omega_k|) \cdot \begin{bmatrix} \hat{v}_{k+1_{M_p}} \\ \hat{u}_{k+1_{M_p}} \end{bmatrix} \tag{9}$$

where $\{\hat{v}_{k+1_{M_p}}, \hat{u}_{k+1_{M_p}}\}$ are the estimates computed with the module M_p, $p \in \{1,\ldots,\mathcal{P}\}$, using (7a,7b). The model C in (9) defines the interpolation of the estimates of the local models. The maximum number of trainable parameters for KS-NN is $N_p\mathcal{P} = (6\mathcal{Q} + 8)\mathcal{P}$. Note that the number of neurons $\mathcal{Q}$ and the number of local models $\mathcal{P}$ are adjustable hyperparameters, through which one can change the complexity and learning capability of the model. The role of $\mathcal{Q}$ and $\mathcal{P}$ is conceptually similar to the number of hidden layers and states of traditional recurrent neural networks [38].

IV. Interpretation of the Neural Network Structure

In this section, we show how the proposed neural network KS-NN can be interpreted in various ways.

A. Intepreting the Extended Kinematic Laws

In ideal conditions, with no measurement noise and on a perfectly planar surface, the evolution of the lateral speed could be in principle predicted through the integration of the exact kinematics, represented by equations (4a,4b). However, the existence of sensor noise, and the variation in chassis roll and pitch angles make the integration of (4a,4b) diverge. The feedback functions $\bar{f}_1(\cdot)$ and $\bar{f}_2(\cdot)$ in (5a,5b) are used to correct the divergence.

The model presented in equations (7a,7b) adapts the kinematic laws to the noise levels in the measured channels: compared to the discrete-time kinematic estimator (6a,6b), it has additional learnable parameters $\{\gamma_1, \gamma_2, \beta_1, \beta_2\}$, and feedback functions $\{f_1(\cdot), f_2(\cdot)\}$ that are more general than those used in the Luenberger observer [11] (Appendix A). The additional parameters and extended feedback functions improve noise rejection and overcome the unobservability issue.

1) Interpreting the Output of KS-NN: We can decompose the lateral velocity estimate produced by KS-NN into three independent and fused estimates, each of which has a physical interpretation. To simplify the analysis, let us focus on a single neural module M, represented by equations (7a,7b), and consider the case where the feedback functions $\{f_1(\cdot), f_2(\cdot)\}$ in (8) are linear ($\mathcal{Q} = 0$):

$$\begin{cases} f_1(\hat{u}_k - u_{\mathrm{m}_k}) = \mathrm{w}_{1_1} \cdot (\hat{u}_k - u_{\mathrm{m}_k}) + b_{1_1} & \text{(10a)} \\ f_2(\hat{u}_k - u_{\mathrm{m}_k}) = \mathrm{w}_{1_2} \cdot (\hat{u}_k - u_{\mathrm{m}_k}) + b_{1_2} & \text{(10b)} \end{cases}$$

Since the resulting neural module M (7a,7b) is linear, it is possible to apply the $\mathcal{Z}$ transform, obtaining:

$$\begin{aligned} \hat{v}(z) = {} & \overbrace{\frac{A(z)}{D(z)}\big(a_y(z) - \omega u_{\mathrm{m}}(z)\big)}^{\text{first sub-model}} \\ & \underbrace{+ \frac{B(z)}{D(z)}\left(u_{\mathrm{m}}(z)\frac{z-1}{\tau} - a_x(z)\right)}_{\text{second sub-model}} + \underbrace{\frac{C(z)}{D(z)}\omega}_{\text{third sub-model}} \end{aligned} \tag{11}$$

where the quantities $\{A(z), B(z), C(z), D(z)\}$ are given by:

$$\begin{cases} A(z) = \tau\beta_1(z + \mathrm{w}_{1_2} - \gamma_2), \quad B(z) = \tau\beta_2\mathrm{w}_{1_1}\omega & \text{(12a)} \\ C(z) = \big(\mathrm{w}_{1_1}(z - z\beta_2 + \beta_2 - \gamma_2)\big)u_{\mathrm{m}}(z) \\ \qquad + \dfrac{\big(b_{1_2}\mathrm{w}_{1_1} - b_{1_1}(z + w_{1_2} - \gamma_2)\big)z}{z-1} & \text{(12b)} \\ D(z) = z^2 + z(\mathrm{w}_{1_2} - \gamma_1 - \gamma_2) + \mathrm{w}_{1_1}\beta_2\tau\omega^2 + \\ \qquad - \mathrm{w}_{1_2}\gamma_1 + \gamma_1\gamma_2 & \text{(12c)} \end{cases}$$

The equation (11) shows that $\hat{v}$ is determined by merging three sub-models:

- The first sub-model estimates $\hat{v}$ given the measurements of the slip rate $a_y - \omega u_{\mathrm{m}}$, which is the right-hand side of the basic kinematic model (4a).
- The second sub-model uses the quantity $u_{\mathrm{m}}(z) \cdot (z-1)/\tau - a_x(z)$, which is another estimate of v derived from (4b), i.e. $v = (\dot{u} - a_x)/\omega$.
- The third sub-model estimates the lateral velocity $\hat{v}$ by learning the relationship between v and ω, for a certain forward speed u_{m} contained in $C(z)$.[2]

[2] This third model implicitly considers vehicle dynamics, which can vary among different cars, but is weighted against the previous two models.

The learnable transfer functions $A(z), B(z), C(z), D(z)$ weight the three sub-models. It is worth noting that equations (11) and (12) apply to one neural module M. However, KS-NN employs $\mathcal{P}$ modules M, which activate in specific ω intervals through channel coding. This approach allows the transfer function's pole-zero maps and gains to adapt to varying ω, enhancing the model's learning ability.

The role of the main learnable parameters in (11,12) will now be explained in detail.

2) Explaining the Role of γ_1: We now focus on (7a), assuming a fixed $\beta_1 = 1$ and a learnable $\gamma_1 \in (0, 1]$. The term $\gamma_1 \hat{v}_k + \tau(a_{y_k} - u_{\mathrm{m}_k}\omega_k)$ in (7a), with $\gamma_1 \in (0, 1]$, operates like a leaky integrator, accumulating an estimate of the lateral velocity, while forgetting some less recent input noise. During straight driving ($\omega \approx 0$), the parameter $\gamma_1 < 1$ helps to let $\hat{v}$ converge to zero, therefore overcoming the observability problem of the basic kinematic model for low yaw rates.[3] The measured yaw rate ω_k is introduced in the $\omega_k f_1(\cdot)$ term of (7a), to force (7a) to be decoupled from (7b) during straight driving, so that $\hat{v} \to 0$.

In the proposed approach, each of the $M_1, \dots, M_{\mathcal{P}}$ local models can learn different values of γ_1. However, to preserve the kinematic relations, the learned γ_1 values are close to 1 in the local models at higher yaw rates, where there are no observability issues.

3) Explaining the Role of γ_2 and β_2: The learnable parameters γ_2 and β_2 in (7b) modify the interpretation of the state $\hat{u}$, which is no longer necessarily the true forward speed, but rather a transformed version of it, that is optimized for improved estimation accuracy of $\hat{v}$.

The parameters γ_2 and β_2 modify the weighting transfer function $C(z)$ in (12b), by adding a term proportional to u_{m} when they are not equal to 1. This modification enables the third sub-model in (11) to learn a dynamic relationship among v, ω, and u_{m}, that cannot be captured by the basic kinematic estimator (6a,6b) or the Luenberger observer of [11].

The parameters γ_2 and β_2 also represent additional degrees of freedom to learn the transfer functions $\{A(z), B(z), C(z), D(z)\}$ in (11), which further improves the observer performance.

4) Explaining the Role of β_1: The learnable parameter β_1 in (7a) generalizes the kinematic relation between v and the slip rate $a_y - \omega u_{\mathrm{m}}$, which improves noise rejection. The optimized β_1 values in the neural modules $M_1, \dots, M_{\mathcal{P}}$ are typically lower than 1. As a result, (7a) still integrates $a_y - \omega u_{\mathrm{m}}$, but the integrand has a reduced weight (β_1), which leads to improved noise rejection. The missing contribution is provided by the term $\omega_k f_1(\hat{u}_k - u_{\mathrm{m}_k})$ in (7a), which represents a lateral velocity dynamic model that is learned with the state $\hat{u}$.

5) Explaining the Role of b_{1_1} and b_{1_2}: The biases $\{b_{1_1}, b_{1_2}\}$ in (10a) have the capacity to learn the measurement biases specific to a particular vehicle and sensor setup. However, this specialization can reduce the estimator's robustness, as discussed in Section VIII.

