This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2025.3563347

Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version XX Month, XXXX.

Digital Object Identifier 10.1109/OJITS.2022.1234567

A Road Friction-Aware Anti-Lock Braking System Based on Model-Structured Neural Networks

MATTIA PICCININI[1], MATTEO ZUMERLE[2], JOHANNES BETZ[1], AND GASTONE PIETRO ROSATI PAPINI[2]

[1]Professorship of Autonomous Vehicle Systems, Technical University of Munich, 85748 Garching, Germany; Munich Institute of Robotics and Machine Intelligence (MIRMI)

[2]Department of Industrial Engineering, University of Trento, 38123 Trento, Italy

CORRESPONDING AUTHOR: Mattia Piccinini (e-mail: mattia.piccinini@tum.de).

ABSTRACT The anti-lock braking system (ABS) is a vital safety feature in modern vehicles, preventing wheel lock during emergency braking. However, the performance of conventional ABS is often limited by the lack of real-time road friction information. This paper introduces a novel road friction-aware ABS, leveraging model-structured neural networks (MS-NNs) to learn the vehicle longitudinal dynamics in different road conditions. Our framework uses a robust criterion to dynamically select from a set of pre-trained MS-NNs based on the available sensor data, enabling real-time road friction estimation and autonomous adaptation of the ABS parameters. Simulation experiments demonstrate that the proposed MS-NN-based ABS significantly improves safety and performance across varying road conditions: the braking distances are reduced by $3.0\% - 40.4\%$ compared to a conventional ABS, tuned for a specific road condition. Furthermore, the MS-NN's architecture shows better accuracy, generalization and sample-efficiency compared to other neural networks in the literature, and is suitable for real-time deployment on automotive-grade hardware. Our implementation is open source and available in a public repository.

INDEX TERMS Vehicle control, vehicle dynamics, road friction estimation, structured neural networks, explainable artificial intelligence, advanced driver assistance systems.

I. INTRODUCTION

THE anti-lock braking system (ABS) prevents wheel lock during emergency braking [1], and is a mandatory stability system in modern vehicles. The ABS operates by modulating the brake pressure at each wheel, allowing the driver to maintain steering control and avoid skidding. However, the performance of conventional ABS is limited by the lack of real-time road friction information. For example, on slippery or icy roads, a standard ABS may not prevent wheel lock, leading to longer stopping distances and increased risk of accidents.

In this paper, we propose a novel road friction-aware ABS, which autonomously adapts its parameters based on real-time road friction estimation. Our method leverages model-structured neural networks (MS-NNs) to learn the vehicle longitudinal dynamics in different road conditions. In an offline training phase, we collect a dataset of vehicle telemetries in various road conditions, and train a set of MS-NNs to predict the vehicle longitudinal acceleration in each condition. In the online operation phase, we use a robust criterion to estimate the road friction based on the MS-NNs' predictions and the measured vehicle acceleration. The estimated road friction is then used to adapt the ABS parameters, enhancing the braking performance across variable road conditions.

A. Related Work

This section offers a critical review of the related literature, starting from neural networks to model and control the vehicle dynamics, and moving on to friction-aware driver assistance systems.

1) Neural Networks to Model and Control the Vehicle Dynamics

In the recent literature, neural networks (NNs) have been used to model and control the vehicle dynamics [2],

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

[3]. Indeed, NNs may help overcoming the limitations of purely physics-based models [4], [5], whose design requires expertise, manual tuning, and testing campaigns that can be expensive and time-consuming[1]. One of the first examples of NNs in the field of vehicle dynamics is [7], which compared feedforward and convolutional NNs for path tracking at a constant vehicle speed. The authors of [8] used a long short-term memory (LSTM) NN to learn the steering dynamics of a racing vehicle model and execute time-optimal trajectories. For a similar purpose, a two-layer feedforward NN was adopted in [9]. Gated recurrent units were used by [10] to learn the longitudinal and lateral vehicle dynamics. The authors of [11] devised a NN to learn minimum-jerk vehicle maneuvers. In [12], an LSTM was employed to predict future control outputs in autonomous vehicle racing. In [13], [14], NNs and machine learning methods were used to learn stochastic policies and imitate the driving style of professional race car drivers. However, purely data-driven NNs with general-purpose internal architectures often struggle to generalize on unseen data.

Recently, some researchers tried to incorporate the prior knowledge of a system's dynamic laws into neural models, to favor explainability [15] and sample-efficiency. This new paradigm has two main approaches. The first one involves solving complex (but known) partial differential equations (PDEs) with NNs, commonly named *physics-informed neural networks* (PINNs) [16]. One of the very few examples of PINNs in the field of vehicle dynamics is [17]. The novelty of PINNs is that the loss function is the residual of the PDE to be solved. However, the PINNs' internal architectures are classical feedforward NNs. The second approach to incorporate prior knowledge involves neural networks with physics- or model-driven internal architecture, hereafter named *model-structured neural networks* (MS-NNs). The internal structure of an MS-NN embeds and/or extends some physical principles of the system to be modeled, and is designed to improve the generalization and explainability of the NN. One of the first examples of MS-NNs in the field of vehicle dynamics is [18], which modeled the longitudinal vehicle dynamics. The structure of their NN was *pre-wired* to satisfy some physical principles, like the superposition of forces and the Newton's law. In [19], the same authors devised an MS-NN to model and control the lateral vehicle dynamics. Their MS-NN was then extended by [20]–[22], to control the nonlinear lateral dynamics at the handling limits. The authors of [23] estimated the parameters of a single-track vehicle model using a NN, and constrained the NN parameters to lie in ranges with physical meaning. In [24], [25], neural tire models were integrated into a single-track car model. Other examples of MS-NNs are [26], [27], which controlled the steering dynamics for automated parking and learned optimal control vehicle maneuvers, and finally [28], which developed kinematics-structured NNs to estimate the lateral vehicle speed. Compared to general-purpose NNs, the MS-NNs resulted in improved explainability, accuracy and generalization with small training datasets [29].

[1]For example, the identification of tire models requires expensive tire test rigs [6].

2) Road Friction-Aware Driver Assistance Systems

Several studies proposed methods to estimate the tire-road friction potential. Recent surveys on the topic are [30]–[32]. The road friction estimation approaches can be classified in two main categories: *effect-based* and *cause-based* methods. The former relate the road friction to its effects on the vehicle dynamics, like the vehicle acceleration or the wheel slip. Some examples of effect-based methods are [33], [34], which used Kalman filters and neural networks. Cause-based methods, instead, correlate the road surface texture with the road friction, for instance using camera images and machine learning [35]. Our work falls into the effect-based category, which is further analyzed in the following.

In the field of anti-lock braking systems, the authors of [36] scheduled a motorcycle's ABS parameters based on the road friction coefficient and the roll angle. However, they assumed to know the road friction, rather than estimating it. Regarding the ABS control logic, most of the literature and practical implementations use finite state machines [1], [36], but some authors proposed alternative control strategies [37], like barrier Lyapunov functions [38], fuzzy control [39] and sliding mode control [40]. However, none of the previous works devised a road friction-aware ABS with online friction estimation and autonomous parameter adaptation. In [41], the road friction coefficient was estimated using pre-fitted tire models, and was used to adjust the longitudinal slip set-point of the ABS. The authors of [42] estimated the road surface condition by measuring the vehicle deceleration at three longitudinal slip values, and comparing the measurements with pre-fitted tire models. They then used the predicted road condition to adjust the ABS parameters. However, [41] and [42] assumed the prior knowledge of fitted tire models for different road surfaces, and required severe braking to estimate the road condition.

In a commercial application, Pagani (an Italian hypercar manufacturer) recently endowed their *Utopia* model with instrumented intelligent tires, named *CyberTyre* [43], to measure the tire forces and estimate the friction potential. Similarly, instrumented tires were used by [44] to estimate the road friction, in combination with a Brush tire model. These sensorized tires may be employed to enhance driver assistance systems like the ABS, but they are affordable only for very high-end vehicles.

3) Critical Summary

To the best of our knowledge, the existing literature is limited by at least one of the following aspects:

- MS-NNs have never been applied to friction-adaptive driver assistance systems.
- The few existing friction-aware ABS methods employ instrumented tires and/or the prior fitting of tire models, which is time-consuming and requires expensive measurement equipment.

B. Contributions and Structure of the Paper

The main paper contributions are:

- A novel model-structured neural network (MS-NN) to learn the vehicle longitudinal dynamics in different road conditions. The proposed MS-NN is compared with other neural networks in the literature, showing better accuracy and generalization.
- The use of a robust criterion to dynamically estimate the road friction based on standard vehicle sensor data, without the need for pre-fitted tire models.
- An adaptive ABS system that autonomously adjusts its parameters based on real-time road friction estimation.
- Simulation experiments to test the proposed framework in different road conditions, showing significant improvements in safety and performance compared to conventional ABS systems.

This paper is structured as follows. Section II presents an overview of our road friction-aware ABS framework and of the vehicle model used as simulator. Section III introduces our neural network to learn the longitudinal vehicle dynamics, using a novel model-structured neural architecture. Section IV outlines the estimation of the road friction, by training multiple neural networks and performing an online model selection. The control logic of our friction-aware ABS framework is described in Section V, while Section VI presents the main results. Finally, Section VII draws the conclusions and the directions of future work.

II. Framework Overview

Fig. 1 shows a block diagram of our road friction-aware ABS framework. In the online operation phase (right side of Fig. 1), the human driver controls the brake pedal p_b, which is processed by our adaptive ABS logic to modulate the brake torques $\boldsymbol{T}_{b_{\text{ABS}}}$ at the front and rear wheels. The ABS uses as input the vehicle speed v_x, the wheel rotational speeds $\boldsymbol{\omega}$, and a set of parameters $\boldsymbol{\beta}$, which are dynamically adapted based on the estimated road conditions. The road friction is assessed with a set of $\mathcal{Q}$ model-structured neural networks (MS-NNs): each of the MS-NNs is trained (offline) on a specific road condition, and predicts the longitudinal acceleration $\widehat{a}_x$ from past measurements of v_x, p_b, engine torque T_e, and road slope θ. A noise-robust criterion compares the measured vehicle acceleration a_x with the MS-NNs' predictions $\widehat{a}_{x_i}$, $i = 1, \ldots, \mathcal{Q}$, to estimate online the road friction. The estimate is then used to adapt the ABS parameters $\boldsymbol{\beta}$, enhancing the braking performance in non-nominal road conditions.

A. Vehicle Model used as Simulator

All the experiments in this paper are performed with a high-fidelity model of a C-segment electric rear-wheel-drive car, which is used as a simulator. We employ a double-track multibody vehicle model [4], [5], with the full Pacejka '96 magic formula [6] to model the tire forces.

Our vehicle model has 5 controls, namely the traction/brake torques for the four wheels and the steering angle δ_D, and the following 23 states:

- x, y, ψ: planar position and yaw angle of the vehicle's center of mass,
- v_x, v_y, Ω: longitudinal speed, lateral speed, and yaw rate,
- $F_{z_{rr}}, F_{z_{rl}}, F_{z_{fr}}, F_{z_{fl}}$: vertical forces on the four wheels,
- δ: steering angle at the ground,
- $\omega_{rr}, \omega_{rl}, \omega_{fr}, \omega_{fl}$: rotational speeds of the four wheels,
- $\alpha_{rr}, \alpha_{rl}, \alpha_{fr}, \alpha_{fl}$: side slip angles of the four tires,
- $\lambda_{rr}, \lambda_{rl}, \lambda_{fr}, \lambda_{fl}$: longitudinal slips of the four tires.

