IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 4, 2020, accepted April 11, 2020, date of publication April 17, 2020, date of current version May 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2988592

Longitudinal Vehicle Dynamics: A Comparison of Physical and Data-Driven Models Under Large-Scale Real-World Driving Conditions

SEBASTIAN S. JAMES[1], SEAN R. ANDERSON[1], AND MAURO DA LIO[2], (Member, IEEE)
[1]Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield S1 3JD, U.K.
[2]Department of Industrial Engineering, University of Trento, 38123 Trento, Italy

Corresponding author: Sebastian S. James (seb.james@sheffield.ac.uk)

This work was supported by the EU under H2020 Grant 731593 (Dreams4Cars).

ABSTRACT Mathematical models of vehicle dynamics will form essential components of future autonomous vehicles. They may be used within inverse or forward control loops, or within predictive learning systems. Often, nonlinear physical models are used in this context, which, though conceptually simple (especially for decoupled, longitudinal dynamics), may be computationally costly to parameterise and also inaccurate if they omit vehicle-specific dynamics. In this study we sought to determine the relative merits of a commonly used nonlinear physical model of vehicle dynamics versus data-driven models in large-scale real-world driving conditions. To this end, we compared the performance of a standard nonlinear physical model with a linear state-space model and a neural network model. The large-scale experimental data was obtained from two vehicles; a Lancia Delta car and a Jeep Renegade sport utility vehicle. The vehicles were driven on regular, public roads, during normal human driving, across a range of road gradients. Both data-driven models outperformed the physical model. The neural network model performed best for both vehicles; the state-space model performed almost as well as the neural network for the Lancia Delta, but fell short for the Jeep Renegade whose dynamics were more strongly nonlinear. Our results suggest that the linear data-driven model gives a good trade-off in accuracy and simplicity, whilst the neural network model is most accurate and is extensible to more nonlinear operating conditions, and finally that the widely used physical model may not be the best choice for control design.

INDEX TERMS Data-driven, state estimation, linear, neural network, nonlinear, state-space, vehicle dynamics.

I. INTRODUCTION

Driver assistance systems such as cruise control, adaptive cruise control [1], [2], vehicle platooning schemes [3]–[5] and future driverless cars [6]–[9] all depend on mathematical models of vehicle dynamics. Physically derived models are commonly employed for these applications; examples can be found for lane change manoeuvres [10], [11], lane keeping [12]–[14] and cruise control [2]–[5]. In addition, the development of low-level longitudinal and lateral vehicle control algorithms, e.g. using model predictive control (MPC), has been based on physically derived models, either fully nonlinear, e.g., [15], or based on linearised time-varying models, e.g., [14], [16]. This paper focuses on the development of vehicle dynamics models for control design, where the key objective is model parsimony, i.e. the model should be no more complicated than needed for the purpose of control design.

The approach of using physical models is attractive because they tend to be interpretable and well-understood. Physical models are also low-cost if data-collection and parameter estimation can be avoided (i.e. if the parameters can be defined from prior knowledge). However, the use of physical models for control design requires significant human designer intervention at both the modelling and implementation stages and suffers from:

The associate editor coordinating the review of this manuscript and approving it for publication was Lin Zhang.

- only containing the assumptions of the human-modeller, not the full range of dynamics expressed through the data;
- a tendency to be rigidly defined whereas data-driven models can be adapted over the life of the vehicle, accounting for e.g. degradation of parts and changing operational conditions (tyres, road surface and weather);
- being overly-complicated versus typically parsimonious data-driven model structures;
- parameters typically being difficult and time consuming to estimate due to lack of knowledge for good initialisation in nonlinear optimisation algorithms [10].

At the modelling stage, decisions about which phenomena should be modelled and which can be neglected are taken. It may be thought that a particular, nonlinear physical phenomenon will have a negligible effect on the vehicle dynamics and so the terms describing that phenomenon are omitted from the formulation of the model. Without the benefit of hindsight it is not realistic to expect human designers to account for every effect. We must assume that on occasion, the modeller will make the wrong decision about what should be included in the model. On these occasions, the physical model is likely to break down.

Even physical models containing few unknown parameters can be computationally expensive to parameterise. Additionally, the cost of data collection must be borne for each model [17]; the parameters for one vehicle will not match those for another; drag properties, structural stiffness and tyre properties will all affect the dynamics of the vehicle and may not be available without a parameter search. It is not unknown for the parameters from several different vehicles to be combined to approximate the model for another vehicle [10].

Some of these issues can be alleviated by the linearisation of the physical model, an approach taken by [16], [18]. This approach is possible because at typical legal highway speeds and in typical traffic conditions, neither air drag, nor tyre slip are strongly nonlinear [19].

The alternative approach of data-driven learning of the vehicle forward dynamics model resolves many of the issues encountered with physical models. This may be as simple as learning a suitable linear state-space representation [20], combining local, linear models [21], [22] or more complex nonlinear approaches based on neural networks [23]–[26]. An advantage of this data-driven approach is that, if data collection continues during operation of the vehicle, multiple models for different operational conditions can be automatically generated offline (following an approach that mimics the human wake-sleep stages in the learning of motor control [27]). Other authors have discussed the online adaptation of the vehicle model in a less brain-inspired manner [20], [21].

In this study, we focus on modelling longitudinal vehicle dynamics in the range of conditions that are realised by ordinary human drivers, as better delimited in Section II.A; specifically on the longitudinal vehicle velocity. Longitudinal dynamics are generally assumed to be dominated by longitudinal forces. This assumption is widely made, often without being explicitly stated (see, for example [20], a study of cruise control). We make the same assumption that the longitudinal and lateral dynamics of the vehicle (in ordinary human driving) are weakly coupled. Indeed, an analysis of the data presented in this study for longitudinal-lateral dynamics indicates that the coupling is very weak, and that the longitudinal data-driven models are not improved by inclusion of lateral input. We thus consider physical and data-driven models whose only inputs are longitudinal and which do not take account of steering wheel position or vehicle yaw rate.