B. Interpreting KS-NN as an LPV Model

This section shows that the proposed KS-NN model can be interpreted as a linear parameter varying (LPV) system.

The combination of the local neural modules $\{M_1, \dots, M_{\mathcal{P}}\}$ in (9) can be expanded using (7a,7b):

$$\begin{cases} \hat{v}_{k+1} = \displaystyle\sum_{p=1}^{\mathcal{P}} \phi_p(|\omega_k|) \cdot \big(\gamma_{1_p}\hat{v}_k + \beta_{1_p}\tau(a_{y_k} - \omega_k u_{\mathrm{m}_k}) + \\ \qquad\qquad - \omega_k f_{1_p}(\hat{u}_k - u_{\mathrm{m}_k})\big) & \text{(13a)} \\ \hat{u}_{k+1} = \displaystyle\sum_{p=1}^{\mathcal{P}} \phi_p(|\omega_k|) \cdot \big(\gamma_{2_p}\hat{u}_k + \beta_{2_p}\tau(a_{x_k} + \omega_k \hat{v}_k) + \\ \qquad\qquad - f_{2_p}(\hat{u}_k - u_{\mathrm{m}_k})\big) & \text{(13b)} \end{cases}$$

If the feedback functions $\{f_{1_p}(\cdot), f_{2_p}(\cdot)\}$ are linear, then the dynamics can be represented as a linear parameter varying (LPV) discrete-time system. This is because, when performing the products among $\phi_p(|\omega_k|)$ and the M_p models in (13a,13b), the resulting equations can be written in a linear form, with parameters that vary with the value of $|\omega_k|$.

$$\begin{cases} \hat{v}_{k+1} = \gamma_1(|\omega_k|)\hat{v}_k + \beta_1(|\omega_k|)\tau(a_{y_k} - \omega_k u_{\mathrm{m}_k}) + \\ \qquad - \omega_k \tilde{f}_1(\hat{u}_k - u_{\mathrm{m}_k}, |\omega_k|) & \text{(14a)} \\ \hat{u}_{k+1} = \gamma_2(|\omega_k|)\hat{u}_k + \beta_2(|\omega_k|)\tau(a_{x_k} + \omega_k \hat{v}_k) + \\ \qquad - \tilde{f}_2(\hat{u}_k - u_{\mathrm{m}_k}, |\omega_k|) & \text{(14b)} \end{cases}$$

where the linear functions $\tilde{f}_i(\hat{u}_k - u_{\mathrm{m}_k}, |\omega_k|)$, $i \in \{1, 2\}$, are a parameter-varying extension of (10a,10b):

$$\begin{aligned} \tilde{f}_i(\hat{u}_k - u_{\mathrm{m}_k}, |\omega_k|) &= \sum_{p=1}^{\mathcal{P}} \phi_p(|\omega_k|) \cdot f_{i_p}(\hat{u}_k - u_{\mathrm{m}_k}) \\ &= \mathrm{w}_{1_i}(|\omega_k|) \cdot (\hat{u}_k - u_{\mathrm{m}_k}) + b_{1_i}(|\omega_k|) \quad (15) \end{aligned}$$

The functions $\{\gamma_1(|\omega|), \gamma_2(|\omega|), \beta_1(|\omega|), \beta_2(|\omega|)\}$, together with $\{\mathrm{w}_{1_i}(|\omega|), b_{1_i}(|\omega|)\}$, $i \in \{1, 2\}$, inherit from $\phi_p(|\omega|)$ the piecewise linearity in $|\omega|$ (Fig. 3), providing the interpolation among the local models.

V. Literature Benchmarks

The neural network KS-NN is compared with the following literature benchmarks:

- Kinematic Luenberger observer [11], with 2 trainable parameters. The implementation is detailed in Appendix A.
- Gated recurrent network (GRU). The implementation follows the E1 estimator proposed by [9], using the same input signals of KS-NN.
- Long short-term memory network (LSTM). The implementation follows [10], using the same input signals of KS-NN.
- Factor graph (FG) and Kalman filter (KF) estimators of [12]. The authors used the same race car dataset as our paper; hence their results are directly comparable with ours.

[3] The parameter $\gamma_1 \in (0, 1]$ plays a role similar to the threshold ω_{th} used to switch off the benchmark Luenberger observer [11] for $|\omega| < \omega_{\mathrm{th}}$ (refer to the Appendix A).

TABLE I

STATISTICS OF THE EXPERIMENTAL RACE CAR DATASETS. (* 0.99 QUANTILES)

ID	Vehicle	File names (Revs database)	u range (m/s)	v range* (m/s)	ω range (rad/s)	max* $\lvert a_y \rvert$ (m/s^2)	length (km)	duration (min)
C-1	Corvette Grand Sport	20130222_01_01_03 20130222_01_02_03 20130222_02_01_03	11.5 – 63.7	-1.03 – 1.3	-0.67 – 0.61	11.33	35.4	19.0
C-2	Corvette Grand Sport	20130223_01_01_03 20130223_01_02_03	11.1 – 61.2	-1.28 – 1.43	-0.67 – 0.63	11.60	24.1	12.6
F-1	Ferrari 250 LM GT	20140221_03_02_03 20140221_03_03_03 20140221_04_01_03	10.1 – 61.2	-1.66 – 1.80	-0.56 – 0.55	10.55	45.8	25.7
F-2	Ferrari 250 LM GT	20140222_01_01_03 20140222_02_01_03	7.6 – 65.4	-1.73 – 1.62	-0.59 – 0.59	10.78	33.6	18.1

VI. EXPERIMENTAL DATASET AND MODEL PARAMETERIZATION

A. Experimental Dataset

We use real telemetries of race cars from the Revs Vehicle Dynamics database (Stanford University), which is openly accessible [40]. It contains data from two cars —with challenging top speeds of 240 km/h and lateral accelerations up to 12 m/s^2. They are:

- Corvette Grand Sport 1963, hereafter named "Corvette";
- Ferrari 250 LM GT 1965, hereafter named "Ferrari".

We focus on the Palm Beach sessions listed in Table I. For each vehicle, the available data are divided into two subsets, named {C-1,C-2} for the Corvette and {F-1,F-2} for the Ferrari: the -1 and -2 subsets will be used for training and testing, as explained below. Table I shows the names of the corresponding files in the Revs database and some lap statistics.

To comply with the design sampling time of the neural estimators, the original signals, stored at 1 kHz, are down-sampled to $\tau^{-1} = 20$ Hz.

B. Training of the Estimators

We now describe how the datasets of Table I are used to train, validate and test the novel and benchmark neural estimators.

As mentioned in the introduction, observers can either be tailored to specific vehicles, or designed to be insensitive to variations in parameters such as operating conditions, vehicle types, and aging. Our objective is to develop observers with the latter characteristic, which can be achieved by designing and training neural networks with strong generalization abilities. This means that a network trained on data from one vehicle should be able to function effectively on another vehicle.

Networks with this level of generalization tend to be robust. Instead, networks specialized for a specific vehicle design may suffer from unpredictable degradation when vehicle parameters vary.

The benchmark Luenberger observer is constrained by design to use only kinematics. To a slightly lesser extent, the KS-NN networks introduced in this paper are also constrained to adhere to kinematics (Sections III and IV).

TABLE II

TRAINING, VALIDATION AND TEST SETS FOR THE NOVEL AND BENCHMARK NEURAL ESTIMATORS

ID	Training set	Validation set	Test set
1	Corvette C-1	Ferrari F-1	Corvette C-2 + Ferrari F-2
2	Ferrari F-1	Corvette C-1	Corvette C-2 + Ferrari F-2

However, the generic LSTM and GRU networks used for benchmarking are not ideal for our purposes. These networks are designed to fit the training data by identifying any hidden correlations in the input-output examples. With hundreds of parameters, these generic networks can easily learn characteristics that depend on fewer parameters, that differentiate one vehicle from another.

1) Controlling LSTM and GRU Overfitting: To force the benchmark GRU and LSTM networks to learn universal input-output relations as much as possible, we adopt the early stopping technique. The networks are trained with the training set of one vehicle and, as a validation set,[4] with the training set of the other vehicle. For example, the Corvette set C-1 for training, and the Ferrari set F-1 for validation, or vice versa, as shown in Table II. For convenience, this technique is referred here as *cross-vehicle regularization*. To have a uniform training procedure, the same training method with cross-vehicle regularization is adopted for the novel KS-NN networks as well.