This model is integrated into our high-fidelity driving simulator, at the Industrial Engineering Department of the University of Trento (Italy). Our simulator employs the CARLA graphic environment [45] and runs the double-track vehicle model using real-time hardware.

By driving this vehicle model, we collect a dataset of vehicle telemetries in different road conditions. These telemetries are then used to train our MS-NNs for road friction estimation, as will be explained in the next sections.

III. Model-Structured Neural Network to Learn the Longitudinal Vehicle Dynamics

A. Proposed Architecture of the MS-NN

Let us now describe the novel architecture of our model-structured neural network (MS-NN) to learn the longitudinal vehicle dynamics. Fig. 2a shows the block diagram of the MS-NN, which computes the vehicle longitudinal acceleration $\widehat{a}_{x_k}$ at the time step k using the following inputs:

- A window of engine torque T_e values for the last n time steps: $\boldsymbol{T_e} = [T_{e_{k-n+1}}, \ldots, T_{e_k}]$,
- A window of brake pedal p_b values for the last n time steps: $\boldsymbol{p_b} = [p_{b_{k-n+1}}, \ldots, p_{b_k}]$,
- The current vehicle speed v_{x_k},
- The current road slope θ_k.

The sampling time for the input data windows $\{\boldsymbol{T_e}, \boldsymbol{p_b}\}$ is T_s, and is a hyperparameter of the MS-NN.

As shown in Fig. 2a, our MS-NN consists of two main parts: one model-based and the other data-driven. The model-based part combines the Newton dynamic equations with a *neuro-fuzzy* approach, and its output is the vehicle longitudinal acceleration $\widehat{a}_{x_{\text{model}_k}}$. The data-driven (DD) part learns the phenomena not captured by the model-based part, and its output is $\widehat{a}_{x_{\text{DD}_k}}$. Still, we design the data-driven

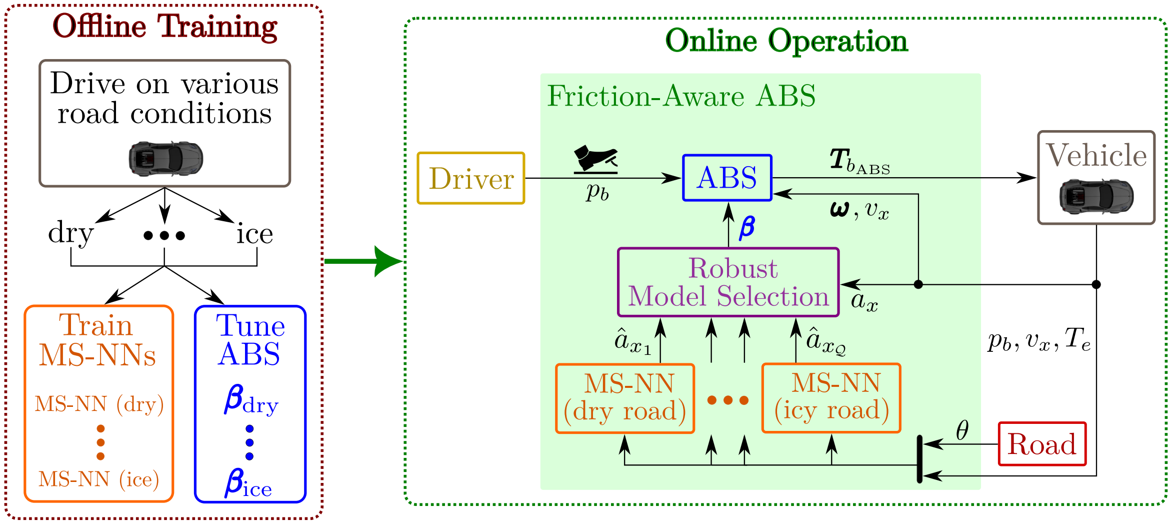


FIGURE 1. **Road friction-aware ABS framework: offline training (left) and online operation (right). In the online phase, the ABS modulates the brake torques $\boldsymbol{T}_{b_{\text{ABS}}}$ based on the estimated road friction. The road friction is assessed with a set of $\mathcal{Q}$ model-structured neural networks (MS-NNs), each trained on a specific road condition, and selected online with a noise-robust criterion.**

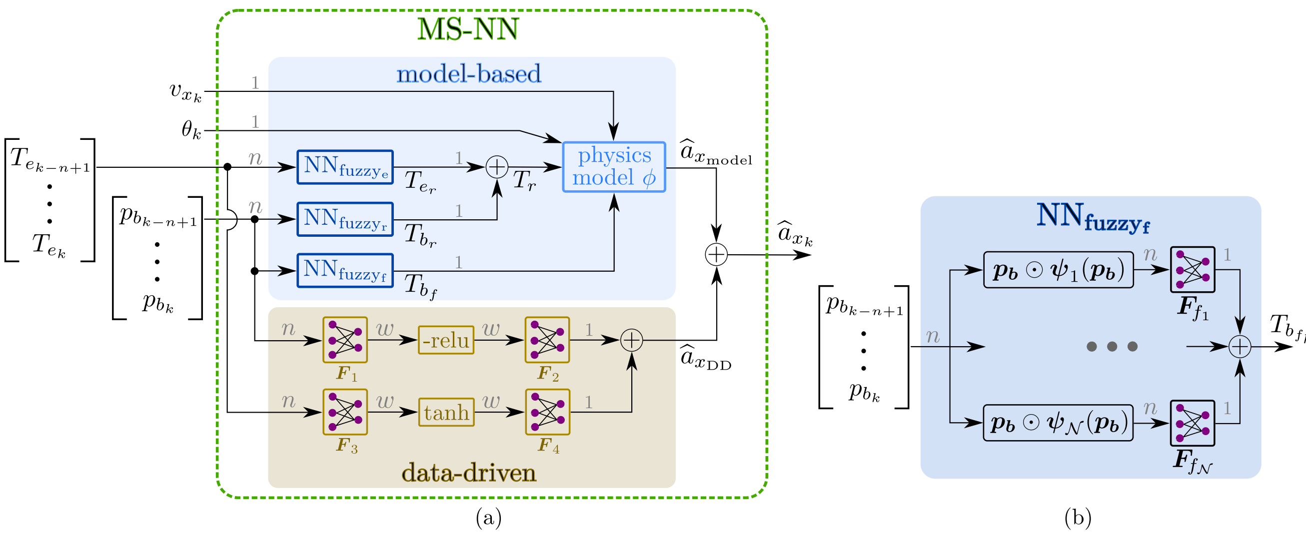


FIGURE 2. **(a) Internal architecture of our model-structured neural network (MS-NN) to learn the vehicle longitudinal dynamics. (b) local neuro-fuzzy models to estimate the front brake torque T_{b_f}. The numbers in gray over the arrows represent the data size of the corresponding signals.**

part to incorporate some prior knowledge of the vehicle dynamic laws, which improves its generalization potential and accuracy.

The final acceleration estimate $\widehat{a}_{x_k}$ is obtained as the sum of the model-based and data-driven parts:

$$\widehat{a}_{x_k} = \widehat{a}_{x_{\text{model}_k}} + \widehat{a}_{x_{\text{DD}_k}} \tag{1}$$

The following subsections illustrate the model-based and data-driven contributions in more detail.

Before proceeding, we remark that our MS-NN is conceptually different from the PINNs approach in the literature [16]. As explained in Section 1, PINNs have *structured* loss functions that enforce the solution of known PDEs, but their internal architectures are classical feedforward NNs. In contrast, our MS-NN has a *structured* internal architecture that embeds the vehicle dynamics principles, which further enhances the interpretability and sample-efficiency.

1) Model-Based Part

Let us start from the Newton equation modeling the longitudinal vehicle dynamics [4]:

$$a_x = \frac{F_{x_f} + F_{x_r} - c_v\, v_x - k_d\, v_x^2}{m} - g\, c_r \cos(\theta) - g \sin(\theta) \tag{2}$$

where m is the vehicle mass, c_v and k_d are the laminar and aerodynamic drag coefficients, g is the gravitational acceleration, c_r is the rolling resistance coefficient, θ is the road slope, F_{x_f} and F_{x_r} are the longitudinal forces at the front and rear axles. To express the longitudinal forces F_{x_f} and F_{x_r}, let us consider the rotational wheel dynamics for the front and rear axles:

$$\begin{cases} 2I_{w_f}\dot{\omega}_f = -r_f F_{x_f} + T_f \longrightarrow F_{x_f} = \dfrac{T_f - 2I_{w_f}\dot{\omega}_f}{r_f} & \text{(3a)} \\ 2I_{w_r}\dot{\omega}_r = -r_r F_{x_r} + T_r \longrightarrow F_{x_r} = \dfrac{T_r - 2I_{w_r}\dot{\omega}_r}{r_r} & \text{(3b)} \end{cases}$$

where $\{I_{w_f}, I_{w_r}\}$, $\{\omega_f, \omega_r\}$, $\{r_f, r_r\}$, and $\{T_f, T_r\}$ are the front and rear wheel inertias, angular velocities, radii, and torques, respectively. Equations (3a,3b) assume that the rotational speed of the right and left wheels on the same axle is the same (pure longitudinal dynamics). Moreover, under the assumption that the longitudinal slips are small, the wheel accelerations $\{\dot{\omega}_f, \dot{\omega}_r\}$ in (3a,3b) can be related to the vehicle longitudinal acceleration a_x as follows:

$$\dot{\omega}_f \approx \frac{\dot{v}_x}{r_f} = \frac{a_x}{r_f} \quad \text{and} \quad \dot{\omega}_r \approx \frac{\dot{v}_x}{r_r} = \frac{a_x}{r_r} \tag{4}$$

By substituting (3) and (4) into (2), we can write the longitudinal acceleration estimate $\widehat{a}_{x_{\text{model}}}$ of our physics-based model $\phi(\cdot)$ as:

$$\widehat{a}_{x_{\text{model}}} = \phi(T_f, T_r, v_x, \theta) = \\ = \frac{\frac{1}{m}\left(\frac{T_f}{r_f} + \frac{T_r}{r_r} - k_d\, v_x^2 - c_v\, v_x\right) - g\, c_r \cos(\theta) - g \sin(\theta)}{1 + \frac{2}{m}\left(I_{w_f}/r_f^2 + I_{w_r}/r_r^2\right)} \tag{5}$$

Neuro-Fuzzy Local Models of the Wheel Torques

The physics-based model $\phi(\cdot)$ in (5) uses as input the wheel torques T_f and T_r, at the front and rear axles. In this section, we devise neuro-fuzzy local models to estimate $\{T_f, T_r\}$ from the brake pedal p_b and the engine torque T_e, which are the actual inputs of our MS-NN. As shown in Fig. 2a, these neuro-fuzzy models (NN_{fuzzy}) are incorporated into the model-based part of the MS-NN, and they are trained together with the rest of the network.