The main contribution of this paper is thus to make a comparative evaluation of longitudinal vehicle dynamics between physical and data-driven models, using experimental data collected across large-scale real-world driving (the 'ordinary domain') using two different vehicles. This work extends a recent conference paper by two of the authors into linear versus nonlinear modelling of longitudinal vehicle dynamics [28].

For the traditional physical modelling approach we employed a standard model based on the longitudinal component of the well-known bicycle model of vehicle dynamics [29], i.e. based on Newton's second law of motion with forces due to engine propulsion, braking, air drag, static friction and the effect of gravity on inclined roads. For data-driven approaches a linear state-space model was identified using a prediction error method (PEM) with initialisation by subspace identification [30], and a neural network model was also identified [31], based on capturing the nonlinear impulse response due to the propulsion, braking and road gradient inputs [26]. To identify the models, we used experimental data that were collected using a typical medium-sized car and a sport utility vehicle driven along a varied urban and extra-urban route.

There are few existing modelling studies that identify car vehicle dynamics from sampled data but one investigation that uses step responses found that a linear second order model was sufficient to accurately describe the car dynamics during single manoeuvres of about 10 seconds duration [1]. That modelling study therefore agrees with the finding here that linear models are sufficient to describe vehicle dynamics for some vehicles.

Our results are comparable with some of those reported by Hunt *et al.* [21] in a similar study based on empirical data for a commercial lorry. In that work an emphasis was placed on using local linear models, with the gear, throttle position and speed variously used to schedule the different models. The model closest to the state-space model reported here was their *M3*, in which a linear first order model was determined for each gear, which was of a comparably high accuracy to our linear model.

However, the data in that study was collected on a flat road, the vehicle's brakes were never applied and it did not include road gradient, which were key extensions here for studying real-world driving conditions.

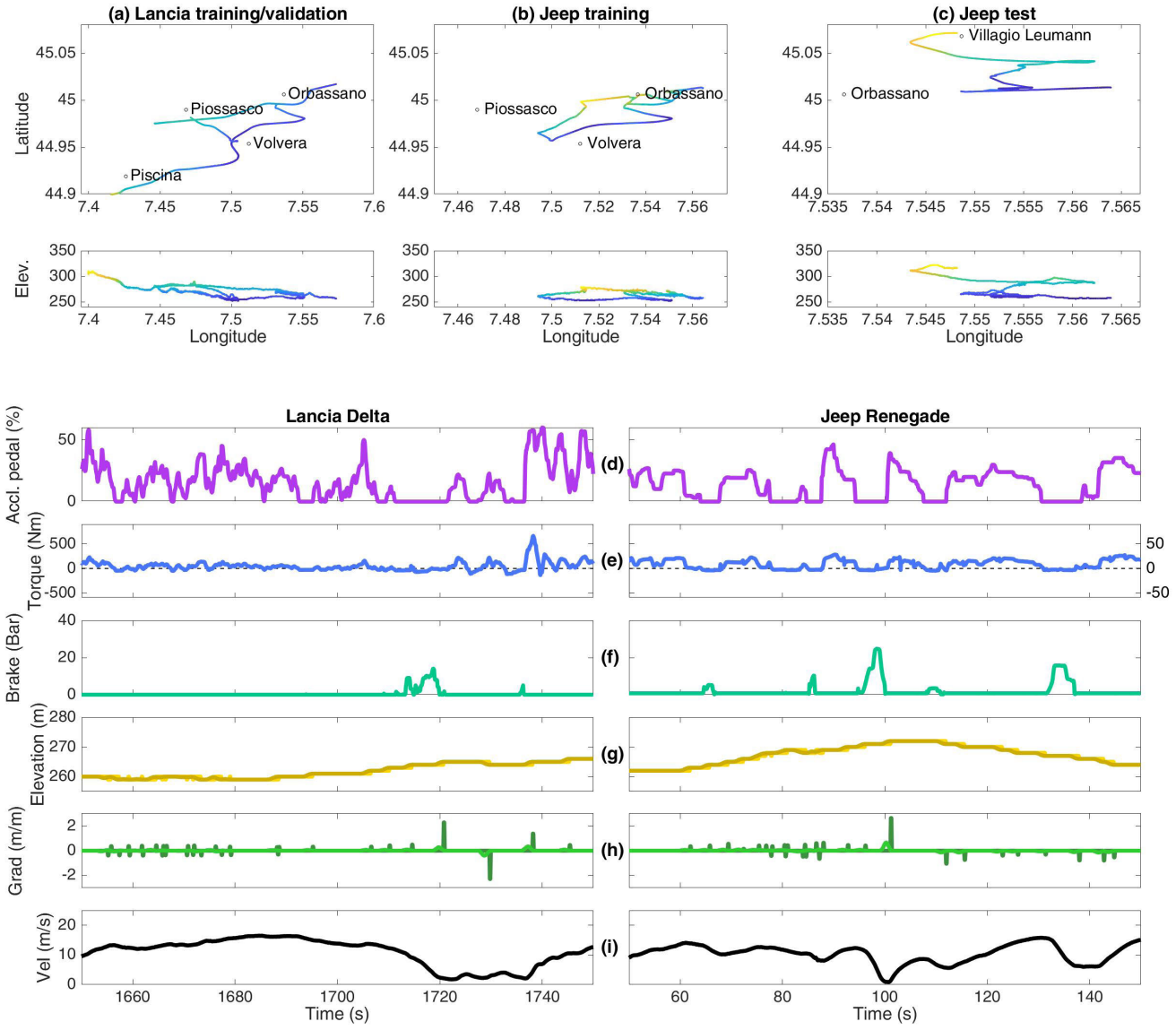

FIGURE 1. Training and test routes from the experiments conducted near Turin, Italy on public roads in normal driving conditions, and example data over 100 seconds. (a) The training loop around which the Lancia Delta vehicle was driven. Elevation is given by the colour and plotted against longitude in the lower graph. (b) Jeep Renegade training route (c) Jeep Renegade test route. (d-i) Signals provided by the engine control unit. Lancia examples are in the left column, Jeep examples in the right. (d) Accelerator pedal position, percent of maximum. (e) Engine torque, inclusive of engine friction. Note that the torque signal provided by the Jeep's ECU is shown on a different vertical scale than that for the Lancia due to an additional gearing being incorporated into this signal. (f) Brake master cylinder pressure (bar). (g) Elevation in meters (h) Gradient. Note that the plots of elevation and road gradient show the raw data and the smoothed data obtained by convolving with a Gaussian function. (i) Longitudinal velocity.