With early stopping, the training process terminates as soon as the validation loss stops improving. This mitigates the risk of updating the network parameters in ways that are not general (at least not for the given validation examples), and yields the most robust network.

With early stopping, the trained network may vary with the initial parameter seeds (because of premature stopping).

[4] We recall that, in neural network terminology, the training set is used for updating the model parameters in gradient descent iterations, and the validation set is used for model selection.

TABLE III
MODELS HYPERPARAMETERS AND INSTANTIATIONS

Model name	N. variants	Hyperparameters			N. param.
		States	$\mathcal{P}$	$\mathcal{Q}$	
Benchmarks					
Luenberger	1	2	–	–	2
GRU	5	2-6	–	–	45-205
LSTM	5	2-6	–	–	59-271
Novel KS-NN observers					
Linear no bias	5	2	2-6	0	12-36
Linear + bias	5	2	2-6	0	16-48
Nonlinear no bias	3	2	2	1-3	20-36
Nonlinear + bias	15	2	2-6	1-3	32-168

Hence, we repeated the training several times with randomized initial guesses, and evaluated the distributions of the resulting quality metrics, taking the best performing networks of the random search.[5]

2) Training Procedure: The neural networks are developed and trained in Wolfram Mathematica 12.3.1, which uses the MXNet Deep Learning framework.

The training set for KS-NN consists of the input → output data $\{\boldsymbol{r}_{k-N+1}, \ldots, \boldsymbol{r}_{k-1}, \boldsymbol{r}_k\} \rightarrow v_{k+1}$, with v_{k+1} being the ground truth lateral speed measurements and $\boldsymbol{r}_k = \{\omega_k, u_{\mathrm{m}_k}, a_{y_k}, a_{x_k}\}$. The model estimates $\hat{v}_{k+1}$ using the provided windows (with size N) of past measurements $\boldsymbol{r}$, according to the recursion scheme of Fig. 1. Using supervised learning, the network is trained to minimize the mean square error (MSE) of $\hat{v}_{k+1} - v_{k+1}$. The Adam method [41] is employed for numerical optimization, with a maximum of 1000 training epochs (but early stopping typically occurs at about 200 epochs) and a small batch size of 64 (which helps generalization abilities).

C. Model Hyperparameters and Model Instantiations

Except for the benchmark Luenberger observer, all models have hyperparameters that control model complexity and the number of trainable parameters. For the benchmark LSTM and GRU networks, it is the dimension of the internal state that controls the model complexity. For the novel KS-NN networks of this article, the number of local models $\mathcal{P}$ and the number of hidden neurons in the feedback functions $\mathcal{Q}$ play a similar role.

Table III lists the models that were instantiated for each category. In total, there are 39 different candidates, with the number of parameters spanning the range 12-271 (except for the Luenberger, which has only two parameters).

VII. INVESTIGATION ON THE QUALITY OF THE DATA

A. Measurement Noise Analysis of the Experimental Vehicles

Equation (4a) states that $\dot{v}$ equals $a_y - \omega u_{\mathrm{m}}$. However, if we obtain $\dot{v}$ from the derivative of the ground truth lateral velocity v, it may not match $a_y - \omega u_{\mathrm{m}}$. This is because a_y, ω and u_{m} are noisy and contain measurement biases, such as errors in the mounting position of the accelerometer and gyroscope, chassis roll and pitch angles, and sensor offsets. Assuming that the ground truth lateral velocity v is more precise than the other signals, the quantity $\dot{v} - (a_y - \omega u_{\mathrm{m}})$ is an estimate of the overall noise on the right-hand side of (4a).

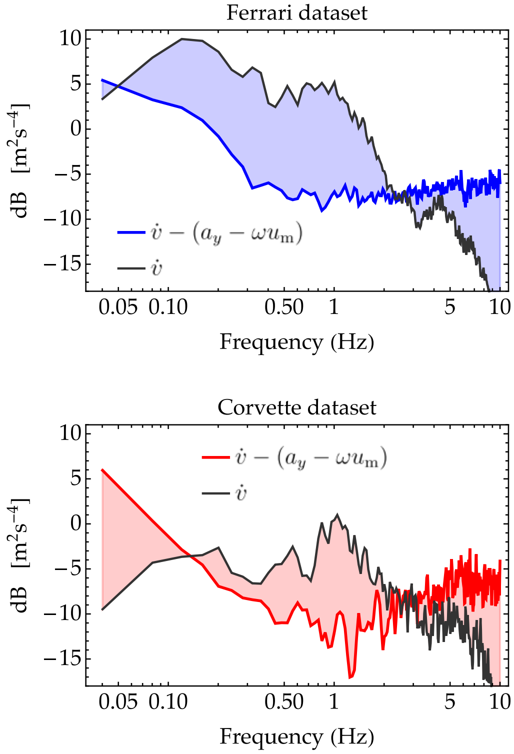


Fig. 4. The periodograms compare the spectral densities of the measured signal $\dot{v}$ and the process noise $\dot{v} - (a_y - \omega u_{\mathrm{m}})$, for the Ferrari (top) and the Corvette (bottom). They show that the noise-to-signal ratio is more favorable in the Ferrari than in the Corvette, especially at low frequencies.

Fig. 4 compares the periodograms [42] of $\dot{v} - (a_y - \omega u_{\mathrm{m}})$, for the Ferrari and the Corvette, to the periodograms of the measured $\dot{v}$, to visualize the signal-to-noise ratios, shown with shaded areas. The signal-to-noise ratio is more favorable in the Ferrari, where the power of $\dot{v}$ is relatively high compared to the power of $\dot{v} - (a_y - \omega u_{\mathrm{m}})$. In the Corvette, the power of $\dot{v}$ is lower, resulting in a higher noise-to-signal ratio. In particular, in the Corvette, below 0.1 Hz, the signal $\dot{v}$ is below the noise level.

1) Mean Bias: The mean value of $\dot{v} - (a_y - \omega u_{\mathrm{m}})$ is 0.10 m/s^2 for the Corvette, and a better -0.03 m/s^2 for the Ferrari. We can interpret this figure as a mean bias that originates somewhere on the right-hand side of (4a), which is the primary signal to estimate v kinematically.

B. Lateral Velocity Dynamics of the Experimental Vehicles

Fig. 5 shows that the measured lateral velocity v and lateral acceleration a_y exhibit different dynamics for the Ferrari and the Corvette. The plot suggests that the Ferrari achieves higher lateral velocities for the same lateral acceleration, compared to the Corvette. These differences emphasize the importance of testing the observer's ability to generalize and potential overfitting issues, which is achieved by testing the observer on the vehicle not used for training.

[5] In one case, an outlier was removed.

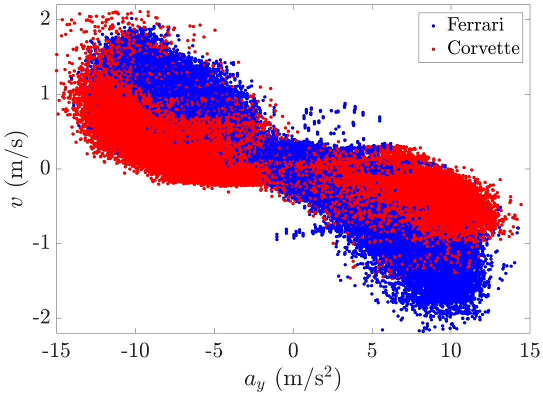


Fig. 5. The plot shows the measured lateral velocity v versus the lateral acceleration a_y, for the Ferrari and the Corvette. It can be observed that the Ferrari achieves larger absolute values of v than the Corvette for the same a_y, indicating different dynamic characteristics between the two cars. These differences support the robustness analyses presented in the paper, which aim to evaluate the observer's ability to generalize by testing it on a car that was not used for training.

VIII. Results and Discussion

Fig. 6 gives a visual representation of the performance of the 39 candidate models in Table III.

The left column concerns models trained according to scheme 1 of Table II, i.e., with the Corvette set C-1 for training, and the Ferrari set F-1 for validation. The right column concerns models trained with scheme 2, i.e., using F-1 for training, and C-1 for validation. The performance metrics are evaluated on both test sets C-2 and F-2, and are the coefficient of determination R^2 in the top row and the root mean square error (RMSE) of the predicted lateral velocity in the bottom row.

More precisely, $R^2 = 1 - \text{var}(v - \hat{v})/\text{var}(v)$ —with $\hat{v}$ the estimated lateral velocity, v the ground truth lateral velocity, and var$(\cdot)$ the variance operator— is a non-dimensional index showing how much of the variation of v is successfully accounted for by a model. Its complement to 1, $\text{FVU} = 1 - R^2 = \text{var}(v - \hat{v})/\text{var}(v)$, is known as the fraction of variance unexplained by a model.