Let us start from T_f, which is the total torque at the front wheels. Since the vehicle is rear-wheel-drive, then T_f is equal to the braking torque T_{b_f} at the front wheels. Our neuro-fuzzy model, named $\text{NN}_{\text{fuzzy}_\text{f}}$, is shown in Fig. 2b and is designed as:

$$T_f = T_{b_f} = \text{NN}_{\text{fuzzy}_\text{f}}(\boldsymbol{p_b}) = \sum_{i=1}^{\mathcal{N}} \overbrace{\left(\boldsymbol{p_b} \odot \boldsymbol{\psi}_{b_i}(\boldsymbol{p_b})\right) \cdot \begin{bmatrix} F_{f_{1_i}} \\ \vdots \\ F_{f_{n_i}} \end{bmatrix}}^{i-\text{th local model}} \tag{6}$$

where the symbol $\odot$ denotes the Hadamard element-wise vector product. The input $\boldsymbol{p_b} = [p_{b_{k-n+1}}, \dots, p_{b_k}]$ is a vector of brake pedal values for the last n time steps. Using a history of past samples for $\boldsymbol{p_b}$ allows our model to learn the braking dynamics [18] and to predict the brake torque T_{b_f} at the current time step. Moreover, in (6) we use $\mathcal{N}$ local models to learn the braking dynamics in different ranges of p_b. The $i-$th local model is activated around a center point $p_{b_{0i}}$, through a fuzzy-like activation function $\boldsymbol{\psi}_{b_i}(\boldsymbol{p_b})$:

$$\boldsymbol{\psi}_{b_i}(\boldsymbol{p_b}) = \begin{bmatrix} \psi_{b_i}(p_{b_{k-n+1}}) & \dots & \psi_{b_i}(p_{b_{k-1}}) & \psi_{b_i}(p_{b_k}) \end{bmatrix} \tag{7}$$

with $i = 1, \dots, \mathcal{N}$. As depicted in Fig. 3, $\psi_{b_i}(p_b)$ is piecewise linear in p_b, $\psi_{b_i}(p_b) = 1$ for $p_b = p_{b_{0i}}$, and $\sum_{i=1}^{\mathcal{N}} \psi_{b_i}(p_b) = 1$. The centers $p_{b_{0i}}$ and the numbers of local models $\mathcal{N}$ are hyperparameters of the $\text{NN}_{\text{fuzzy}_\text{f}}$ model, and are found through a grid search (Table 1).

As shown in Fig. 2b, the $i-$th local model of (6) has a fully connected layer $\boldsymbol{F}_{f_i}$, with n learnable weights $\{F_{f_{1_i}}, \dots, F_{f_{n_i}}\}$. $\boldsymbol{F}_{f_i}$ learns a linear combination of the window $\boldsymbol{p_b}$ of past brake pedal values, and models the discrete-time impulse response of the braking dynamics around $p_b = p_{b_{0i}}$ [18]. Finally, the outputs of all the local models are summed, yielding the front brake torque T_f in (6), which is then fed to (5).

Similarly, the rear wheel torque T_r is modeled as:

$$\begin{cases} T_{e_r} = \text{NN}_{\text{fuzzy}_\text{e}}(\boldsymbol{T_e}) = \sum_{j=1}^{\mathcal{M}} \left(\boldsymbol{T_e} \odot \boldsymbol{\psi}_{e_j}(\boldsymbol{T_e})\right) \begin{bmatrix} F_{e_{1_j}} \\ \vdots \\ F_{e_{n_j}} \end{bmatrix} & \text{(8a)} \\ T_{b_r} = \text{NN}_{\text{fuzzy}_\text{r}}(\boldsymbol{p_b}) = \sum_{i=1}^{\mathcal{N}} \left(\boldsymbol{p_b} \odot \boldsymbol{\psi}_{b_i}(\boldsymbol{p_b})\right) \begin{bmatrix} F_{r_{1_i}} \\ \vdots \\ F_{r_{n_i}} \end{bmatrix} & \text{(8b)} \\ T_r = T_{e_r} + T_{b_r} & \text{(8c)} \end{cases}$$

where T_r is the sum of the engine-generated torque T_{e_r} and the rear brake torque T_{b_r}. The engine-generated torque

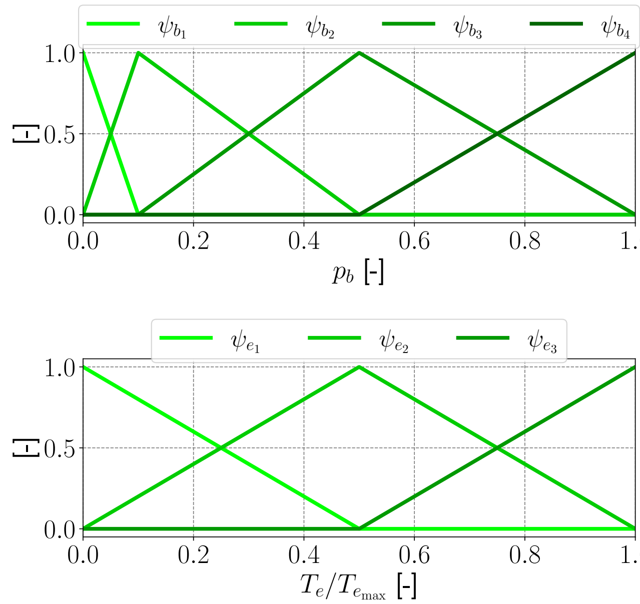


FIGURE 3. Activation functions $\psi_{b_i}(p_b)$ and $\psi_{e_j}(T_e)$, $i = 1, \dots, \mathcal{N}$ and $j = 1, \dots, \mathcal{M}$, for the neuro-fuzzy models $\text{NN}_{\text{fuzzy}_\text{f}}(\boldsymbol{p_b})$, $\text{NN}_{\text{fuzzy}_\text{r}}(\boldsymbol{p_b})$ and $\text{NN}_{\text{fuzzy}_\text{e}}(\boldsymbol{T_e})$ in (6) and (8). The best values for the numbers of local models are $\mathcal{N} = 4$ and $\mathcal{M} = 3$, and they are found through a grid search (Table 1). Note that the brake pedal p_b ranges in $[0, 1]$, while the engine torque T_e is normalized with respect to its maximum value $T_{e_{\max}}$.

T_{e_r} is estimated using the neuro-fuzzy model $\mathrm{NN}_{\mathrm{fuzzy_e}}(\boldsymbol{T_e})$, whose input is the past history of the engine torque $\boldsymbol{T_e} = [T_{e_{k-n+1}}, \ldots, T_{e_k}]$. $\mathrm{NN}_{\mathrm{fuzzy_e}}(\boldsymbol{T_e})$ uses $\mathcal{M}$ local models, and its architecture (8a) is similar to the one of $\mathrm{NN}_{\mathrm{fuzzy_f}}(\boldsymbol{p_b})$ in (6). The design of the activation functions $\boldsymbol{\psi}_{e_j}(\boldsymbol{T_e})$ is analogous to the $\boldsymbol{\psi}_{b_i}(\boldsymbol{p_b})$, and they are plotted in Fig. 3.

Finally, the rear brake torque T_{b_r} is estimated using the neuro-fuzzy model $\mathrm{NN}_{\mathrm{fuzzy_r}}(\boldsymbol{p_b})$ in (8b), which has the same structure as the $\mathrm{NN}_{\mathrm{fuzzy_f}}(\boldsymbol{p_b})$ previously described.

Learnable Parameters in the Model-Based Part

To summarize, the learnable parameters in the model-based part of the MS-NN are the following:

- In the analytical relation (5): the vehicle mass m, the laminar and aerodynamic drag coefficients $\{c_v, k_d\}$, the rolling resistance coefficient c_r, the front and rear wheel inertias $\{I_{w_f}, I_{w_r}\}$,
- In the neuro-fuzzy local models (6)-(8): the weights $\{F_{f_{1_i}}, \ldots, F_{f_{n_i}}\}$, $\{F_{e_{1_j}}, \ldots, F_{e_{n_j}}\}$, and $\{F_{r_{1_i}}, \ldots, F_{r_{n_i}}\}$, with $i = 1, \ldots, \mathcal{N}$ and $j = 1, \ldots, \mathcal{M}$.

All these parameters are treated as learnable weights for the training. Section B explains the training method and the dataset collection for our MS-NN. Let us now describe the data-driven part of the MS-NN.

2) Data-Driven Part

The data-driven part of our MS-NN (gold block in Fig. 2a) learns the effects not captured by the model-based part, and is an extension of the structured convolution-like NN of [18].

The data-driven part consists of two branches. The upper branch in Fig. 2a processes the window of brake pedal values $\boldsymbol{p_b}$ with a two-layer feedforward neural network: the first fully-connected layer ($\boldsymbol{F}_1$ in Fig. 2a) has w neurons and a rectified linear unit (relu) activation function, and the second layer ($\boldsymbol{F}_2$) has a single neuron. The relu (with a minus sign in front) encodes the fact that the brake pedal p_b can cause only negative accelerations.

The second branch uses a similar two-layer NN (with a $\tanh$ activation function) to process the window of engine torque values $\boldsymbol{T_e}$.

The two branches provide the braking and traction effects due to the brake pedal and engine torque, respectively. Finally, the outputs of the two branches are summed, to preserve the additivity of forces in the Newton's law (2). Thus, the data-driven side of our MS-NN is still partly inspired by the physical laws of the vehicle dynamics.

Learnable Parameters in the Data-Driven Part

The learnable parameters in the data-driven part of our MS-NN are the weights and biases of the fully-connected layers: $\boldsymbol{F}_1$ (with w neurons), $\boldsymbol{F}_2$ (1 neuron), $\boldsymbol{F}_3$ (w neurons), and $\boldsymbol{F}_4$ (1 neuron). w is a hyperparameter of the MS-NN, and is tuned as described in Section 2. The model-based and data-driven parts of our MS-NN are trained together, and Section B explains the training method.

B. MS-NN Implementation with the NNodely Open Framework

Implementing highly structured neural networks, such as MS-NNs, can be challenging when using general-purpose machine learning libraries like PyTorch. These tools, while versatile, require significant expertise in machine learning to manually encode domain-specific knowledge into neural architectures, making the process complex and time-consuming.

Our new open-source software tool, named NNodely, simplifies this process by providing a modular and intuitive framework designed specifically for MS-NNs. The framework simplifies the process of embedding physical principles into neural networks, reducing the complexity of development and making these methods accessible to users with a background in classical modeling, such as engineers and physicists, while supporting deployment in PyTorch and ONNX.

In this paper, NNodely is used to enable rapid exploration of multiple MS-NN architectures. Our approach significantly accelerates the development process while maintaining the flexibility required for experimentation. The code for the implemented network is available at `https://github.com/tonegas/nnodely-applications/tree/main/vehicle/road_friction_aware_ABS`. Also, the NNodely framework is released as an open-source software, and can be accessed at `https://github.com/tonegas/nnodely`.

IV. Road Friction Estimation

A. Workflow Overview

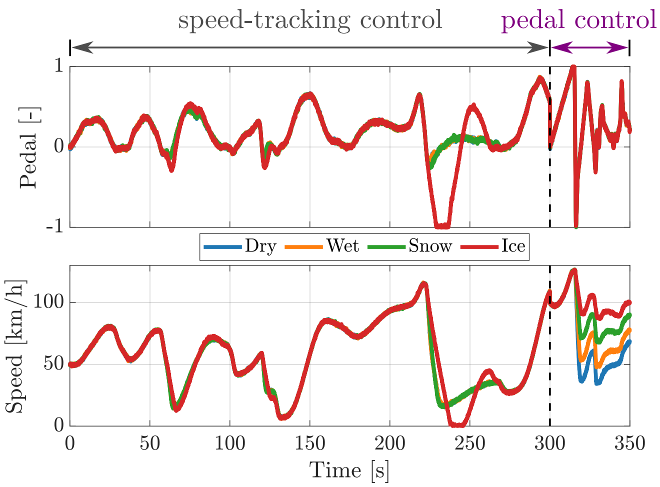
This section explains our method for real-time road friction estimation. Section B describes the data collection and training of the MS-NNs on different road conditions, which is part of the offline training step in Fig. 1. Section C deals with the real-time model selection for road friction estimation, which is carried out in the online operation phase of Fig. 1.