II. METHODS

A. EXPERIMENTAL DATA

Two vehicles were used for experimental data collection on public roads in normal conditions. A Lancia Delta car (compact 5 door hatchback) was used to collect the first set of data using a 53 km route. An additional dataset was collected using a Jeep Renegade sport utility vehicle (SUV) along a similar route in the same region near Turin, Italy. The routes incorporated a typical selection of motorways, extra-urban and urban roads, roundabouts and intersections. The first route [Lancia Delta car—Fig. 1(a)] began at Centro Ricerche Fiat in Orbassano, near Turin (Italy), then went to Pinerolo, then Piossasco before returning to Orbassano, a distance of 53 km driven in about 42 minutes, and is the same data as described in [8] and as used in our previous work [28]. The Jeep Renegade was driven along a loop of approximately 25 km similar to the first route [using some of the same roads—Fig. 1(b)]; and a different test route, of approximately 15 km, from Centro Ricerche Fiat to Cascine Vica [Fig. 1(c)]. Both routes provide data within the 'ordinary driving domain', with velocities, speeds and in particular accelerations within the ranges discussed in [32]. Accordingly, the accelerations used fall amply within the limits of tyre adherence.

In both vehicles, the following signals were recorded: longitudinal velocity, accelerator pedal position, brake pedal position, selected gear, engine torque and GPS co-ordinates. Signals were sampled at 20 Hz. For the physical and state-space models, the road gradient (contributing to the longitudinal acceleration/deceleration) was also obtained from the GPS signal, which was first smoothed by convolving with a Gaussian function (a normal distribution with mean 0 and standard deviation 0.5 s). For the neural network model, the raw elevation data were used, sampled every 10 m in the 200 m range ahead of and behind the vehicle, thus leaving to the network the task of finding the best estimate of the gradient. There was an overall height bias in the GPS height signal collected for each vehicle. We verified that the smoothed gradient signal was consistent between the vehicles by analysing the road gradients collected on the same roads. We found that the mean gradient difference between the road gradients was -0.0015(0.046) m/m whereas the mean elevation difference was -2.7(2.2) m (with standard deviations in brackets).

Each route was circumnavigated twice. The first route provided about 1 hour and 20 minutes of data from which to parameterise and identify the model for the Lancia Delta. We divided this into two sections; one of 2500 s duration for model identification/training [first loop of Fig 1(a)] and another of 2000 s for validation [second circumnavigation]. We also divided the dataset into four sub-datasets of similar length to examine the statistics of models trained on each section. The Jeep Renegade was also driven twice along its two routes, providing about 3200 s of data from the first route and 1600 s from the second in four separate datasets. For the main results, we trained on the first loop of the training route [Fig. 1(b)] and validated on the first loop of the test route [Fig. 1(c)]. The Jeep Renegade data naturally formed four datasets for the statistical analysis. Fig. 1(d)-(i) shows some example control input and the output for part of the training data for each vehicle.

B. PHYSICAL MODEL

The decoupling of the dynamics of a four wheeled vehicle into lateral and longitudinal components is widely practiced [29]. We adopt this simplification and use the resulting longitudinal model which can be written as the Newtonian equation,

$$M\dot{v} = F_p(t) - F_b(t) - Mg\,\sin(\gamma(t)) - k_D\,v^2 - Mg\,k_R \qquad (1)$$

where M is the mass of the vehicle and v is the vehicle's longitudinal velocity, which we will solve for. $F_p(t)$ is the collective propulsive force due to the engine actuating through all of the drive wheels. $F_b(t)$ is the collective braking force. g is the acceleration due to gravity and $\gamma(t)$ is the gradient of the road. $k_D v^2$ is the air drag and k_R is the rolling friction parameter.

It has to be stressed that this equation holds for $v > 0$. Pressure on the brake pedal determines a maximum allowed brake force $F_{b,max}$, which is applied whilst the vehicle is moving. At rest, the actual brake force, $F_b(t)$, is determined by the equilibrium, balancing any other force less than $F_{b,max}$ acting on the vehicle. To avoid the need to model the transition from static to dynamic conditions, we have removed from the training and validation data subsets all portions related to the equilibrium state (that is, any section for which the vehicle's velocity falls below 0.6 m/s [Lancia] or 0.5 m/s [Jeep]). Furthermore, we set $F_b = 0$ and $\gamma = 0$ for $v < 0.5$ m/s to prevent the simulated velocity from becoming negative.