The root mean square error of the lateral velocity, on the bottom charts, complements the above information with the expected magnitude of the estimated error: $\text{RMSE} = \sqrt{E[(v - \hat{v})^2]}$.

A. Model Quality in Terms of Accounted Variance

Let us first study the top row of Fig. 6. The dots show the coefficient of determination R^2 of the 39 models. The benchmark Luenberger model is shown in gray. The benchmark LSTM and GRU models have similar behavior and are shown together in purple. The KS-NN models with linear feedback ($Q = 0$) are colored blue. They form two clusters: models without biases ($b_{1_i} = 0$ in (10a)), and models with biases ($b_{1_i} \neq 0$ in (10a)). The KS-NN models with nonlinear feedback ($Q > 0$) are colored in orange. They also form two clusters: without biases ($b_{2_{ji}} = 0$ in (8)) and with biases ($b_{2_{ji}} \neq 0$ in (8)).

1) Training Scheme 1: Let us now focus on training scheme 1 (Corvette C-1 as training set and Ferrari F-1 as validation set), shown in the top left sub-plot of Fig. 6. By training on C-1 with F-1 as the validation set, the models seek to optimize the performance of the Corvette, subject to not worsening the performance of the Ferrari. So, they seek to best model one vehicle while maintaining the maximum generalization capacity measured on the other. The graph plots R^2, evaluated in test sets C-2 (Corvette) and F-2 (Ferrari), which were not used for training or selection. C-2 is on the y-axis.

All dots are above the main diagonal, meaning that the performance on C-2 (the Corvette test set) is better than the performance on F-2 (the Ferrari test set). This is expected, since we trained with Corvette data, and used Ferrari data only to stop the training process. The difference between C-2 and F-2, i.e., the distance from the main diagonal, can be seen as a measure of model specialization for one vehicle, and thus, the generalization loss.

The benchmark Luenberger observer performs similarly in C-2 and F-2. However, it only accounts for 76.5% and 74.7% of the variance of v, respectively, indicating that about 25% of the variance is not predicted by the Luenberger.

The benchmark LSTM and GRU models outperform the Luenberger on C-2, exceeding 90% of explained variance. However, their performance worsens on F-2, indicating specialization for one vehicle at the expense of the other.

The KS-NN linear and nonlinear models with biases perform even better than the benchmarks in the C-2 test set, and marginally better than the Luenberger observer in F-2. This means that they are able to improve both vehicles, with the greatest gains for the vehicle that provided the training set.

The KS-NN models without biases are the most robust, as they remain close to the main diagonal and exhibit equal improvements in both vehicles. The cost of such a greater generalization capacity is a decrease of R^2 to 86.8% in C-2 (from 92.9% of the models with biases), which, however, is balanced with an increase to 85.5% in F-2 (from 75.8% of the models with biases). We argue that models with biases (and, among others, the benchmarks GRU and LSTM) learn to compensate for the measurement biases discussed in Section VII. However, since the measurement biases vary in time and across different sensors and cars, specializing for one vehicle reduces performance for others with different biases.

2) Training Scheme 2: The above considerations hold also for training scheme 2, shown in the top right sub-plot of Fig. 6, in which the Ferrari F-1 set is used for training, and the Corvette C-1 set for validation. However, the signal-to-noise ratio of the Ferrari is better than that of the Corvette (Section VII-A). Therefore, while performance in the Ferrari test set F-2 (y-axis) tends to improve, it comes at the cost of reduced generalization capacity for most models, except for the KS-NN nonlinear model without biases, which is the most robust.

In the top right chart, the most general models and the most specialized models are indicated.

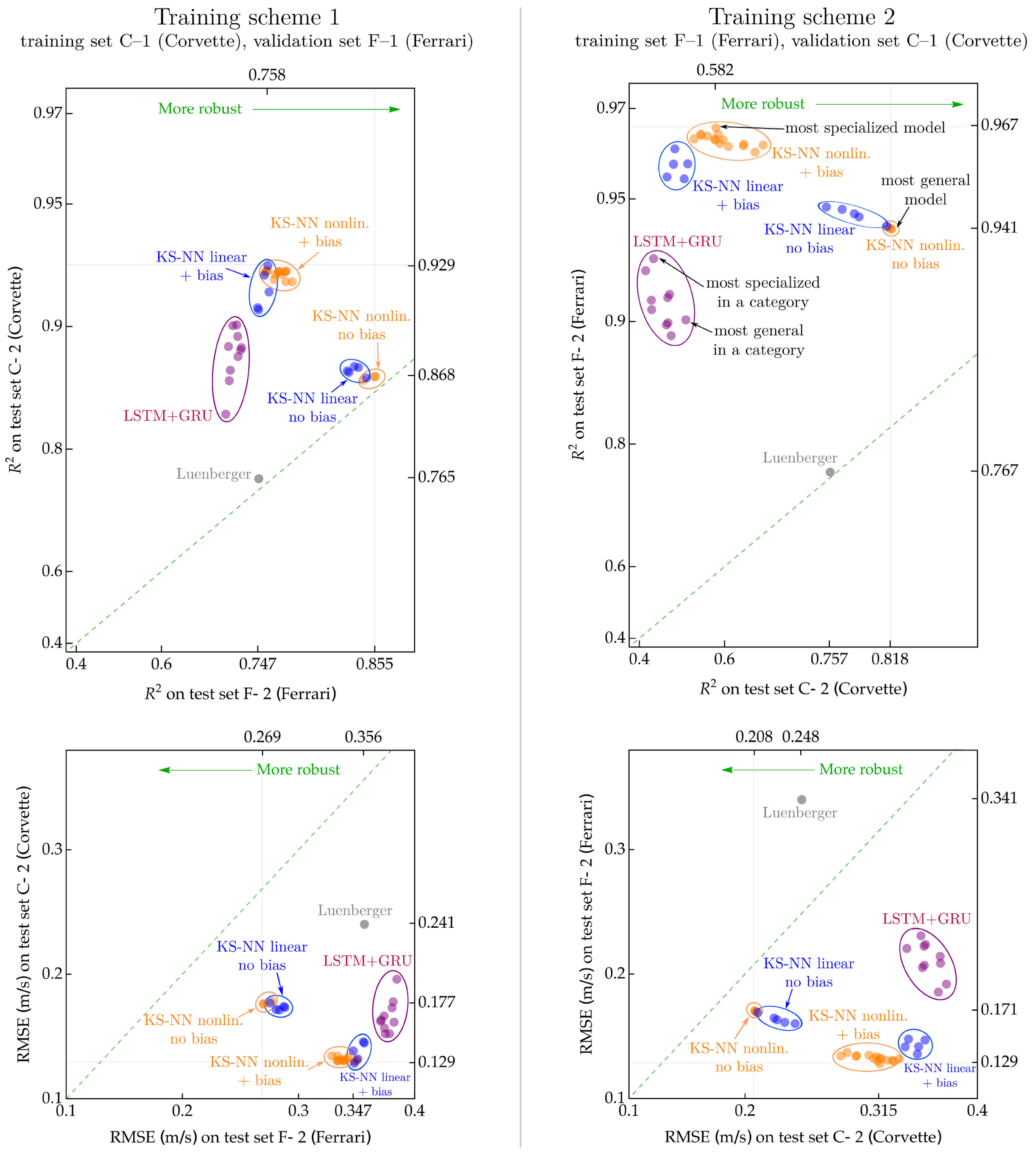


Fig. 6. The charts illustrate the performance of the candidate observers listed in Table III. The top row shows the explained variance R^2 on test datasets of the Ferrari (F-2) and the Corvette (C-2), while the bottom row shows the root mean square error (RMSE) on the same datasets. The first column corresponds to training scheme 1 (Table II), in which the C-1 (Corvette) dataset is used for training and the F-1 (Ferrari) is employed for early stopping. The second column is for the opposite training scheme 2. The "KS-NN nonlinear no bias" model exhibits the best generalization capacity, outperforming all other models on the vehicle not used for training. Compared to the benchmark Luenberger, the increase in R^2 is 25-43%, while the improvement over the benchmarks GRU and LSTM is 47-65%.

B. Model Quality in Terms of Expected Error

The bottom row of Fig. 6 displays the root mean square error (RMSE) of the predicted velocity. The KS-NN nonlinear model without biases exhibits the best generalization capacity: compared to the benchmark Luenberger, it improves the RMSE by 16-24% on the vehicle not used for training; compared to the benchmarks GRU and LSTM, it improves by 27-38%.