B. Training the MS-NNs on Different Road Conditions

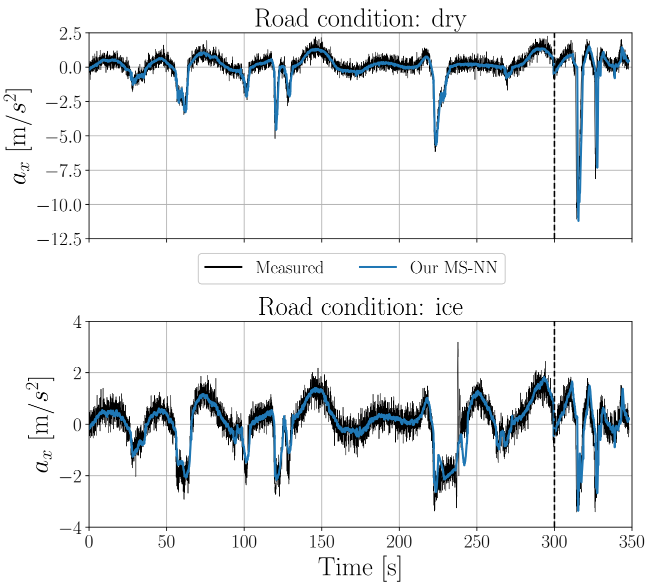
1) Training Dataset

We collect the training datasets for our MS-NNs by driving our vehicle simulator (Section A) in four different road conditions: dry, wet, snowy, and icy asphalt. We generate a target speed profile that can be executed in nearly all road conditions, and we use a PID controller with speed-dependent gains to let the vehicle simulator track the desired speed [20].

Fig. 4a shows the speed v_x profiles and the corresponding pedal p profiles ($p \in [0, 1]$ for throttle, $p \in [-1, 0]$ for braking) used for training. The training datasets in Fig. 4a consist of two parts (separated by a dashed line at 300 s): in



(a)



(b)

FIGURE 4. (a) Pedal and speed profiles used to train our MS-NN on different road conditions: dry, wet, snow and ice. The dashed line at 300 s separates two parts of the training dataset: the first part tracks a given speed profile with a tracking controller, while the second sets the pedal control directly, to better explore the domain. (b) Comparison between the measured acceleration a_x and the MS-NN's predictions on the dry (top) and ice (bottom) training sets.

the first part (up to 300 s), our speed-tracking PID controller computes the pedal profile to track a given speed trajectory, which we design to explore most of the vehicle speed range. In the second part (last 50 s), the pedal profile is manually set to better explore the pedal range. Note that, in the last 50 s, the vehicle speed on ice is higher than on dry asphalt: this is because, during the hard-braking event around 315 s, the vehicle decelerates less on ice than on dry asphalt (for the same pedal values), and then keeps a higher speed until the end of the profile.

Hyperparameter	Meaning	Value
T_s	Sampling time of input data windows	0.05 s
n	Length of the input data windows	10
$\mathcal{N}$	N. neuro-fuzzy local models for the brake pedal p_b	4
$\mathcal{M}$	N. neuro-fuzzy local models for the engine torque T_e	3
w	N. neurons in the fully connected layers $\boldsymbol{F}_1$, $\boldsymbol{F}_3$	5

TABLE 1. Hyperparameters of our MS-NN, tuned with a grid-search.

We remark that the training dataset for each road condition has a duration of less than 6 minutes (350 s, corresponding to 7000 samples with a sampling time $T_s = 0.05$ s), which is very short in comparison with the large amounts of data typically required to train traditional deep neural networks [46]. Indeed, the model-structured architecture of our NN needs small training datasets to generalize well on unseen scenarios.

2) Training and Hyperparameters Tuning

For each of the four road conditions, we train a model-structured neural network (MS-NN) on the collected datasets. The MS-NNs are trained with supervised learning, where the loss function is the mean squared error between the predicted and the actual vehicle acceleration a_x. We use the Adam optimizer [47], with a batch size of 128 and 200 epochs. Thanks to their model-structured architecture, our MS-NNs need a very low number of training epochs to converge, which is an advantage in terms of training time and computational resources.

The main hyperparameters of the MS-NN's internal architecture are tuned with a grid-search approach, and the final hyperparameter values are reported in Table 1.

Fig. 4b shows that our MS-NNs can accurately predict the vehicle acceleration a_x in different road conditions. Section VI will assess the generalization capabilities of our MS-NNs on unseen data, and will compare the MS-NNs against other neural networks in the literature.

C. Real-time Road Friction Estimation through Robust Model Selection

As explained in Section 2 and Fig. 1, we train four MS-NNs on four corresponding road conditions. During online operation, we need to select the most likely MS-NN for the current road condition, to adapt the parameters of the ABS system accordingly.

As shown in Fig. 1, we use a model selection algorithm to choose the most likely road condition. At each time step, this algorithm compares the measured longitudinal acceleration a_x with the predictions $\widehat{a}_{x_i}$, $i = 1, \ldots, \mathcal{Q}$, of our $\mathcal{Q} = 4$ MS-NN models. The algorithm then selects the model with

Parameter	Meaning	Value
σ_e	Parameter of the evidence function (9b)	0.18 m/s^2
N_s	Window size for the salience (10)	10
$\mathcal{L}_{\text{th}}$	Likelihood threshold for the model selection (12)	0.26

TABLE 2. Parameters of the WTA and MSPRT algorithms for model selection.

the highest likelihood, and uses the corresponding road condition to adapt the ABS parameters.

We evaluate two model selection algorithms: the *winner takes all* (WTA) and the *multi-hypothesis sequential probability ratio test* (MSPRT) [48]–[50]. Section 1 will quantitatively compare these two algorithms, and will show that the MSPRT performs a more robust model selection in the presence of noise. In the following, we provide a brief description of the two algorithms.

1) Winner Takes All (WTA)

The WTA selects the model with the highest evidence at each time step. In our case, we compare the predictions of the $\mathcal{Q} = 4$ MS-NN models. At the time step k, the prediction error e_{i_k} of the $i-$th model is defined as (9a), and we compute the corresponding evidence ϵ_{i_k} with a Gaussian function (9b):

$$\begin{cases} e_{i_k} = a_{x_k} - \widehat{a}_{x_{i_k}} & \text{(9a)} \\ \epsilon_{i_k} = \exp\left(-\dfrac{e_{i_k}^2}{2\sigma_e^2}\right) & \text{(9b)} \end{cases}$$

where σ_e is a tunable parameter, whose numerical value is reported in Table 2. At each time step k, the WTA selects the model with the highest evidence ϵ_{i_k}, and uses it to adapt the ABS parameters. However, the WTA may lead to very frequent model switches in the presence of measurement noise, which can be detrimental for the ABS performance.

2) Multi-Hypothesis Sequential Probability Ratio Test (MSPRT)

In contrast to the WTA, the MSPRT accumulates evidence over time to make a more robust model selection [50]. The accumulated evidence is named *salience*, and is calculated at each time step k for the $i-$th model as the mean of the past evidence values:

$$S_{i_k} = \frac{1}{N_s} \sum_{j=k-N_s+1}^{k} \epsilon_{i_j} \tag{10}$$

where N_s is the window size for the mean calculation, whose value is reported in Table 2, and the evidence ϵ_{i_j} is defined in (9b). Finally, to select the most likely model, we compute the likelihood $\mathcal{L}_{i_k}$ of the $i-$th model as:

$$\mathcal{L}_{i_k} = \exp\left(S_{i_k} - \log\left(\sum_{j=1}^{\mathcal{Q}} \exp(S_{j_k})\right)\right) \tag{11}$$

where $\mathcal{Q}$ is the number of models (in our case $\mathcal{Q} = 4$), $\mathcal{L}_{i_k} \in [0, 1]$, and the sum of the likelihoods of all models is equal to 1, *i.e.,* $\sum_{i=1}^{\mathcal{Q}} \mathcal{L}_{i_k} = 1$. When the likelihood of the $i-$th model exceeds a threshold value $\mathcal{L}_{\text{th}}$, then we select this model, and use it to adapt the ABS parameters. Otherwise, we keep the previously selected model. If more than one model exceeds the threshold, we select the one with the highest likelihood:

$$i_k = \begin{cases} \arg\max_i \mathcal{L}_{i_k} & \text{if } \mathcal{L}_{i_k} > \mathcal{L}_{\text{th}} \\ i_{k-1} & \text{otherwise} \end{cases} \tag{12}$$

where i_k is the index of the selected model at time step k, with $i_k = 1, \ldots, \mathcal{Q}$. The threshold value $\mathcal{L}_{\text{th}} \in [0, 1]$ is a tunable parameter, whose numerical value is reported in Table 2. Low values of $\mathcal{L}_{\text{th}}$ lead to frequent model switches, while higher values of $\mathcal{L}_{\text{th}}$ yield a slower but more noise-robust model selection.

Note that we do not need to set any friction coefficient thresholds for our online road friction estimation. Instead, we need to define only the likelihood threshold $\mathcal{L}_{\text{th}}$, above which a model is selected during operation.

All the parameters of the WTA and MSPRT algorithms in Table 2 are tuned with a grid-search approach, to improve the braking performance (*i.e.,* minimize the braking distances) in the test scenarios of Section VI.

V. Road Friction-Aware Anti-Lock Braking System

This section describes the working principles of our ABS, starting from the control logic (Section A), and then explaining the ABS parameters tuning (Section B) and the online adaptation (Section C).

A. ABS Control Logic

Fig. 5a depicts a block diagram of our ABS control system. The ABS operates on each wheel individually, and uses the following inputs: the set of requested wheel brake torques $\boldsymbol{T_b}$ (which depend on the driver's brake pedal command), the measured wheel rates $\boldsymbol{\omega}$ and vehicle speed v_x, and a set of ABS parameters $\boldsymbol{\beta}$ that are adapted in real-time by the road friction estimation module. The ABS outputs the actual brake torques $\boldsymbol{T_{b_{\text{ABS}}}}$ to be applied to the wheels.

We adopt the four-state ABS control logic of [1, Chapter 4], which is based on the finite state machine (FSM) in Fig. 5b. For each wheel, the FSM takes as input the requested brake torque T_b and the wheel slip λ, which we estimate using the method in [1, Chapter 5]. The FSM's control output u can have three values, $u \in \{+k, 0, -k\}$, where k is a constant that represents the maximum rate of change of the brake torque. The four states of the FSM are $q = \{0, 1, 2, 3\}$, and they correspond to the following actions, depicted in Fig. 5b and 6a:

- $q = 0 \rightarrow$ it is the initial state, when a braking maneuver starts. The control u is set to k: the brake torque T_b can increase up to a threshold $T_{b_{\max}}$ (Σ_3 line in Fig. 6a).