The engine control unit (ECU) provides an engine torque signal which can be used to compute the motive force. The Lancia Delta's ECU provides a requested, pre-gearbox, engine torque signal, T_a, in the range 0 to 400 Nm. The post-gearbox requested torque is obtained by multiplying the engine torque by the gearbox ratio, R: $T_r = T_a.R$. The Lancia ECU also provides a delivered pre-gearbox engine torque signal, T_d, which can be less than the requested torque in the following cases: a) the requested torque is greater than the engine is able to deliver at the current engine speed and, b) the automatic gearbox is controlling a gear shift and is overriding the engine torque control. T_a and T_r do not account for engine friction and the power drawn by engine accessories. The Lancia ECU reports the torque due to friction/accessories in the engine, T_f (which is engine-speed dependent). The delivered engine-shaft torque, T_e, is given by subtraction of T_f:

$$T_e = T_d - T_f \qquad (2)$$

For control system design it is most pertinent to use the requested torque, T_a, and its post-gearbox partner, T_r, as the control system may not have access to the engine-friction signal or any ability to predict when the engine may become unable to deliver the requested torque. However, the most accurate physical model against which to compare alternative models will be given by making use of the engine-shaft torque, T_e, along with the gear ratio information. In our physical modelling, we therefore have F_p given by

$$F_p(t) = \frac{T_e(t)\,R[G(t)]\,\eta_g\,R_d\,\eta_d}{r_w} \qquad (3)$$

where $T_e(t)$ is the delivered engine-shaft torque, $R[G(t)]$ is the gearbox ratio as a function of the gear, $G(t)$, and R_d is the driveline gear ratio. η_g and η_d model the frictional loss in the gearbox and driveline, respectively and r_w is the wheel radius. $R[G(t)]$ is provided by the ECU as a signal, which we combine with T_e to give the pre-loss gear-shaft torque, T_g:

$$T_g(t) = T_e(t)\,R[G(t)] \qquad (4)$$

The ECU of the Jeep Renegade provides $T_g(t)$ directly; that is it provides a torque signal which has the gear ratios and the engine/accessory friction factored in. The (unknown) driveline ratio R_d also appears to be incorporated into the signal T_g on the Jeep, because the magnitude of the signal is roughly a tenth of the T_g signal obtained from the Lancia Delta [Fig. 1(e)].

We collect the terms R_d (unknown), η_d and η_g (unknown; approximately 0.93) and r_w (known for Lancia; 0.292 m)

together into a single parameter, k_τ, so that F_p is, finally,

$$F_p(t) = k_\tau \, T_g(t) \tag{5}$$

with the value of k_τ to be found by the parameter search.

The collective braking force, $F_b(t)$, in the dynamic condition ($v > 0.5$ m/s), is governed by the relationship

$$F_b(t) = \begin{cases} \mu_b \, Mg, & k_b \, p_b(t) > \mu_b \, Mg \\ k_b \, p_b(t), & \text{otherwise} \end{cases} \tag{6}$$

where k_b is a coefficient of braking force in Newtons per bar of brake pressure (189 N/bar [Lancia], 236 N/bar [Jeep]) and $p_b(t)$ is the master brake cylinder pressure. The upper limit on the available braking force, $\mu_b \, Mg$, is the force at which the tyres lose traction with the road surface. μ_b has a complex dependence on the road-surface quality and on tyre conditions (temperature, pressure and wear). However, because our data were collected under normal driving conditions, with no sharp or emergency braking taking place, $F_b(t) < \mu_b \, Mg$ at all times and the first case in Eq.6 was not modelled (meaning that the model would be expected to fail for data collected in slippery conditions).

To estimate the parameters we used a nonlinear interior-point optimisation method [33] applied to the training data, implemented in the MATLAB function *fmincon* (MATLAB version R2017b). A multi-start approach was used to initialise the parameters. 100 different parameter sets, randomly sampled in the ranges $k_D \in [0, 20]$, $k_R \in [0, 0.05]$ and $k_\tau \in [1, 100]$ [Lancia] or $k_D \in [0, 40]$, $k_R \in [0, 0.05]$ and $k_\tau \in [1, 300]$ [Jeep] initialised the parameter estimation. In the work, we found the choice of ODE solver to be of significance. MATLAB's *ode23* solver [34] operated well for 'good' parameters—those which would provide a reasonable fit to the data. However, *ode23* became computationally inefficient towards the boundaries of the chosen parameter ranges. To allow the parameter search to complete quickly, the *ode23s* solver [34] was employed, which is more effective at computing stiff differential equation systems. An instability in the *ode23s* solver produced occasional, random, but quickly corrected deviations from the putative solution of the system for some parameter sets, introducing noise into the simulation and affecting the fit metrics. For this reason, after finding an initial, candidate set of parameter values, we reduced the parameter ranges and re-ran the parameter search using *ode23* initialised with 100 different parameter sets. The Lancia Delta reduced ranges were $k_D \in [0.1, 3]$, $k_R \in [0.0001, 0.04]$ and $k_\tau \in [5, 15]$; The Jeep Renegade reduced ranges were $k_D \in [1, 4]$, $k_R \in [0.000001, 0.005]$ and $k_\tau \in [40, 150]$. Note that *ode23s* was used only during the initial parameter search; *ode23* was used when computing all of the simulation results shown below and all of the reported fit metrics were also generated using *ode23*.

C. LINEAR STATE-SPACE SYSTEM IDENTIFICATION

To perform linear state-space system identification, we used a continuous-time model to represent the longitudinal component of the vehicle dynamics,

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \tag{7}$$

$$y(t) = C\mathbf{x}(t) + D\mathbf{u}(t) \tag{8}$$

where the output $y(t) \in \mathbb{R}$ is the velocity of the vehicle at time t; the input $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ is composed of the delivered post-gearbox engine torque (T_g) and the brake pressure signal (p_b), and in some versions of the model also the smoothed road gradient ($\gamma(t)$); $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ is the vehicle state vector, and n_x is the model order. The model order is determined as part of the identification procedure. The matrices A, B, C and D are assumed fully parameterised and comprise the unknown model parameters.

The state-space model parameters were estimated using a prediction error method (PEM) directly in continuous-time [30]. If we assume all parameters in A, B, C and D are collected in the parameter vector $\boldsymbol{\theta}$, then the estimation problem is defined as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{k=1}^{N} e(t_k, \boldsymbol{\theta})^2 \tag{9}$$

where the prediction error is

$$e(t_k, \boldsymbol{\theta}) = y(t_k) - \hat{y}(t_k, \boldsymbol{\theta}) \tag{10}$$

and $\hat{y}(t_k, \boldsymbol{\theta})$ is the simulated output of the state-space model at sample-time t_k, and N is the number of data samples. The PEM is typically solved using a numerical search algorithm [30].