C. Models With the Best Generalization Capacity

Tables IV and V list the most general models per category. The general models significantly improve the observer's ability to predict the velocity of the testing vehicle, with only a minor loss in accuracy for the training vehicle. The model with the best generalization capacity is highlighted: it is the KS-NN network, with nonlinear feedback functions and no biases. Compared to the benchmark Luenberger, it improves the FVU

TABLE IV

MOST GENERAL MODELS PER CATEGORY, TRAINING SCHEME 1: TRAINING ON CORVETTE, VALIDATION ON FERRARI, TEST ON CORVETTE (C-2) AND FERRARI (F-2). THE KS-NN NONLINEAR MODEL WITHOUT BIASES IS THE MOST ACCURATE ON F-2, I.E., HAS THE BEST GENERALIZATION CAPACITY

Model	States	$\mathcal{P}$	$\mathcal{Q}$	N. pars.	C-2 FVU	C-2 RMSE (m/s)	F-2 FVU	F-2 RMSE (m/s)
Luenberger (benchmark)	2	–	–	2	0.235	0.241	0.253	0.356
GRU (benchmark)	5	–	–	156	0.118	0.167	0.279	0.373
LSTM (benchmark)	5	–	–	206	0.112	0.163	0.275	0.370
KS-NN linear + bias	2	2	0	16	0.082	0.139	0.241	0.346
KS-NN nonlinear + bias	2	2	3	56	0.077	0.135	0.216	0.328
KS-NN linear no bias	2	2	0	12	0.133	0.178	0.152	0.275
KS-NN nonlinear no bias (best generalization)	2	2	3	36	0.132	0.177	**0.145**	0.269

TABLE V

MOST GENERAL MODELS PER CATEGORY, TRAINING SCHEME 2: TRAINING ON FERRARI, VALIDATION ON CORVETTE, TEST ON CORVETTE (C-2) AND FERRARI (F-2). THE KS-NN NONLINEAR MODEL WITHOUT BIASES IS THE MOST ACCURATE ON C-2, I.E., HAS THE BEST GENERALIZATION CAPACITY

Model	States	$\mathcal{P}$	$\mathcal{Q}$	N. pars.	C-2 FVU	C-2 RMSE (m/s)	F-2 FVU	F-2 RMSE (m/s)
Luenberger (benchmark)	2	–	–	2.	0.243	0.248	0.233	0.341
GRU (benchmark)	2	–	–	45.	0.518	0.351	0.108	0.232
LSTM (benchmark)	2	–	–	59.	0.483	0.338	0.098	0.221
KS-NN linear + bias	2	3	0	24.	0.479	0.337	0.041	0.142
KS-NN nonlinear + bias	2	2	2	44.	0.334	0.282	0.037	0.135
KS-NN linear no bias	2	2	0	12.	0.186	0.210	0.058	0.170
KS-NN nonlinear no bias (best generalization)	2	2	2	28.	**0.182**	0.208	0.059	0.171

by 25-43% on the test vehicle; compared to the benchmarks GRU and LSTM, it improves by 47-65%.

From Tables IV and V, one may observe that robust solutions tend to have few parameters.

The nonlinear feedback functions $\{f_1(\cdot), f_2(\cdot)\}$ (8) in the best general models realize variable feedback gains, as already found in [18].

D. Comparison With the Factor Graph Benchmark [12]

This section presents a comparison between the KS-NN network and the factor graph (FG) and Kalman filter (KF) estimators proposed by [12]. The comparison is based on the same Ferrari telemetries used in the open race car dataset [40] presented in this paper. To directly compare our results with [12], we compute the RMSE of the side slip angle estimate ($\beta = \mathrm{atan}(v/u)$).

Table VI shows the results obtained with our KS-NN,[6] the benchmarks Luenberger, GRU, LSTM, and the FG and KF of [12, Section 4].

Our KS-NN model shows superior performance compared to the FG and KF benchmarks, with an RMSE of 0.40°, which is 30% and 54% lower, respectively. It is worth noting that the FG and KF benchmarks were not cross-evaluated on the Corvette in [12]. On the other hand, as shown in Table V, our KS-NN also exhibits good generalization capacity on the Corvette dataset, which was not used for training.

TABLE VI

RMSE OF THE SIDESLIP ANGLE (β) ESTIMATE ON THE FERRARI F-2 TEST SET: THE KS-NN MODEL OUTPERFORMS THE BENCHMARKS, INCLUDING THE FG AND KF OF [12]

Model	RMSE (°)
Factor graph (FG) of [12] (benchmark)	0.57
Kalman filter (KF) of [12] (benchmark)	0.87
Luenberger (benchmark)	0.73
GRU (benchmark)	0.57
LSTM (benchmark)	0.52
KS-NN nonlinear no bias	**0.40**

E. Noise Sensitivities

Assuming that most of the noise is contained in the a_y and a_x signals, one can study the propagation of noise in the KS-NN linear networks and in the benchmark Luenberger observer [11] as follows.

Starting from the characteristic equations of each model ((7a,7b) for KS-NN), the quantities $\{a_{y_k}, a_{x_k}\}$ are replaced with $\{a_{y_k} + \delta a_{y_k}, a_{x_k} + \delta a_{x_k}\}$, where $\{\delta a_{y_k}, \delta a_{x_k}\}$ stand for the acceleration noise. Applying the $\mathcal{Z}$ transform, the variation $\delta\hat{v}(z)$ for given $\{\delta a_y(z), \delta a_x(z)\}$ is:

$$\delta\hat{v}(z) = H_{Y_\mathrm{M}}(z)\delta a_y(z) + H_{X_\mathrm{M}}(z)\delta a_x(z) \qquad (16)$$

with $H_{Y_\mathrm{M}}(z)$ and $H_{X_\mathrm{M}}(z)$ being the noise sensitivities, and M={KS-NN,Luenberger} indicating the model type. The expressions of $\{H_{Y_\mathrm{M}}(z), H_{X_\mathrm{M}}(z)\}$ for KS-NN and the benchmark Luenberger are reported in Appendix B.

$H_{Y_\mathrm{M}}(z)$ and $H_{X_\mathrm{M}}(z)$ (Appendix B) depend on the yaw rate ω. Figure 7 compares the sensitivities of the KS-NN model and the Luenberger benchmark for two different values of ω: 0.2 rad/s (which is the median value of $|\omega|$ for both the Corvette and the Ferrari) and 0.4 rad/s. The plot shows that the KS-NN model has significantly lower sensitivities compared to the Luenberger, for both yaw rate values. The sensitivities

[6] We use the bias-free nonlinear variant of our KS-NN, which has the best generalization capacity (Table V).

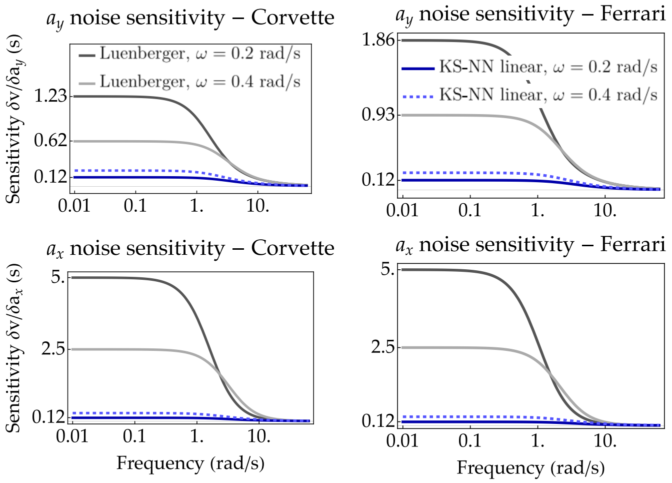


Fig. 7. Magnitudes of the transfer function sensitivities to acceleration noise, for the KS-NN model and the benchmark Luenberger, trained with the Corvette and the Ferrari training sets. The left column corresponds to the Corvette, while the right column corresponds to the Ferrari. For both vehicles, the KS-NN model has significantly lower sensitivities than the Luenberger benchmark, indicating that the KS-NN is more robust to acceleration noise. Note that the sensitivities to a_y and a_x are not equal, due to the different levels of noise in the signals. During training, the sensitivities are balanced to maximize the estimation accuracy.

to lateral acceleration (a_y) and longitudinal acceleration (a_x) for the two vehicles are not equal, as they have different levels of noise in their respective signals. During training, the sensitivities are adjusted to balance the trade-off between accuracy in estimating the states and robustness to noise.