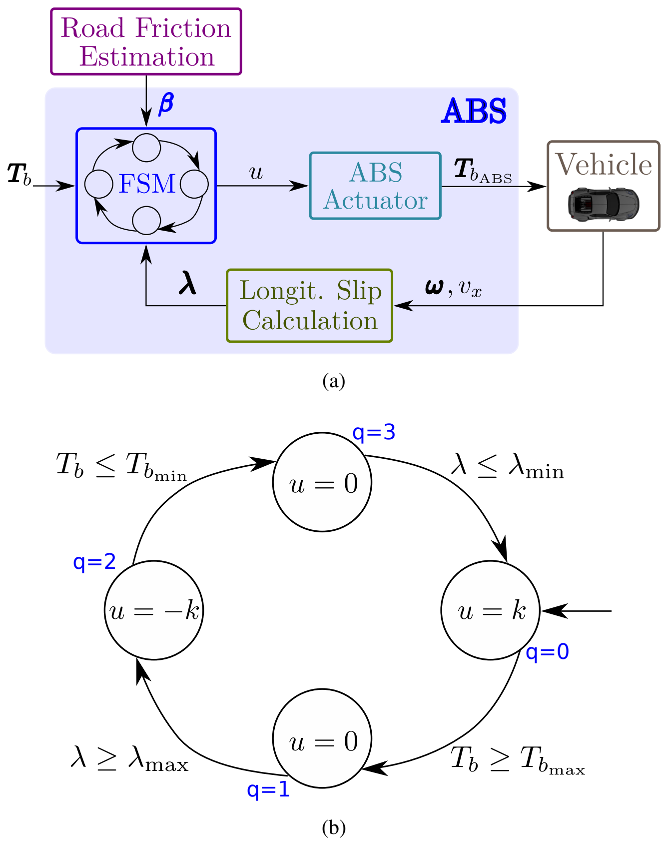


FIGURE 5. **(a) Block diagram of the ABS control system, and (b) finite state machine (FSM) of the ABS control logic.**

- $q = 1 \rightarrow$ the control u is set to 0: the brake torque T_b is kept constant until the wheel slip λ reaches a threshold λ_{max} (Σ_1 line in Fig. 6a).
- $q = 2 \rightarrow$ the control u is set to $-k$: the brake torque T_b decreases until it reaches a threshold $T_{b_{\text{min}}}$ (Σ_2 line in Fig. 6a).
- $q = 3 \rightarrow$ the control u is set to 0: the brake torque T_b is kept constant until the slip λ drops to a threshold λ_{min} (Σ_4 line in Fig. 6a).

Hence, the set $\boldsymbol{\beta}$ of tunable parameters of the ABS logic is given by:

$$\boldsymbol{\beta} = \{T_{b_{\text{max}}}, T_{b_{\text{min}}}, \lambda_{\text{min}}, \lambda_{\text{max}}\} \tag{13}$$

B. ABS Parameters Tuning

The ABS parameters $\boldsymbol{\beta}$ in (13) are tuned by combining a theoretical approach with braking experiments.

The theoretical method is based on the so-called equilibrium manifold $\Psi(\lambda)$, plotted in Fig. 6 for the vehicle model of this paper (Section A) under different road conditions. Following [1], $\Psi(\lambda)$ is a set of equilibrium points between the wheel slip λ and the brake torque T_b. $\Psi(\lambda)$ can be derived analytically from a single-corner vehicle model, and the interested reader can find the details in [1]. Using the equilibrium manifold in Fig. 6a, we suitably tune the ABS parameters $\boldsymbol{\beta}$ in (13) to ensure the stability of the limit cycle (red box in the figure): the conditions for the existence and stability of the limit cycle are discussed in [1, Chapter 4]. In practice, these conditions require that the limit cycle must be centered around the peak of the equilibrium manifold curve, as depicted in Fig. 6a. Using the theory of the equilibrium manifold, we tune the ABS parameters $\boldsymbol{\beta}$ in the four nominal road conditions (dry, wet, snow, ice) described in Section B.

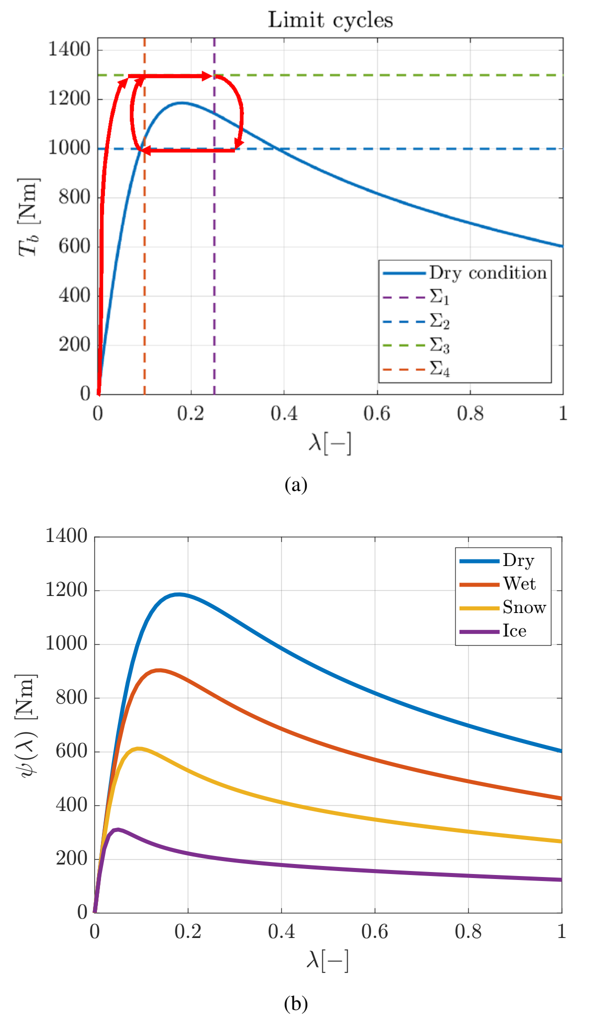


FIGURE 6. **(a) Stable limit cycle (red box) in the equilibrium manifold diagram of our four-state ABS control logic. The positions of the four lines $\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4$ correspond to the ABS parameters $\boldsymbol{\beta}$ in (13), which we tune to ensure the stability of the limit cycle. (b) Equilibrium manifold $\Psi(\lambda)$ under the four road conditions used to train our framework.**

However, the equilibrium manifold $\Psi(\lambda)$ is a theoretical and simplified construct, and the real vehicle behavior may locally differ from it. Thus, the ABS parameters $\boldsymbol{\beta}$ found with the equilibrium manifold analysis are further fine-tuned with the braking maneuver in Fig. 7. This test consists of two braking phases, with different brake pedals and vehicle speed. For each of the four road conditions used to train our framework, we fine-tune the ABS parameters $\boldsymbol{\beta}$ to keep the wheel slips in bounded regions and avoid wheel lock. As shown in Fig. 7 for the cases of wet and icy roads, our tuned

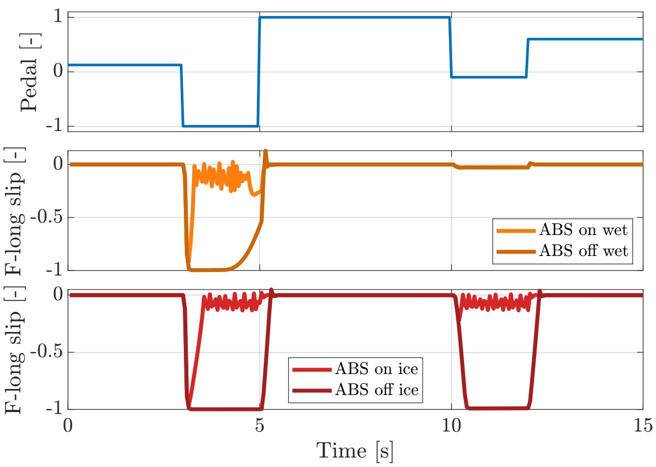


FIGURE 7. Braking maneuver used to fine-tune the ABS parameters β in the different road conditions. Top: pedal profile (< 0 for braking); center and bottom: front wheel slips λ in the wet and icy road conditions. When the ABS is OFF, the wheel slips go to -1 (wheel lock), while the ABS ON keeps the slips in bounded regions, which decreases the stopping distances.

Parameter	Dry front/rear	Wet front/rear	Snow front/rear	Ice front/rear
$T_{b_{\max}}$ [Nm]	1250/830	700/650	600/550	300/220
$T_{b_{\min}}$ [Nm]	500/500	350/350	200/200	50/50
$\lambda_{\max}$ [-]	0.25/0.25	0.17/0.17	0.14/0.14	0.12/0.12
$\lambda_{\min}$ [-]	0.1/0.1	0.075/0.075	0.03/0.03	0.015/0.015

TABLE 3. Tuned ABS threshold parameters for the front and rear axles, under the four road conditions used to train our framework.

ABS limits the wheel slips in ranges close to the peaks of the equilibrium manifolds, which reduces the stopping distances.

Table 3 reports the final tuned values of the ABS parameters $\boldsymbol{\beta}$ for the front and rear axles, under the four road conditions used to train our framework (Section B).

C. ABS Parameters Adaptation through Road Friction Estimation

As explained in the previous section, we tune the ABS parameters $\boldsymbol{\beta}$ in (13) for four road conditions (dry, wet, snow, ice). Also, we train our MS-NNs on the same road conditions (Section B). At inference time (online operation), we use the road friction estimation module (Section C) to select the most likely road condition, and we adapt the ABS parameters $\boldsymbol{\beta}$ accordingly.

VI. Results

A. Simulation Environment

The results shown in this paper are obtained with our vehicle model of Section A, which we use as a simulator. Our vehicle model and ABS controller are implemented in MATLAB-SIMULINK, from which we generate the C++ code to be run in our simulations.

The measured vehicle states and controls (vehicle speed, acceleration, wheel rates, brake pedal, engine torque) are provided by sensor models, which contain realistic noise levels.

Our MS-NNs and the online model selection algorithm are implemented in Python, using the NNodely open framework, and our code is open source (Section B). During operation, the MS-NNs and the model selection algorithm run in parallel with the vehicle model simulator.

B. Test Scenario

Fig. 8 shows the test scenario used to evaluate the performance of our adaptive ABS system. The test scenario is different from the training conditions: as depicted in Fig. 8a, the vehicle experiences varying and non-nominal road friction coefficients, from near-dry to near-icy asphalt. Also, the pedal and speed profiles in Fig. 8b cover a wide range of the operating domain, and they differ from the training profiles in Fig. 4a.

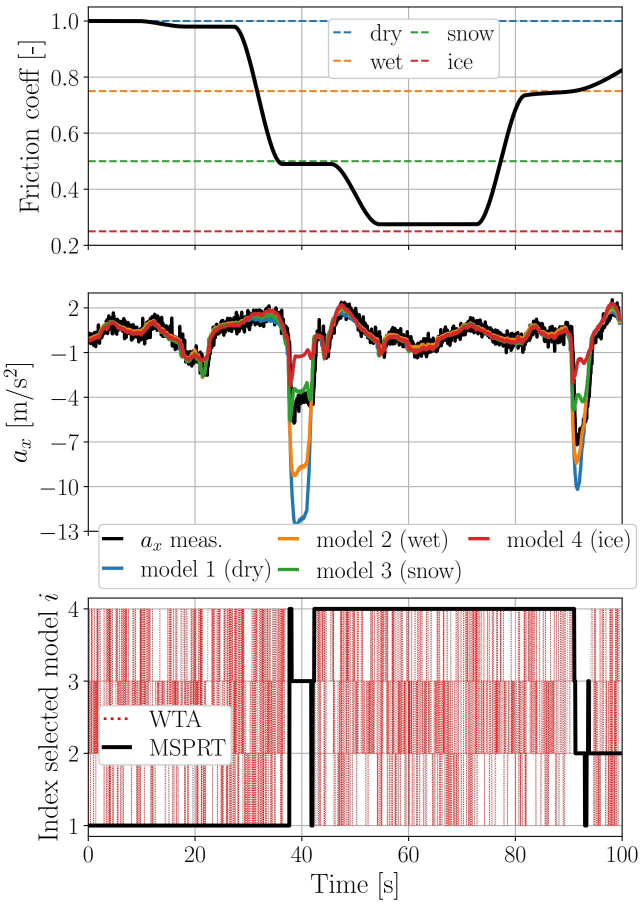
C. Road Friction Estimation and ABS Parameters Adaptation

Let us now evaluate the performance of our friction-aware ABS system in the test scenario of Fig. 8. We start by analyzing the road friction estimation, and then we assess the performance of our adaptive ABS in comparison with a standard ABS.