One problem with the PEM algorithm is parameter initialisation. Here, we initialise the numerical search algorithm using a subspace state-space system identification (N4SID) algorithm [35], [36]. The N4SID algorithm requires a number of choices (insight into these choices is given in [37]), such as the forward prediction horizon and the number of past inputs and outputs that are used for the prediction. In this work, these were chosen by estimating multiple models over a range of values and using Akaike's information criterion (AIC) to select between them. The weighting scheme used in the N4SID algorithm was canonical variate analysis (CVA) [35].

Parameter estimation was performed in MATLAB using the System Identification Toolbox, using the function *ssest*, which conveniently combines the N4SID initialisation of the state-space model with the PEM estimation stage in one function. Key user-choices for the estimation algorithms are summarised in Table 1. The *ssest* function, by default, uses a combination of search methods in the PEM stage including Gauss-Newton (GN), adaptive Gauss-Newton (AGN), Levenberg-Marquardt (LM) and gradient descent (GD). Also, note for initialisation that the N4SID identification was performed in discrete-time, then the model was mapped to continuous-time using a zero-order hold.

To select the model order, n_x, model orders $n_x = 1, \ldots, 10$ were initially tested using an analysis of singular values of the input-output covariance matrix [30], which suggested only

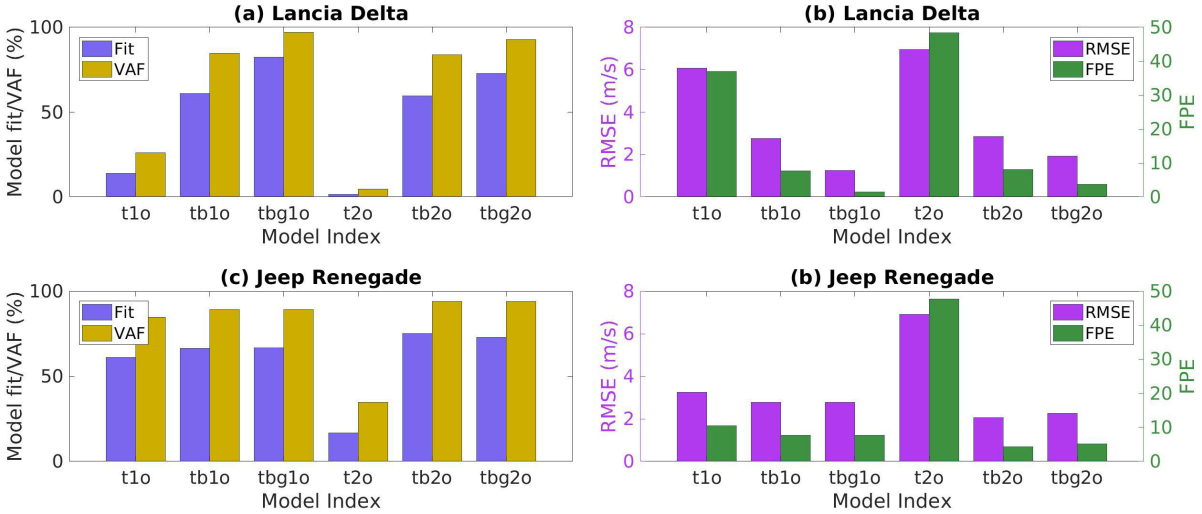

FIGURE 2. Model selection for the linear system identification model, where the model index is as follows: t1o - torque only, first order; tb1o - torque-brake input, first order; tbg1o - torque-brake-gradient input, first order; t2o - torque only, second order; tb2o - torque-brake input, second order; tbg2o - torque-brake-gradient input, second order; (a)-(b) Model selection plots for the Lancia Delta dataset, which indicate that the models with torque-brake-gradient inputs should be preferred. Note that FPE can be misleading because it is based on one-step-ahead prediction errors, unlike the other measures. (c)-(d) Model selection plots for the Jeep Renegade dataset. These suggest that little additional information is conferred by the gradient signal collected by the Jeep.

TABLE 1. State-Space system identification Algorithm parameters.

Algorithm Option	Method
Parameter Initialisation	N4SID
Initial State	Estimated
N4SID Weighting Scheme	CVA
N4SID Prediction Horizon	Chosen by AIC
Loss Function	Simulation Error
Loss Function Weighting	None
Search Method	GN/AGN/LM/GD

models of first or second order should be further investigated. First and second order models were then analysed and compared with more in-depth methods, including Model Fit, Variance Accounted For (VAF), and residual analysis (see section below on model evaluation and validation). The model was identified using the same training data as for the physical and neural network models.

D. NEURAL NETWORK MODEL

The neural network model is shown in Fig. 3. The network adopts a structure that is inspired by the physical model: there are 4 converging sub-networks (labelled as 1, 2, 3 & 4) that are intended to learn the effects of $F_p, F_b, Mg\ \sin(\gamma(t))$ and $k_D\, v^2 + Mg\, k_R$ separately. More details on the structured network approach are given in [26].

The network was trained with the same training data used for the state-space and physical model estimation. Over-fitting was monitored by means of the validation sets, again matching those used for the state-space and physical models. A combination of dropout layers (D blocks in Fig. 3) and L2 regularisation was adopted to avoid over-fitting: the dropout probability (about 0.01-0.015) and the L2 regularisation weight decay parameter (about 0.1–0.05) were chosen to obtain very similar losses for the training and the validation subsets (the latter being never used for training).

The network predicts $\dot{v}_k$ defined as $\dot{v}_k = (v_{k+1} - v_k)/\Delta t$ where $\Delta t = 50$ ms is the sampling rate of the signals. Except for the scaling factor Δt, and given that the current value of the velocity v_k is among the inputs (input to branch 1), the network is a predictor of the next value of the forward velocity v_{k+1}. For training, the raw signal $\dot{v}_k = (v_{k+1} - v_k)/\Delta t$, computed as the forward differences of vector v_k, was quite noisy. It was therefore smoothed using a Gaussian filter with radius $n = 8$.