F. Power Spectral Densities of the Residuals

Figure 8 displays the power spectral densities (PSDs) [42] of the lateral speed estimation errors, obtained by the KS-NN network and the benchmarks Luenberger and LSTM. The top plot illustrates the results of the Ferrari test set, using the most robust models trained with the Corvette, while the bottom plot depicts the results of the Corvette test set, using the most robust models trained with the Ferrari.

The PSD of the measured lateral velocity signal is plotted as well (in black), to appreciate the signal-to-noise ratios.

The benchmark Luenberger residuals have a higher PSD than the KS-NN model, across most frequencies. Beyond 1-2 Hz, the PSD of the Luenberger exceeds even the signal level, indicating noise introduced by the observer.

The benchmark LSTM residuals have greater power than the KS-NN network, with a 7 dB difference at 0.1 Hz and 8-10 dB difference at 1 Hz, on the Corvette. In contrast, the PSD of the residuals of the KS-NN model is consistently below the signal level, and has the lowest values for all frequencies in both cars, confirming its improved estimation robustness.

G. Computational Efficiency of Implementations

As noted, the KS-NN estimator is made of parallel models, whose outputs are weighted by activation functions. The discrete-time equations of a module (7a,7b), feedback function (8), and the activation functions ($\phi_p(|\omega|)$) are not difficult to code. Weights can be obtained from the trained network. The total number of mathematical operations is small and easily supported by low-cost automotive-grade hardware.

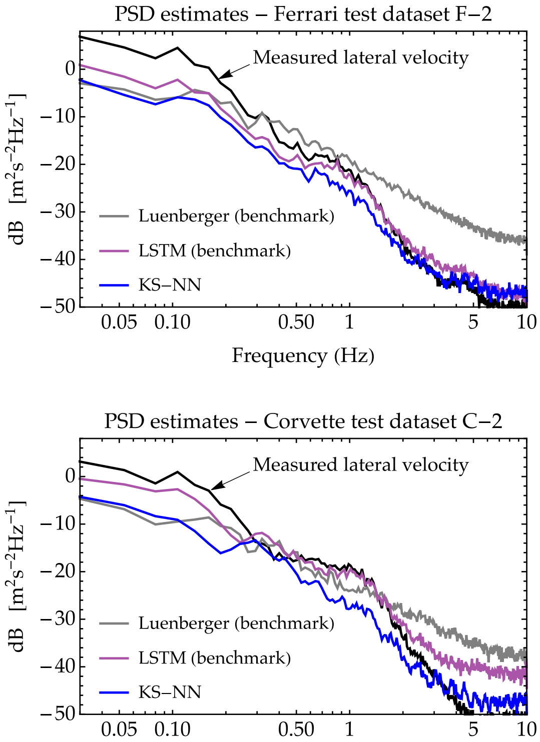


Fig. 8. This figure compares the power spectral densities of the v estimation errors, for the KS-NN network and the benchmarks Luenberger and LSTM. The top and bottom plots present results on the Ferrari and Corvette test sets, respectively, using the most robust models trained on the opposite car (Tables IV and V). The KS-NN network outperforms the benchmarks, as its residuals have the lowest power.

H. What Happens Without Cross-Vehicle Regularization

In Section VI-B.1, the cross-vehicle regularization technique was introduced to prevent overfitting of the training data, using early stopping with a validation set of another vehicle. In this section, we investigate the effects of removing cross-vehicle regularization.

Let us consider the findings in Table V, and focus on the benchmark LSTM model and the novel "KS-NN nonlinear no bias (best generalization)." Let us re-train these two architectures (LSTM with 2 states and KS-NN with $\mathcal{P} = 2$ and $\mathcal{Q} = 2$) as follows. Instead of using the Ferrari F-1 set for training and the Corvette C-1 set for validation, let us use the first two files of F-1 for training and the last file of F-1 for validation. This means that we train on Ferrari data, and the validation set is from the same vehicle. Using both training and validation sets from the same vehicle permits exploiting characteristics that are specific to that vehicle: on the one hand, the trained network improves specialization, but on the other hand, it loses generality. We show that the loss of generality is negligible for KS-NN, but is severe for LSTM.

The results are shown in Table VII. When cross-vehicle regularization is *not* used, the benchmark LSTM performs better on the Ferrari F-2 test set, with an FVU of 0.042,

TABLE VII

Models Trained With and Without Cross-Vehicle Regularization (Training Scheme 2). In the Absence of Regularization, the Benchmark LSTM Considerably Worsens the Accuracy on the Corvette (C-2), Not Used for Training. In Contrast, the KS-NN Model Preserves Good Generalization Capacity

Model	States	$\mathcal{P}$	$\mathcal{Q}$	N. pars.	C-2 FVU	C-2 RMSE (m/s)	F-2 FVU	F-2 RMSE (m/s)
LSTM (with regularization, same of Table V)	2	–	–	59.	0.483	0.338	0.098	0.221
LSTM (without regularization)	2	–	–	59.	0.840	0.446	0.042	0.144
KS-NN nonlinear no bias (same of Table V)	2	2	2	28.	0.182	0.208	0.059	0.171
KS-NN nonlinear no bias (without regularization)	2	2	2	28.	0.197	0.216	0.059	0.171

which is more than twice as good as the regularized version's FVU of 0.098. However, in the absence of regularization, the LSTM model fails to accurately predict the sideslip on the Corvette (C-2) test set, with an FVU of 0.84. In Fig. 6, the point of this case would fall outside the left edge of the charts (and still below the most specialized KS-NN model with nonlinear feedback, at $R^2 = 0.967$). In contrast, the KS-NN model maintains its performance on the Ferrari F-2 test set, and exhibits only a slight decline in accuracy on the Corvette C-2 test set.

IX. Conclusion and Future Work

This paper introduces novel kinematics-structured neural network (KS-NN) models for lateral speed estimation in vehicles. The KS-NN model is designed to incorporate and generalize the underlying kinematic laws.

The kinematics-driven internal structure of the KS-NN models makes them physically explainable and enhances their generalization capacity. This reduces the risk of the models learning vehicle- and environment-specific characteristics, as they are designed to embed and generalize the underlying kinematic laws.

We propose a physical interpretation of the neural network's internal structure, explaining the role of its learnable parameters, and discussing the advantages over existing unexplainable neural estimators.

We evaluate KS-NN against existing Luenberger [11], GRU and LSTM neural networks (similar to [9] and [10]), and factor graph [12] estimators, using an openly accessible dataset with the telemetries of two race cars. The cars have significantly different lateral velocity dynamics, noise levels, and measurement biases.

The KS-NN variants that exhibit the highest generalization capacity are those with nonlinear feedback functions and no biases. When tested on a vehicle that was not used during training, these models outperform the Luenberger observer by 25-43% in terms of explained variance R^2, as well as the GRU and LSTM networks by 47-65%. Furthermore, KS-NN outperforms the factor graph proposed in [12] by 30%, in terms of RMSE. Although the best linear variants of KS-NN have slightly lower accuracy (up to 2% RMSE) than their nonlinear counterparts, they are simpler and physically explainable LPV systems with only 12 parameters. Despite their simplicity, they still exhibit superior generalization performance compared to all the benchmarks.

The KS-NN models have low computational complexity and require only few parameters, with the best nonlinear variant having up to 36 parameters and the linear variant having only 12. They rely on standard odometers, gyroscopes, and bi-axial accelerometers to measure forward speed, yaw rate, and longitudinal/lateral accelerations. Unlike other recurrent neural network estimators, which may require specialized hardware for real-time operation, the KS-NN models can be easily deployed using standard sensors and automotive-grade hardware.

A. Limitations and Future Work

This study used an open dataset with the telemetries of two different race cars, indicating how well the models may generalize. Future work might aim to study a broader set of experimental cars.

In this study, roll and pitch biases were not compensated for, as using a 6-degree-of-freedom inertial measurement unit for commercial vehicles is not affordable. Hence, estimators that do not compensate for acceleration biases were studied. It was found that models that can learn biases perform better on the training vehicle, but perform worse on others, as different vehicles have different biases. Therefore, it is expected that removing biases from the measurements before feeding the networks, e.g. via bias estimation and removal, would achieve better performance without sacrificing generalization capacity.

Patent

The method described in this paper is patented (PCT/IT2023/050058).