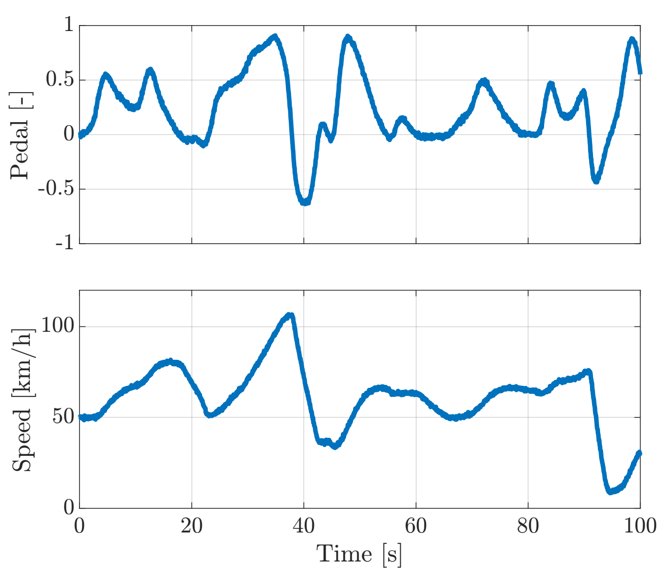
1) Road Friction Estimation: WTA versus MSPRT

The central plot in Fig. 8a compares the measured a_x acceleration signal with the predictions $\widehat{a}_{x_i}$, $i = 1, \ldots, \mathcal{Q}$, of our $\mathcal{Q} = 4$ MS-NN models, each of which is trained on a different road condition (respectively dry, wet, snowy and icy asphalt). When the vehicle brakes, the prediction errors vary significantly among the 4 models. For example, in the first braking event (around $t = 40$ s), the prediction of the model trained on snowy asphalt (green line) is the closest to the measured acceleration, and indeed the road condition is near-snowy.

The bottom plot in Fig. 8a shows the index i of the selected model at each time step, $i = 1, \ldots, \mathcal{Q}$, and compares the WTA and MSPRT selection algorithms (Section C). The WTA (red line) selects the model with the highest evidence at each time step, but yields very frequent model switches in the low-acceleration regions, due to the noise in the acceleration signal. These frequent switches can result in a suboptimal adaptation of the ABS parameters, with potential stability issues and ABS performance degradation. Indeed, the WTA was shown to work well only in noise-free simulated scenarios [51]. Conversely, the MSPRT (black line in the bottom plot of Fig. 8a) accumulates evidence over time, and makes a more robust model selection in the presence of noise. Indeed, the MSPRT's selection is stable even in the low-acceleration regions, and model switches



(a) Friction coefficient profile.



(b) Pedal and vehicle speed profiles.

FIGURE 8. (a) Road friction estimation with the WTA and MSPRT model selection algorithms, on a test scenario. Top: friction coefficient profile. Middle: measured acceleration a_x and predictions $\widehat{a}_{x_i}$, $i = 1, \dots, \mathcal{Q}$, of the $\mathcal{Q} = 4$ pre-trained MS-NNs, where $i = 1$ is dry asphalt, $i = 2$ is wet asphalt, $i = 3$ is snowy asphalt, and $i = 4$ is icy asphalt. Bottom: index i of the selected model at each time step, $i = 1, \dots, \mathcal{Q}$, with the WTA and MSPRT algorithms. (b) Pedal and vehicle speed profiles used in the test scenario.

occur only when the accumulated likelihood of the selected model exceeds a threshold value (Section C). In the two hard-braking events of Fig. 8a, the MSPRT correctly selects the models whose predictions are closest to the measured acceleration: the snowy asphalt model in the first event (around $t = 40$ s), and the wet asphalt model in the second event (around $t = 90$ s).

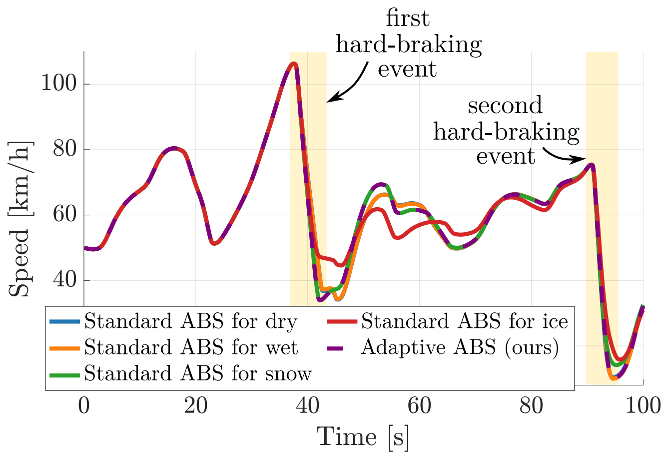
These results show that the MSPRT algorithm is less sensitive to measurement noise than the WTA. By computing the likelihood over a time window, the MSPRT accumulates evidence before making a decision, which avoids frequent noise-induced model switches and provides a more stable road friction estimation. This confirms the findings of [51], [52], which recently applied the MSPRT for noise-robust maneuver selection in autonomous driving agents. Indeed, the MSPRT was shown to yield the shortest decision time in noisy multi-alternative scenarios [48], for given error rates and decision thresholds. Finally, it is worth mentioning that the MSPRT is strongly related to biological decision-making processes, and in particular to the way actions are selected in the basal ganglia of the human brain [50]. This further motivates the use of the MSPRT in autonomous decision-making and model selection tasks, like the road friction estimation in our adaptive ABS system.

The following subsection shows how the adaptation of the ABS parameters based on the road friction estimation affects the vehicle's braking performance.

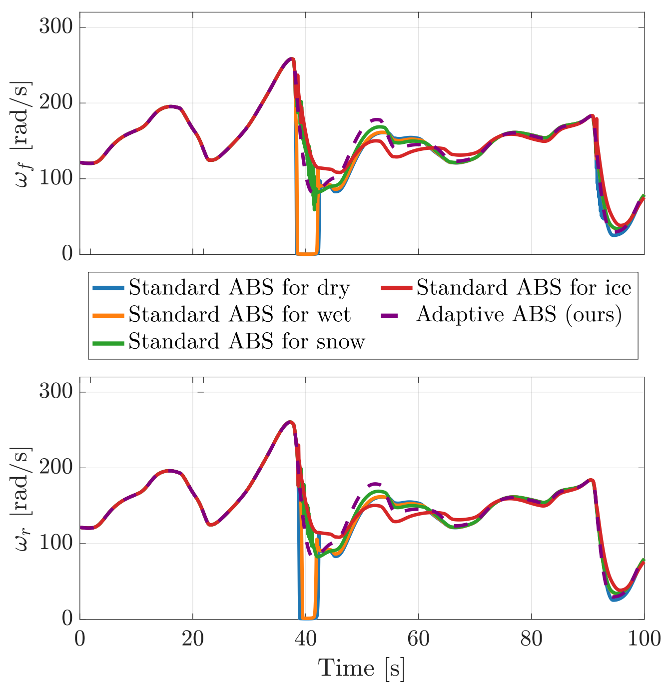
2) ABS with Adaptive Parameters versus Standard ABS

By exploiting the road friction estimation outlined in the previous section, we perform online adaptation of the ABS parameters. Specifically, we adapt the threshold values of the ABS system according to the road condition, as described in Section C.

Fig. 9 compares the performance of our adaptive ABS against standard ABS systems, which we use as benchmarks. The standard ABS has a four-state control logic [1, Chapter 4] like our adaptive system (Section A), but the standard ABS is fine-tuned for single road conditions, and its parameters are not adapted in real-time. Specifically, the standard ABS systems are tuned for dry, wet, snowy, and icy asphalt, and we use the corresponding parameters reported in Table 3. Note that these ABS parameters are the same as our adaptive system in the single road conditions, but they are kept fixed during operation. The test scenario is the same as in Fig. 8: the vehicle experiences varying road friction conditions and goes through two hard-braking events. As shown in Fig. 9, our adaptive (friction-aware) ABS outperforms the standard ABS in both braking events: the vehicle reaches the lowest speeds (Fig. 9a) and wheel-lock never occurs (Fig. 9b). Conversely, the standard ABS systems tuned for single road conditions (dry, wet, snow, ice) degrade the vehicle's braking performance in non-nominal road conditions. For example, both the standard

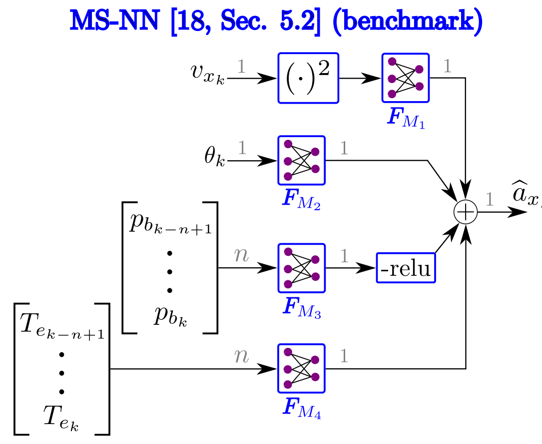


(a)

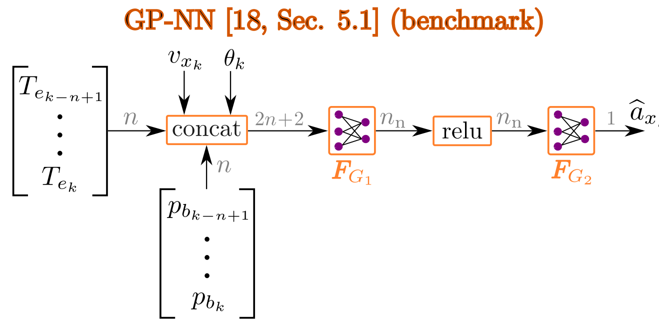


(b)

FIGURE 9. Comparison of our adaptive and standard ABS (benchmark) systems in the test scenario of Fig. 8. (a) Vehicle speed v_x profiles and (b) front and rear wheel rates $\{\omega_f, \omega_r\}$, during the two hard-braking events. The standard ABS systems are tuned for single road conditions (dry, wet, snow, ice), while our adaptive friction-aware ABS adapts its parameters in real-time according to the road friction estimation. Our adaptive ABS outperforms the standard ABS, avoiding wheel-lock and achieving the lowest vehicle speeds during braking.



(a)



(b)

FIGURE 10. Architectures of the neural network benchmarks that we use for comparison with our MS-NN: (a) the MS-NN derived from [18, Section 5.2]; (b) a general-purpose feedforward NN (GP-NN) derived from [18, Section 5.1]. The numbers in gray over the arrows represent the data size of the corresponding signals.