Learning occurs at the fully-connected layers (see Fig. 3 legend) where the size of the input and hidden (output dimension) layers is also indicated. Hence, the layer in branch 1 of the network has two trainable parameters (one weight and one bias). The remaining layers have no biases and so the number of trainable parameters are n weights for branch 2, $2\ n_h + 1$ weights for branch 3 and 8 n weights for branch 4 (8 being the number of forward gears, including the neutral gear), for a total of 248 trainable parameters ($n = 25$, $n_h = 10$).

The 4 sub-networks may now be explained further. First, let us note that the input to each of the 4 sub-networks is different: this reflects the fact that each network aims to learn the *causal* dependency between its input and the corresponding part of the acceleration $\dot{v}_k$.

So, branch 1 learns $w_1\, v_k^2 + b_1$, i.e. it learns one deceleration term proportional to the square velocity (weight w_1) and one constant deceleration term (bias b_1), which will model the aerodynamic drag and rolling resistance respectively.

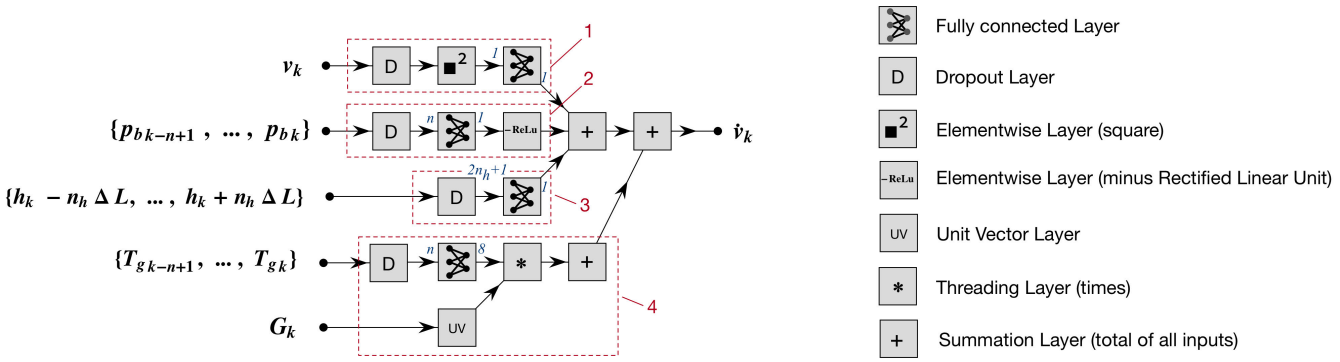

FIGURE 3. Neural network model. The network is made of 4 converging branches that learn the air drag and rolling resistance (1), the brake effect (2), the gradient effect (3) and the engine effect. Branches 2 and 4 learn dynamic models via the learning of the impulse response. Branch 3 estimates the gradient effect from the raw recorded GPS heights. v_k: longitudinal velocity at sample time k; p_{bk}: brake master cylinder pressure at time k; h_k: Road height at time k; T_{gk}: engine torque at time k; G_k: Gear ratio at time k.

Branch 2 learns a linear combination of the past n brake circuit pressure values, i.e., the (acceleration) impulse response of the brake plant (the layer $-ReLu$, a rectified linear unit, helps convergence by taking only the negative values of the sub-network in the training phase; because obviously brakes can only cause deceleration). Made in this way, branch 2 neglects any non-linearity of the brake plant; however it can learn the dynamics of the plant, which was instead implicitly neglected (modelled as a zero order, or instantaneous, plant) in (6).

Branch 3 learns the road gradient effect. Unlike the previous models, the input here is the raw heights recorded by the GPS at $2n_h + 1 = 21$ points, spaced by distance $\Delta L = 10\ m$ ahead and behind the current vehicle position. This way, the network itself determines the optimal gradient estimation (the one minimising the output error) based on the raw heights, wheres for the previous models the road gradient had to be estimated separately (by linear regression). If the road gradient had been available from accurate digital maps (which was not the case), this branch could be replaced with one directly using the gradient as input.

Finally branch 4 learns the acceleration caused by the engine. A fully connected layer with 8 output neurons (one per gear, including the null gear) first learns the acceleration that would be produced by each gear individually, given the last n values of the raw engine torque. So, this layer learns the impulse responses of the drive-line at the different gears. Again, compared to 3 the engine model is still linear, but dynamic instead of static. The eight sub-model outputs are then inhibited except the one corresponding to the current gear (this is obtained by element-wise multiplication of the 8 outputs with a unit vector with '1' on the active gear and '0' elsewhere). This way only the current engaged gear contributes to the vehicle acceleration. It is also worth noting that by learning the impulse response of each gear, the engine/drive-line plant also learns the individual gear ratios (whereas known values are incorporated into the other models).

The network training was carried out with the Adam algorithm [38], which at every iteration updates only the weights of the current gear (the stochastic gradient descent solver would fail, finding infinite gradients for the weights of the inactive gears).

The training process for the neural network model was carried out in two phases. The first step took about 8 minutes on a Intel i7 (single core) CPU. The second took 16 minutes with reduced learning rate and larger batch size.

E. MODEL EVALUATION AND VALIDATION METHODS

Models were evaluated using a normalised fit metric based on the Euclidean norm of the fit error, where

$$\text{Model Fit} = 100\left(1 - \frac{||\mathbf{y} - \hat{\mathbf{y}}||_2}{||\mathbf{y} - \bar{\mathbf{y}}||_2}\right) \tag{11}$$

where $\mathbf{y}$ is the vector of measured output data, $\hat{\mathbf{y}}$ is the vector of model simulation outputs, and $\bar{\mathbf{y}}$ is the mean of measured output data. A value of 100% indicates a perfect model fit, a value of 0% indicates a fit equivalent to the mean of the output data, and becomes negative for poor fits.