Appendix A
Benchmark Luenberger Observer

This section discusses the implementation of the kinematic Luenberger observer of [11], which is used as a benchmark. The observer uses the basic kinematic estimation model (5a,5b), where the feedback functions $\bar{f}_1(\cdot)$ and $\bar{f}_2(\cdot)$ are designed as:

$$\begin{bmatrix} \bar{f}_1(\omega, \hat{u} - u_\mathrm{m}) \\ \bar{f}_2(\omega, \hat{u} - u_\mathrm{m}) \end{bmatrix} = \boldsymbol{L}(\omega) \cdot (\hat{u} - u_\mathrm{m}) = \begin{bmatrix} 2\alpha|\omega| \\ (\alpha^2 - 1)\omega \end{bmatrix} (\hat{u} - u_\mathrm{m}) \tag{17}$$

Following [11], the feedback gain $\boldsymbol{L}(\omega)$ in (17) is designed to place both the eigenvalues of the resulting state matrix in $-\alpha|\omega|$, where $\alpha \in \mathbb{R}_{>0}$ is a design parameter. Large values of α produce faster estimation convergence dynamics, at the price of a higher sensitivity to the measurement noise.

$$\begin{cases} H_{Y_{\text{KS}-\text{NN}}}(z) = \dfrac{(\beta_1 z + \beta_1 \mathrm{w}_{1_2} - \beta_1 \gamma_2)\tau}{z^2 + (\mathrm{w}_{1_2} - \gamma_1 - \gamma_2)z + \beta_2 \omega^2 \tau \mathrm{w}_{1_1} - \mathrm{w}_{1_2}\gamma_1 + \gamma_2\gamma_1} \\ H_{X_{\text{KS}-\text{NN}}}(z) = \dfrac{-\omega \mathrm{w}_{1_1} \beta_2 \tau}{z^2 + (\mathrm{w}_{1_2} - \gamma_1 - \gamma_2)z + \beta_2 \omega^2 \tau \mathrm{w}_{1_1} - \mathrm{w}_{1_2}\gamma_1 + \gamma_2\gamma_1} \end{cases} \tag{18}$$

$$\begin{cases} H_{Y_{\text{Luenberger}}}(z) = \dfrac{(-1 + (2|\omega|\alpha T_s + 1)z)T_s z}{(\alpha^2\omega^2 T_s^2 + 2|\omega|\alpha T_s + 1)z^2 + (-2|\omega|\alpha T_s - 2)z + 1} \\ H_{X_{\text{Luenberger}}}(z) = -\dfrac{\alpha^2 \omega T_s^2 z^2}{(\alpha^2\omega^2 T_s^2 + 2|\omega|\alpha T_s + 1)z^2 + (-2|\omega|\alpha T_s - 2)z + 1} \end{cases} \tag{19}$$

We discretize the observer dynamics (5a,17) using the implicit Euler method, with a fixed time step $T_s = 0.01$ s.

One of the main issues with the basic kinematic estimation model (5a,5b) is that the lateral velocity becomes unobservable for low yaw rates ω (Section II-B). On this account, the observer state $\{\hat{v}, \hat{u}\}$ is reset when $|\omega|$ is below a tunable threshold ω_{th}: $\hat{v}$ is reset to 0, while $\hat{u}$ is reset to u_{m}. In this way, the tunable parameters for the Luenberger observer are two: $\{\alpha, \omega_{\text{th}}\}$.

A Lyapunov function is given in [11], proving the asymptotic stability of the estimation error dynamics (5a,17), if $\omega \neq 0$ and in the absence of measurement noise. The switching dynamics of the observer with reset inherit the asymptotic stability properties of (5a,17) if $|\omega| \geq \omega_{\text{th}}$. For the case $|\omega| \leq \omega_{\text{th}}$, the estimation error remains bounded, but it cannot be proved to converge to zero.

The design parameters $\{\alpha, \omega_{\text{th}}\}$ are optimized by minimizing the mean square estimation error obtained with the Luenberger observer dynamics (with reset). For the optimization, we adopt the direct search Nelder-Mead method [43], [44], since the use of gradient-based techniques is hindered by the discontinuous switching dynamics. Box constraints[7] are imposed for $\{\alpha, \omega_{\text{th}}\}$ to improve the stability of the optimization algorithm [45]. In order to reduce the risk of finding local minima, the optimization is repeated with a 4×4 grid of initial guesses for $\{\alpha, \omega_{\text{th}}\}$.[8]

Appendix B
Expressions of the Noise Sensitivities

The transfer functions $H_{Y_\text{M}}(z)$ and $H_{X_\text{M}}(z)$, M={KS-NN, Luenberger}, providing the noise sensitivities for the novel KS-NN linear networks and the benchmark Luenberger [11] are (18) and (19), shown at the top of the page.

Note that the transfer functions (18) are derived from the infinite impulse response (IIR) filter that approximates the FIR recurrence.

References

[1] A. T. van Zanten, "Bosch ESP systems: 5 years of experience," *J. Passenger Cars, Electron. Elect. Syst.*, vol. 109, pp. 428–436, May 2000.

[2] M. Canale and L. Fagiano, "Vehicle yaw control using a fast NMPC approach," in *Proc. 47th IEEE Conf. Decis. Control*, Dec. 2008, pp. 5360–5365.

[3] G. Panzani et al., "Designing on-demand four-wheel-drive vehicles via active control of the central transfer case," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 4, pp. 931–941, Dec. 2010.

[4] R. Rajamani, *Vehicle Dynamics and Control* (Mechanical Engineering Series). New York, NY, USA: Springer, 2011.

[5] C. Hu, R. Wang, F. Yan, and N. Chen, "Should the desired heading in path following of autonomous vehicles be the tangent direction of the desired path?" *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3084–3094, Dec. 2015.

[6] M. Piccinini, M. Larcher, E. Pagot, D. Piscini, L. Pasquato, and F. Biral, "A predictive neural hierarchical framework for on-line time-optimal motion planning and control of black-box vehicle models," *Vehicle Syst. Dyn.*, vol. 61, no. 1, pp. 83–110, Jan. 2023.

[7] M. Piccinini, S. Taddei, M. Larcher, M. Piazza, and F. Biral, "A physics-driven artificial agent for online time-optimal vehicle motion planning and control," *IEEE Access*, vol. 11, pp. 46344–46372, 2023.

[8] E. Pagot, M. Piccinini, and F. Biral, "Real-time optimal control of an autonomous RC car with minimum-time maneuvers and a novel kineto-dynamical model," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 2390–2396.

[9] T. Gräber, S. Lupberger, M. Unterreiner, and D. Schramm, "A hybrid approach to side-slip angle estimation with recurrent neural networks and kinematic vehicle models," *IEEE Trans. Intell. Vehicles*, vol. 4, no. 1, pp. 39–47, Mar. 2019.

[10] D. Kong et al., "Vehicle lateral velocity estimation based on long short-term memory network," *World Electr. Vehicle J.*, vol. 13, no. 1, p. 1, Dec. 2021.

[11] J. Farrelly and P. Wellstead, "Estimation of vehicle lateral velocity," in *Proc. IEEE Int. Conf. Control Appl., IEEE Int. Conf. Control Appl., IEEE Int. Symp. Intell. Contro*, Nov. 1996, pp. 552–557.

[12] A. Leanza, G. Reina, and J.-L. Blanco-Claraco, "A factor-graph-based approach to vehicle sideslip angle estimation," *Sensors*, vol. 21, no. 16, p. 5409, Aug. 2021.

[13] D. Chindamo, B. Lenzo, and M. Gadola, "On the vehicle sideslip angle estimation: A literature review of methods, models, and innovations," *Appl. Sci.*, vol. 8, no. 3, p. 355, Mar. 2018.

[14] M. Gadola, D. Chindamo, M. Romano, and F. Padula, "Development and validation of a Kalman filter-based model for vehicle slip angle estimation," *Vehicle Syst. Dyn.*, vol. 52, no. 1, pp. 68–84, Jan. 2014.

[15] J. L. Coyte, B. Li, H. Du, W. Li, D. Stirling, and M. Ros, "Decision tree assisted EKF for vehicle slip angle estimation using inertial motion sensors," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2014, pp. 940–946.

[16] G. Baffet, A. Charara, and D. Lechner, "Estimation of vehicle sideslip, tire force and wheel cornering stiffness," *Control Eng. Pract.*, vol. 17, no. 11, pp. 1255–1264, Nov. 2009.

[17] J. Stephant, A. Charara, and D. Meizel, "Virtual sensor: Application to vehicle sideslip angle and transversal forces," *IEEE Trans. Ind. Electron.*, vol. 51, no. 2, pp. 278–289, Apr. 2004.

[18] L. De Pascali, F. Biral, M. Cocetti, L. Zaccarian, and S. Tarbouriech, "A kinematic observer with adaptive dead-zone for vehicles lateral velocity estimation," in *Proc. IEEE 15th Int. Workshop Adv. Motion Control (AMC)*, Mar. 2018, pp. 511–516.