ABS systems tuned for dry and wet conditions (blue and orange lines) lock the wheels in the first braking event, when the road is near-snowy.

Quantitatively, we compare the adaptive and standard ABS in terms of braking distances and maximum peak values of the longitudinal wheel slips. Table 4 reports the main results: our friction-aware ABS achieves the lowest braking distances in both braking events, with an improvement ranging in $3.0\% - 40.4\%$ compared to the standard ABS systems. More precisely, our test scenario (Fig. 8 and 9) involves braking on near-snowy and near-wet asphalt: the standard ABS tuned for a snowy road performs the best among the standard ABS systems, but our friction-aware ABS still achieves a 3.0% improvement in the total braking distance.

Also, our adaptive ABS keeps the maximum wheel slip below a safe threshold of 0.27 in all cases.

Thus, the proposed adaptive ABS significantly decreases the braking distances in hard-braking maneuvers, enhancing the safety and stability of the vehicle in variable non-nominal road conditions.

D. MS-NN versus Literature Benchmarks

This section compares the performance of our novel MS-NN (Section III) with the following benchmarks:

- Another MS-NN, derived from the one of [18, Section 5.2] and shown in Fig. 10a.
- A NN with a general-purpose internal structure (GP-NN), derived from [18, Section 5.1], and shown in Fig. 10b.

IEEE Open Journal of Intelligent Transportation Systems

ABS type	**Braking distance** [m]			**Max tire slip** in braking [-]
	First braking event (Fig. 9a)	Second braking event (Fig. 9a)	Total braking distance	
Standard ABS tuned for dry	82.12	24.91	107.03	0.99
Standard ABS tuned for wet	81.83	25.87	107.70	0.99
Standard ABS tuned for snow	68.59	28.79	97.38	0.24
Standard ABS tuned for ice	121.95	36.65	158.60	0.19
Adaptive friction-aware ABS (ours)	68.59	25.84	94.43	0.27

TABLE 4. Braking distances and maximum peak values of the longitudinal wheel slips for the adaptive friction-aware and standard ABS (benchmark) systems, in the test scenario of Fig. 8 and 9. Our friction-aware ABS achieves the lowest braking distances in the two braking events shown in Fig. 9a, in different road adherence conditions, and keeps the maximum wheel slip below a safe threshold of 0.27.

Both benchmarks have the same inputs and outputs as our MS-NN, and are trained with the same datasets. Let us now briefly describe the architectures of the two benchmarks.

MS-NN Benchmark

The MS-NN benchmark in Fig. 10a is derived from [18, Section 5.2], excluding the selected gear as input. The structure of this NN preserves some prior knowledge of the vehicle dynamics, such as the additivity of forces in the Newton's law (final summation in Fig. 10a) and the quadratic dependency of the aerodynamic drag force on the speed (quadratic layer in the top branch of Fig. 10a). The learnable parameters are in the fully connected layers $\boldsymbol{F}_{M_i}$, $i \in 1, \ldots, 4$.

GP-NN Benchmark

The GP-NN benchmark in Fig. 10b is a general-purpose two-layer feedforward NN, derived from the one of [18, Section 5.1]. All the inputs are concatenated along the first dimension (`concat` layer in Fig. 10b), and are then processed by the two fully connected layers $\boldsymbol{F}_{G_1}$ (with $n_{\mathrm{n}} = 10$ neurons and a relu activation function) and $\boldsymbol{F}_{G_2}$ (with 1 neuron and a linear activation function). This NN does not incorporate any prior knowledge of the vehicle dynamics, and relies on a black-box approach to model the vehicle's longitudinal dynamics.

Benchmarking and Discussion

Fig. 11 compares the predictions of our MS-NN with the two benchmarks, for the dry, wet, snowy, and icy road scenarios. Table 5 reports the root mean square error (RMSE) and fraction of variance unexplained[2] (FVU) for the most extreme cases of dry and icy asphalt. The proposed MS-NN (green line in Fig. 11) yields the most accurate predictions in all cases: the performance gain is $14.9\% - 55.1\%$ in terms of RMSE and $25.7\% - 63.3\%$ in terms of FVU, compared to the benchmarks. The GP-NN benchmark provides a noisier and slightly less accurate prediction, while the MS-NN benchmark's accuracy degrades in certain scenarios, like the dry and snowy road cases. Indeed, the MS-NN benchmark of [18, Section 5.2] was not conceived to capture the vehicle dynamics in the presence of hard-braking maneuvers, but rather to model the dynamics in normal driving conditions.

[2]The FVU indicates the fraction of the total variance in the observed data that remains unexplained by the model (the lower the better). The FVU is computed as $\mathrm{var}(a_x - \widehat{a}_x)/\mathrm{var}(a_x)$, where $\mathrm{var}(\cdot)$ is the variance operator.

Model name	RMSE [m/s^2]		FVU [-]	
	Dry	Ice	Dry	Ice
MS-NN [18, Sec. 5.2]	0.711	0.524	0.177	0.343
GP-NN [18, Sec 5.1]	0.375	0.455	0.098	0.260
Our MS-NN	0.319	0.396	0.065	0.193

TABLE 5. Comparison of the RMSE and FVU metrics for our MS-NN and the two literature benchmarks from [18], in the dry and icy road scenarios. Our MS-NN achieves the lowest RMSE and FVU values, indicating better accuracy and better modeling of the variance in the telemetry data.

Despite training on only 6 minutes of driving data, our MS-NN achieves good generalization and sample-efficiency, making it well-suited for real-world applications where the training data is limited. Additionally, the small size of the MS-NN makes it ideal for deployment on embedded systems with limited computational resources. In fact, a `C++` implementation of the MS-NN runs in less than 20 µs on the 2.6 GHz Intel Core i7 CPU used in our simulations, which means that the MS-NN can be deployed on lower-cost hardware and still provide real-time performance.

VII. Conclusions and Future Work

This paper presented a novel friction-aware ABS framework, which autonomously adapts the ABS parameters according to the estimated road friction conditions. Our method is based on original model-structured neural networks (MS-NNs) to learn the vehicle longitudinal dynamics from limited driving data, and uses a robust model selection algorithm to estimate the road friction in real-time.

We trained four MS-NNs on datasets collected in four corresponding road conditions, and used them for real-time friction estimation with the MSPRT model selection algorithm. We tuned the ABS thresholds for each road condition, and then adapted the ABS parameters online based on the estimated road friction.

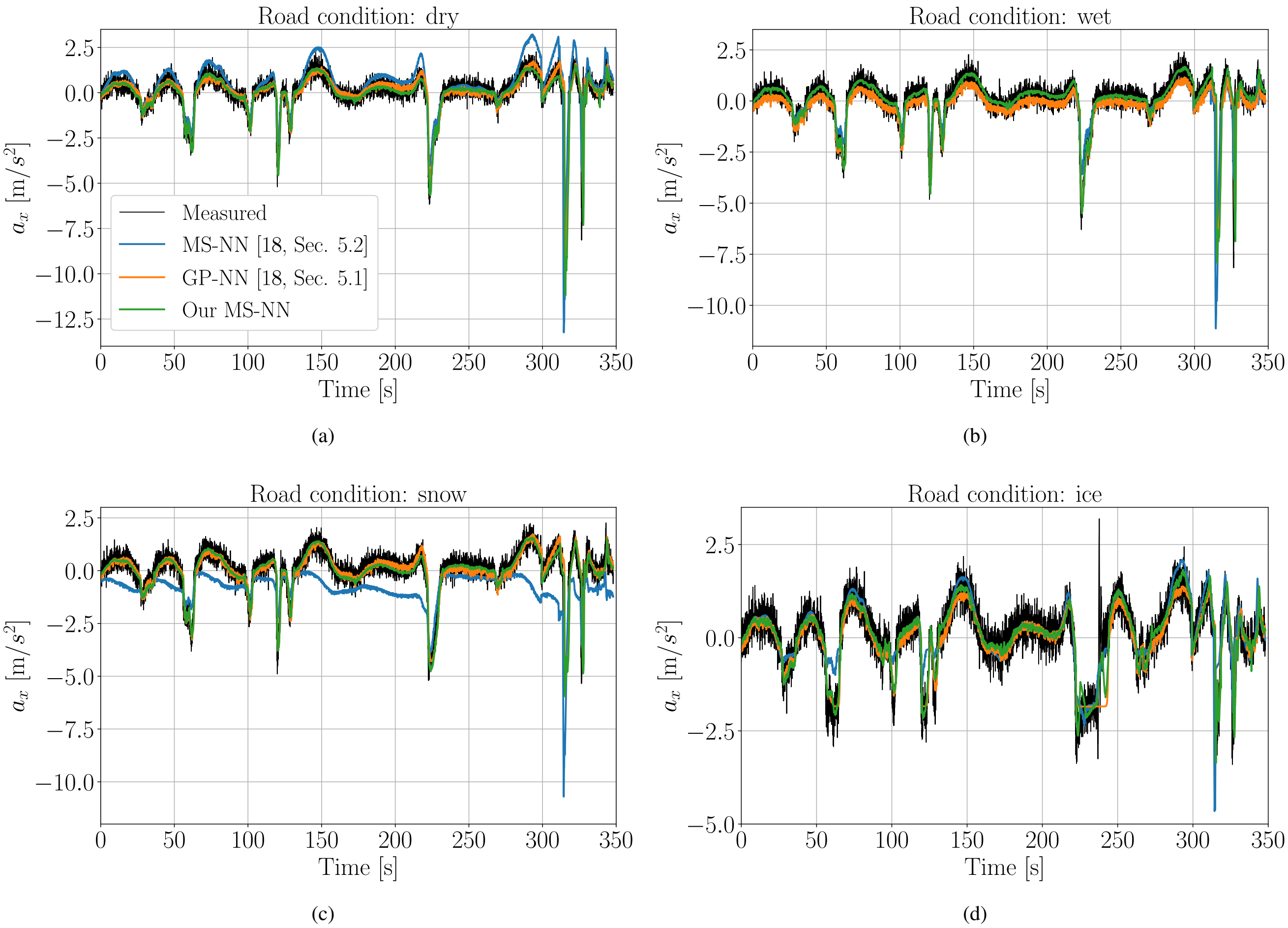


FIGURE 11. Comparing the measured and predicted longitudinal accelerations of our MS-NN and the two benchmarks from [18], for the dry, wet, snowy, and icy road conditions. The proposed MS-NN (green line) achieves the lowest RMSE and FVU values in all the scenarios.

Our friction-aware ABS was tested in challenging simulated scenarios with hard-braking maneuvers on varying non-nominal road conditions. The proposed adaptive ABS outperformed conventional ABS systems tuned for single road conditions, achieving the lowest braking distances ($3.0\% - 40.4\%$ improvement) and avoiding wheel-lock in all scenarios. The novel MS-NNs were compared with two literature benchmarks, showing better accuracy and generalization with limited training data. Finally, the MSPRT model selection criterion outperformed the WTA benchmark in terms of noise robustness and model switching stability.

Future work will extend the proposed approach to lateral stability control systems, and will deploy the adaptive ABS framework on a Raspberry Pi embedded microcontroller, for real-time testing on the small-scale vehicle prototype used in [53].

REFERENCES

[1] S. Savaresi and M. Tanelli, *Active Braking Control Systems Design for Vehicles*. Springer, 01 2011.

[2] S. Kuutti, R. Bowden, Y. Jin, P. Barber, and S. Fallah, "A survey of deep learning applications to autonomous vehicle control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 712–733, 2021.