The model fit was also assessed using the variance accounted for (VAF) metric, which is equivalent to the r^2 metric, defined as

$$\text{VAF} = 100\left(1 - \frac{\text{var}(\mathbf{y} - \hat{\mathbf{y}})}{\text{var}(\mathbf{y})}\right) \tag{12}$$

To provide a comparison with [21], where it is called average prediction error (APE), the root of the mean of the squared velocity error (RMSE) was also computed, and is given in Fig. 9(b).

To evaluate the models in terms of accuracy and complexity, Akaike's final prediction error (FPE) was used,

$$\text{FPE} = \frac{1}{N}\sum_{k=1}^{N} e_1(t_k, \hat{\boldsymbol{\theta}})^2 \times \left(\frac{1 + n_p/N}{1 - n_p/N}\right) \tag{13}$$

where n_p is the number of model parameters and $e_1(.)$ denotes the one-step-ahead prediction error.

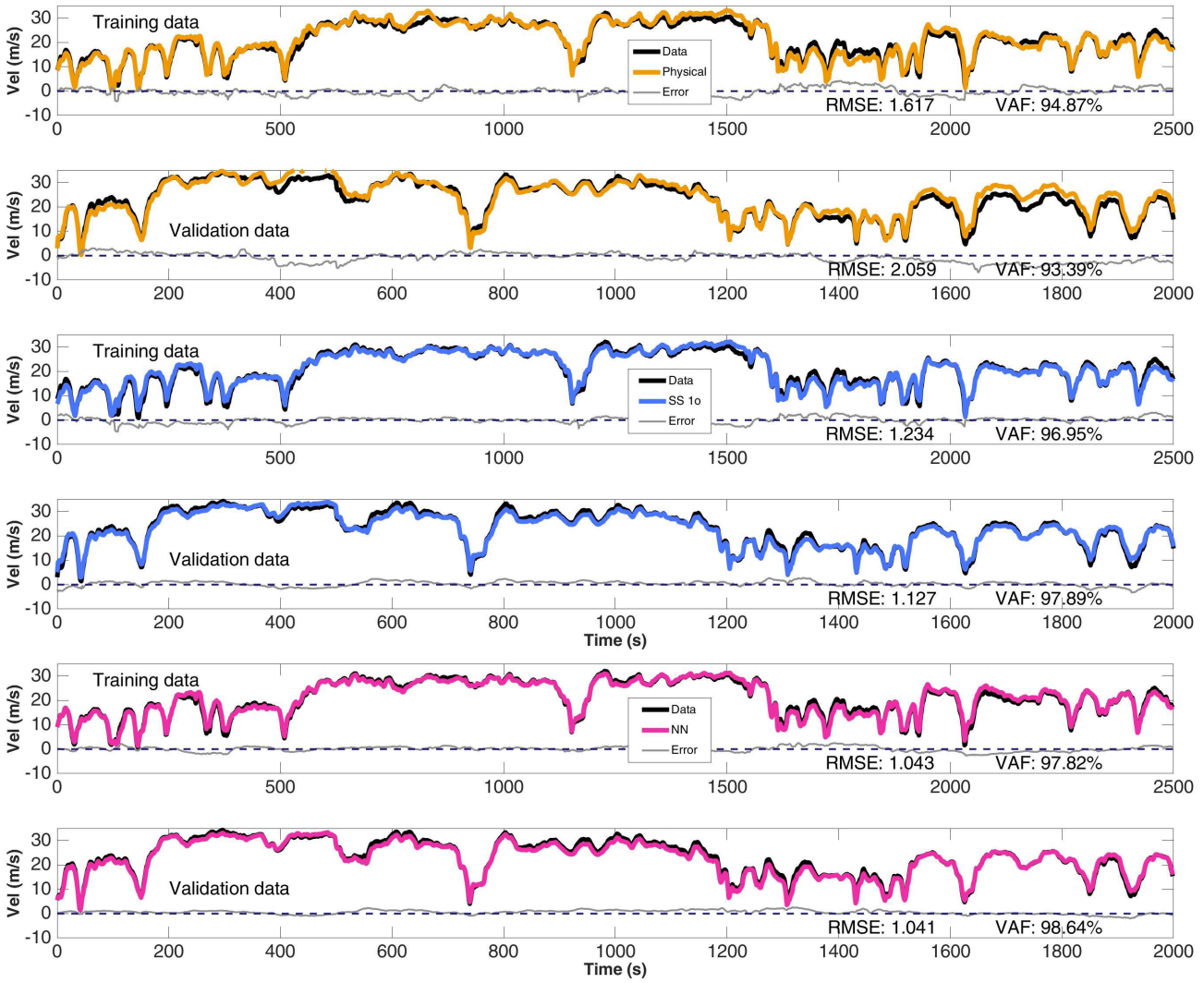

FIGURE 4. Simulation results for physical (top two graphs), state-space (middle two graphs) and neural network models trained with the Lancia Delta data. For each model, the simulation is carried out on the training data and on the validation data and plotted in colour, with the recorded data plotted in black. The difference between simulation and data is plotted in gray.

III. RESULTS

The physical model estimation process with 100 random starting points required 14 hours on a modern, multi-core CPU (Intel i9, 12 cores). This very lengthy compute time was related to the length of the training data, and the fact that some regions of the parameter space produced stiff differential equations. The nonlinear physical model parameters were estimated as $k_D = 0.2777$, $k_R = 0.0101$ and $k_\tau = 9.469$ for the Lancia Delta, and as $k_D = 2.415$, $k_R = 0.00023$ and $k_\tau = 120.9$ for the Jeep Renegade. The full list of parameter values in the physical model is given in Table 2, along with their units and whether they were found by the parameter search or were known a priori. The difference in the drag parameter, k_D may be attributable to the less aerodynamic design of the Jeep SUV; the difference in the torque parameter, k_τ, to different gearing in the drive-trains of the two vehicles. Simulation results for the physical model are plotted against data in Figs. 4 & 5 (top two graphs of each figure, in orange).