[19] D. Selmanaj, M. Corno, G. Panzani, and S. M. Savaresi, "Vehicle sideslip estimation: A kinematic based approach," *Control Eng. Pract.*, vol. 67, pp. 1–12, Oct. 2017.

[20] G. Panzani, M. Corno, M. Tanelli, S. M. Savaresi, A. Fortina, and S. Campo, "Control-oriented vehicle attitude estimation with online sensors bias compensation," in *Proc. ASME Dyn. Syst. Control Conf.*, vol. 1, Jan. 2009, pp. 819–826.

[7] The gain α is constrained in the range $[0, 50]$, while the admissible range for ω_{th} is $[0, 0.5 \cdot \omega_{\max}]$, with $\omega_{\max}$ being the maximum recorded value of $|\omega|$ for a given dataset.

[8] The initial guesses for α are chosen in the range $[3, 20]$, while the initial conditions for ω_{th} are selected in $[0, 0.3 \cdot \omega_{\max}]$.

[21] J. Oh, Y. Noh, and S. B. Choi, "Real-time offset error compensation of 6D IMU mounted on ground vehicles using disturbance observer," *J. Adv. Comput. Netw.*, vol. 1, no. 2, pp. 82–87, 2013.

[22] M. Viehweger et al., "Vehicle state and tyre force estimation: Demonstrations and guidelines," *Vehicle Syst. Dyn.*, vol. 59, no. 5, pp. 675–702, May 2021.

[23] A. G. Alatorre, A. Charara, and A. Victorino, "Sideslip estimation algorithm comparison between Euler angles and quaternion approaches with black box vehicle model," in *Proc. IEEE 15th Int. Workshop Adv. Motion Control (AMC)*, Mar. 2018, pp. 553–559.

[24] J. Ryu and J. C. Gerdes, "Integrating inertial sensors with global positioning system (GPS) for vehicle dynamics control," *J. Dyn. Syst., Meas., Control*, vol. 126, no. 2, pp. 243–254, Jun. 2004.

[25] O. Galluppi, M. Corno, and S. M. Savaresi, "Mixed-kinematic body sideslip angle estimator for high performance cars," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2018, pp. 941–946.

[26] H. F. Grip, L. Imsland, T. A. Johansen, J. C. Kalkkuhl, and A. Suissa, "Vehicle sideslip estimation," *IEEE Control Syst. Mag.*, vol. 29, no. 5, pp. 36–52, Oct. 2009.

[27] H. F. Grip, L. Imsland, T. A. Johansen, T. I. Fossen, J. C. Kalkkuhl, and A. Suissa, "Nonlinear vehicle side-slip estimation with friction adaptation," *Automatica*, vol. 44, no. 3, pp. 611–622, Mar. 2008.

[28] A. Y. Ungoren, H. Peng, and H. E. Tseng, "A study on lateral speed estimation methods," *Int. J. Vehicle Auto. Syst.*, vol. 2, nos. 1–2, pp. 126–144, 2004.

[29] F. Cheli, E. Sabbioni, M. Pesce, and S. Melzi, "A methodology for vehicle sideslip angle identification: Comparison with experimental data," *Vehicle Syst. Dyn.*, vol. 45, no. 6, pp. 549–563, Jun. 2007.

[30] S. Srinivasan, I. Sa, A. Zyner, V. Reijgwart, M. I. Valls, and R. Siegwart, "End-to-end velocity estimation for autonomous racing," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6869–6875, Oct. 2020.

[31] D. Chindamo and M. Gadola, "Estimation of vehicle side-slip angle using an artificial neural network," *MATEC Web Conf.*, vol. 166, Jan. 2018, Art. no. 02001.

[32] M. Acosta and S. Kanarachos, "Tire lateral force estimation and grip potential identification using Neural Networks, Extended Kalman Filter, and Recursive Least Squares," *Neural Comput. Appl.*, vol. 30, no. 11, pp. 3445–3465, Dec. 2018.

[33] M. Milanese, D. Regruto, and A. Fortina, "Direct virtual sensor (DVS) design in vehicle sideslip angle estimation," in *Proc. Amer. Control Conf.*, Jul. 2007, pp. 3654–3658.

[34] A. Bonfitto, S. Feraco, A. Tonoli, and N. Amati, "Combined regression and classification artificial neural networks for sideslip angle estimation and road condition identification," *Vehicle Syst. Dyn.*, vol. 58, no. 11, pp. 1766–1787, Nov. 2020.

[35] E. Hashemi, R. Zarringhalam, A. Khajepour, W. Melek, A. Kasaiezadeh, and S.-K. Chen, "Real-time estimation of the road bank and grade angles with unknown input observers," *Vehicle Syst. Dyn.*, vol. 55, no. 5, pp. 648–667, May 2017.

[36] M. Da Lio, R. Donà, G. P. R. Papini, F. Biral, and H. Svensson, "A mental simulation approach for learning neural-network predictive control (in self-driving cars)," *IEEE Access*, vol. 8, pp. 192041–192064, 2020.

[37] M. Guiggiani, *The Science of Vehicle Dynamics: Handling, Braking, and Ride of Road and Race Cars*. Dordrecht, The Netherlands: Springer, 2018.

[38] O. Nelles, *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models* (Engineering Online Library). Berlin, Germany: Springer, 2001.

[39] A. Plebe, M. Da Lio, and D. Bortoluzzi, "On reliable neural network sensorimotor control in autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 711–722, Feb. 2020.

[40] J. C. Kegelman, L. K. Harbott, and J. C. Gerdes, "Insights into vehicle trajectories at the handling limits: Analysing open data from race car drivers," *Vehicle Syst. Dyn.*, vol. 55, no. 2, pp. 191–207, Feb. 2017.

[41] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–15.

[42] P. Welch, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Trans. Audio Electroacoustics*, vol. AU-15, no. 2, pp. 70–73, Jun. 1967.

[43] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, Jan. 1965.

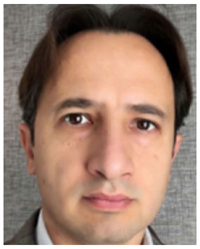
[44] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder–Mead simplex method in low dimensions," *SIAM J. Optim.*, vol. 9, no. 1, pp. 112–147, Jan. 1998.

[45] J. D'Errico, "fminsearchbnd, fminsearchcon," MATLAB Central File Exchange, 2021. [Online]. Available: https://it.mathworks.com/matlabcentral/fileexchange/8277-fminsearchbnd-fminsearchcon

Mauro Da Lio (Member, IEEE) received the Laurea degree in mechanical engineering from the University of Padua, Italy, in 1986. He is currently a Full Professor of mechanical systems with the University of Trento, Italy. His earlier research activity was on modeling, simulation, and optimal control of mechanical multibody systems, in particular vehicle and spacecraft dynamics. More recently his focus shifted to the modeling of human sensory-motor control, in particular drivers and motor impaired people. Prior to his academic career, he worked with an offshore oil research company in underwater robotics (a EUREKA Project). He was involved in several EU framework program 6 and 7 projects (PReVENT, SAFERIDER, interactIVe, VERITAS, AdaptIVe, and No-Tremor). He is also the Coordinator of the EU Horizon 2020 Dreams4Cars Research and Innovation Action: a collaborative project in the robotics domain which aims at increasing the cognition abilities of artificial driving agents by means of offline simulation mechanisms broadly inspired to the human dream state.

Mattia Piccinini (Member, IEEE) received the B.Sc. degree (cum laude) in industrial engineering and the M.Sc. degree (cum laude) in mechatronics engineering from the University of Trento, Italy, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree.

From March 2022 to June 2022, he was a Visiting Ph.D. Student with Universität der Bundeswehr, Munich, Germany. His research interests include motion planning, control, and state estimation methods for racing autonomous vehicles.



Francesco Biral received the master's degree in mechanical engineering from the University of Padua, Italy, and the Ph.D. degree in mechanism and machine theory from the University of Brescia, Italy, in 2000, for his work on minimum lap time of racing vehicles with the use of optimal control.

He is currently an Associate Professor with the Department of Industrial Engineering, University of Trento. He has 15 years experience in the development and validation of ADAS and AD functions, both for cars and PTWs, gained in several European and industrial funded research projects. His research interests include symbolic and numerical multi-body dynamics and optimization, constrained optimal control, mainly in the field of vehicle dynamics with a special focus on intelligent vehicles and optimal maneuver for racing vehicles.

Open Access funding provided by 'Università degli Studi di Trento' within the CRUI CARE Agreement