[3] J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi, and R. Mangharam, "Autonomous vehicles on the edge: A survey on autonomous vehicle racing," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 458–488, 2022.

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

[5] M. Abe, *Vehicle Handling Dynamics: Theory and Application*. Elsevier Science, 2009.

[6] H. Pacejka, *Tire and Vehicle Dynamics Ed. 3*. Elsevier Science, 2012.

[7] G. Devineau, P. Polack, F. Altché, and F. Moutarde, "Coupled longitudinal and lateral control of a vehicle using deep learning," in *2018 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 11 2018, pp. 642–649.

[8] 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 System Dynamics*, vol. 61, no. 1, pp. 83–110, 2023.

[9] N. A. Spielberg, M. Brown, N. R. Kapania, J. C. Kegelman, and J. C. Gerdes, "Neural network vehicle models for high-performance automated driving," *Science Robotics*, vol. 4, no. 28, p. eaaw1975, 2019.

[10] L. Hermansdorfer, R. Trauth, J. Betz, and M. Lienkamp, "End-to-end neural network for vehicle dynamics modeling," in *2020 6th IEEE Congress on Information Science and Technology (CiSt)*, 2020, pp. 407–412.

[11] S. Gottschalk, M. Gerdts, and M. Piccinini, "Reinforcement learning and optimal control: A hybrid collision avoidance approach," in *Proceedings of the 10th International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS*, INSTICC. SciTePress, 2024, pp. 76–87.

[12] T. Weiss and M. Behl, "Deepracing: A framework for autonomous racing," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2020, pp. 1163–1168.

[13] S. Löckel, J. Peters, and P. van Vliet, "A probabilistic framework for imitating human race driver behavior," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2086–2093, 2020.

[14] S. Löckel, S. Ju, M. Schaller, P. van Vliet, and J. Peters, "An adaptive human driver model for realistic race car simulations," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 11, pp. 6718–6730, 2023.

[15] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, "A survey on neural network interpretability," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 5, pp. 726–742, 2021.

[16] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[17] H. Lim, J. W. Lee, J. Boyack, and J. B. Choi, "Ev-pinn: A physics-informed neural network for predicting electric vehicle dynamics," 2024. [Online]. Available: https://arxiv.org/abs/2411.14691

[18] D. B. Mauro Da Lio and G. P. R. Papini, "Modelling longitudinal vehicle dynamics with neural networks," *Vehicle System Dynamics*, vol. 58, no. 11, pp. 1675–1693, 2020.

[19] 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. 192 041–192 064, 2020.

[20] 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. 46 344–46 372, 2023.

[21] M. Piccinini, S. Taddei, E. Pagot, E. Bertolazzi, and F. Biral, "How optimal is the minimum-time manoeuvre of an artificial race driver?" *Vehicle System Dynamics*, vol. 0, no. 0, pp. 1–28, 2024.

[22] M. Piccinini, S. Taddei, M. Piazza, and F. Biral, "Impacts of g-g-v constraints formulations on online minimum-time vehicle trajectory planning," *IFAC-PapersOnLine*, vol. 58, no. 10, pp. 87–93, 2024, 17th IFAC Symposium on Control of Transportation Systems CTS 2024.

[23] J. Chrosniak, J. Ning, and M. Behl, "Deep dynamics: Vehicle dynamics modeling with a physics-constrained neural network for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5292–5297, 2024.

[24] J. Wegrzynowski, G. Czechmanowski, P. Kicki, and K. Walas, "Learning dynamics models for velocity estimation in autonomous racing," 2024. [Online]. Available: https://arxiv.org/abs/2408.15610

[25] T. Kim, H. Lee, and W. Lee, "Physics embedded neural network vehicle model and applications in risk-aware autonomous driving using latent features," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 4182–4189.

[26] E. Pagot, M. Piccinini, E. Bertolazzi, and F. Biral, "Fast planning and tracking of complex autonomous parking maneuvers with optimal control and pseudo-neural networks," *IEEE Access*, vol. 11, pp. 124 163–124 180, 2023.

[27] M. Piccinini, S. Gottschalk, M. Gerdts, and F. Biral, "Computationally efficient minimum-time motion primitives for vehicle trajectory planning," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 5, pp. 642–655, 2024.

[28] M. D. Lio, M. Piccinini, and F. Biral, "Robust and sample-efficient estimation of vehicle lateral velocity using neural networks with explainable structure informed by kinematic principles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 12, pp. 13 670–13 684, 2023.

[29] S. A. Faroughi, N. M. Pawar, C. Fernandes, M. Raissi, S. Das, N. K. Kalantari, and S. Kourosh Mahjour, "Physics-guided, physics-informed, and physics-encoded neural networks and operators in scientific computing: Fluid and solid mechanics," *Journal of Computing and Information Science in Engineering*, vol. 24, no. 4, p. 040802, 01 2024.

[30] M. Acosta, S. Kanarachos, and M. Blundell, "Road friction virtual sensing: A review of estimation techniques with emphasis on low excitation approaches," *Applied Sciences*, vol. 7, no. 12, 2017.

[31] M. Khaleghian, A. Emami, and S. Taheri, "A technical survey on tire-road friction estimation," *Friction*, vol. 5, 05 2017.

[32] L. Hermansdorfer, J. Betz, and M. Lienkamp, "A concept for estimation and prediction of the tire-road friction potential for an autonomous racecar," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1490–1495.

[33] M. Acosta, S. Kanarachos, and M. E. Fitzpatrick, "Optimized tire force estimation using extended kalman filter and fruit fly optimization," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, pp. 4074–4079.

[34] 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, 2018.

[35] F. Holzmann, M. Bellino, R. Siegwart, and H. Bubb, "Predictive estimation of the road-tire friction coefficient," in *2006 IEEE Conference on Computer Aided Control System Design*, 2006, pp. 885–890.

[36] J.-A. Hsu, C.-K. Chen, and X.-D. Zhang, "Hardware-in-the-loop simulations and experiments of anti-lock braking system for cornering motorcycles," *Electronics*, vol. 13, no. 21, 2024.

[37] F. Pretagostini, L. Ferranti, G. Berardo, V. Ivanov, and B. Shyrokau, "Survey on wheel slip control design strategies, evaluation and application to antilock braking systems," *IEEE Access*, vol. 8, pp. 10 951–10 970, 2020.

[38] X. Chen, Z. Dai, H. Lin, Y. Qiu, and X. Liang, "Asymmetric barrier lyapunov function-based wheel slip control for antilock braking system," *International Journal of Aerospace Engineering*, vol. 2015, no. 1, p. 917807, 2015.

[39] J. Guo, X. Jian, and G. Lin, "Performance evaluation of an anti-lock braking system for electric vehicles with a fuzzy sliding mode controller," *Energies*, vol. 7, no. 10, pp. 6459–6476, 2014.

[40] E. Chereji, M.-B. Radac, and A.-I. Szedlak-Stinean, "Sliding mode control algorithms for anti-lock braking systems with performance comparisons," *Algorithms*, vol. 14, no. 1, 2021.

[41] S. Santini, N. Albarella, V. M. Arricale, R. Brancati, and A. Sakhnevych, "On-board road friction estimation technique for autonomous driving vehicle-following maneuvers," *Applied Sciences*, vol. 11, no. 5, 2021.

[42] R. Bhandari, S. Patil, and R. K. Singh, "Surface prediction and control algorithms for anti-lock brake system," *Transportation Research Part C: Emerging Technologies*, vol. 21, no. 1, pp. 181–195, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X11001276

[43] G. Audisio *et al.*, "Cyber tyre for vehicle active safety," in *Proc. XIX AIMETA*, 2009, pp. 14–17.

[44] K. B. Singh and S. Taheri, "Estimation of tire-road friction coefficient and its application in chassis control systems," *Systems Science & Control Engineering*, vol. 3, no. 1, pp. 39–61, 2015.

[45] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[46] X. Nie, C. Min, Y. Pan, K. Li, and Z. Li, "Deep-neural-network-based modelling of longitudinal-lateral dynamics to predict the vehicle states for autonomous driving," *Sensors*, vol. 22, no. 5, 2022.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017. [Online]. Available: https://arxiv.org/abs/1412.6980

[48] V. Draglia, A. Tartakovsky, and V. Veeravalli, "Multihypothesis sequential probability ratio tests—part i: Asymptotic optimality," *IEEE Transactions on Information Theory*, vol. 45, no. 7, pp. 2448–2461, 1999.

[49] C. Baum and V. Veeravalli, "A sequential procedure for multihypothesis testing," *IEEE Transactions on Information Theory*, vol. 40, no. 6, pp. 1994–2007, 1994.

[50] R. Bogacz and K. Gurney, "The basal ganglia and cortex implement optimal decision making between alternative actions," *Neural computation*, vol. 19, pp. 442–77, 03 2007.

[51] M. D. Lio, R. Donà, G. P. R. Papini, and K. Gurney, "Agent architecture for adaptive behaviors in autonomous driving," *IEEE Access*, vol. 8, pp. 154 906–154 923, 2020.

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2025.3563347

[52] G. Valenti, E. Pagot, L. De Pascali, and F. Biral, "Battery aging-aware online optimal control: An energy management system for hybrid electric vehicles supported by a bio-inspired velocity prediction," *IEEE Access*, vol. 9, pp. 164 394–164 416, 2021.

[53] 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 *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2390–2396.

MATTIA PICCININI (Member, IEEE) is a TUM Global Post-doctoral Fellow in the Autonomous Vehicle Systems research group at the Technical University of Munich, Germany. He received an M.Sc. (cum laude) in mechatronics engineering and a Ph.D. (cum laude) in autonomous systems from the University of Trento, Italy, in 2019 and 2024 respectively. From March to June 2022, he was a visiting Ph.D. student at the Universität der Bundeswehr, Munich, Germany.

His research focuses on trajectory planning, control and state estimation methods for urban and racing autonomous vehicles.

MATTEO ZUMERLE received a Bachelor's degree in Industrial Engineering in 2018 and a Master's degree in Mechatronics Engineering on Intelligent Vehicles in 2024, both from the University of Trento, Italy. His interests include robotics, automation systems, and intelligent control strategies, particularly those related to vehicle dynamics.

JOHANNES BETZ is an assistant professor in the Department of Mobility Systems Engineering at the Technical University of Munich (TUM), where he is leading the Autonomous Vehicle Systems (AVS) lab. He is one of the founders of the TUM Autonomous Motorsport team. His research focuses on developing adaptive dynamic path planning and control algorithms, decision-making algorithms that work under high uncertainty in multi-agent environments, and validating the algorithms on real-world robotic systems. Johannes earned a B.Eng. (2011) from the University of Applied Science Coburg, an M.Sc. (2012) from the University of Bayreuth, an M.A. (2021) in philosophy from TUM, and a Ph.D. (2019) from TUM.

GASTONE PIETRO ROSATI PAPINI graduated in Automation Engineering at the University of Pisa in 2012 and earned his Ph.D. at Sant'Anna School of Advanced Studies in Emerging Digital Technologies in 2016. He is currently an Associate Professor in advanced control systems applied to the field of autonomous driving and ADAS at the Department of Industrial Engineering, University of Trento. He is also the Co-Founder of Cheros s.r.l., a spinout company from the University of Pisa that specializes in renewable energies and ICT solutions. With over five years of experience in mechanics and control systems, he has recently focused on the application of machine learning techniques for modeling and controlling mechanical systems. Webpage: https://tonegas.com/