TABLE 2. Physical model parameter results.

Parameter	Delta	Renegade	Unit	Est./known
M	1550	1632	kg	Known
g	9.80665	9.80665	m/s^2	Known
k_τ	9.469	120.9	m^{-1}	Estimated
k_b	189	232†	N/bar	Known/Est.
k_D	0.2777	2.415	kg/m	Estimated
k_R	0.0101	2.286×10^{-4}	-	Estimated
Gear ratios	-	-	-	Known

† 232 N/bar was the value of k_b estimated by the neural network for the Jeep Renegade and was used in the absence of a value provided by the manufacturer.

The state-space modelling process began with the selection of model order. The N4SID algorithm was applied for

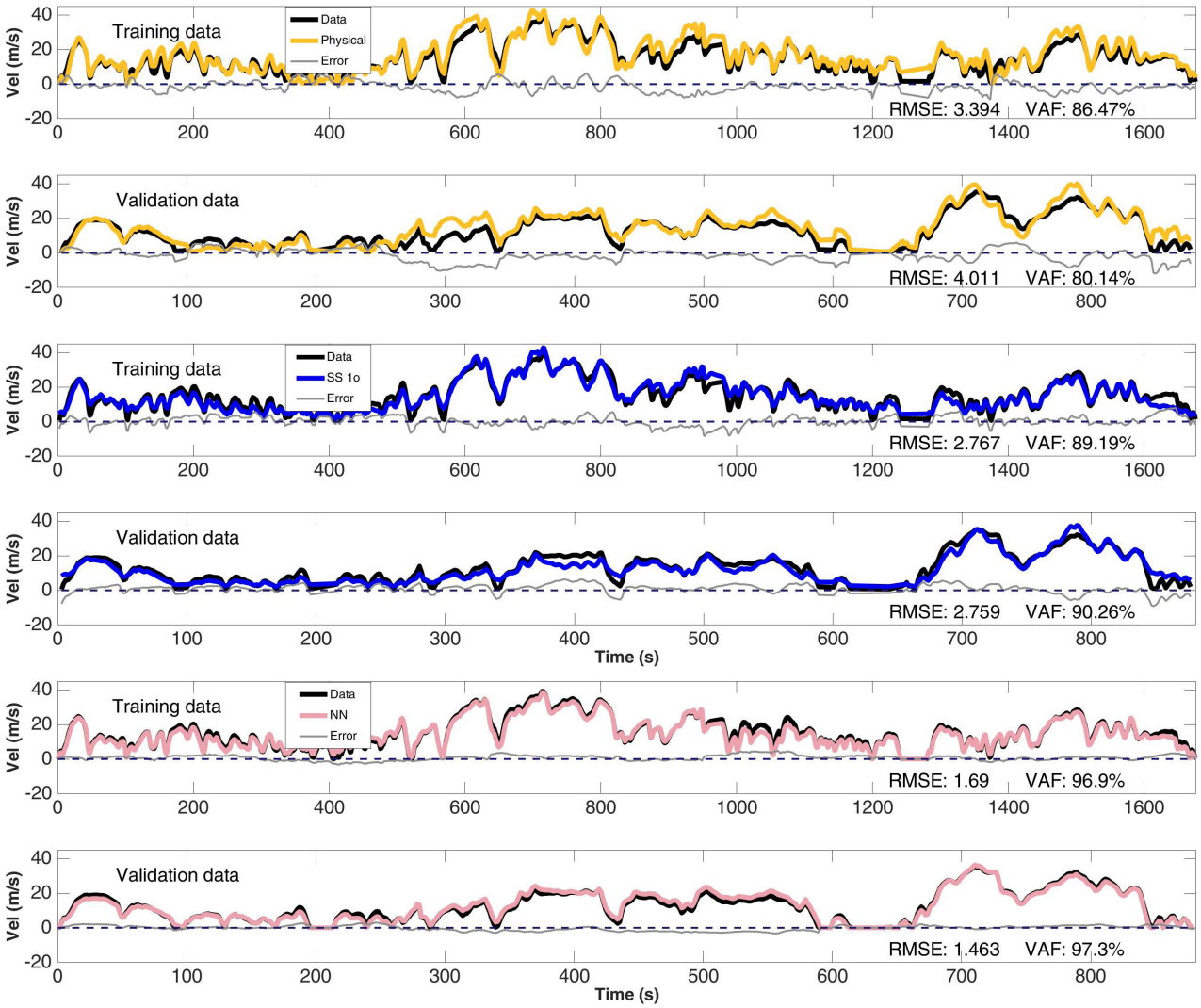

FIGURE 5. Jeep Renegade simulation results for physical (top two graphs), state-space (middle two graphs) and neural network models. As in Fig. 4, the simulation is carried out on the training data and on the validation data and plotted in colour, with the recorded data plotted in black. The difference between simulation and data is plotted in gray.

orders $n_x = 1, \ldots, 10$, using the singular values of the input-output covariance matrix. This indicated that a first or second order model would be likely to provide the best fits (results not shown). The first and second order models were then evaluated with three combinations of input: torque-only, torque-brake and torque-brake-gradient. Several metrics of the quality of the fit are presented in Fig. 2. Inspection of the fit metrics indicated that inputs should consist of torque-brake-gradient for the Lancia Delta [Fig. 2(a,b)] and at least torque-brake inputs for the Jeep [Fig. 2(c,d)], although for this dataset, the information in the elevation does not seem to contribute much improvement to the fit because the training route was relatively flat and the elevation signal somewhat noisy.

Selecting between the first or second order models requires analysis of the model residuals and consideration of the model complexity. Input cross-correlations (not shown) do not strongly favour either first or second order models. On this basis we might choose the first order model as the optimum representation of the Lancia, and the second order model for the Jeep even though the first order model would usually be preferred for its simplicity. Also the FPE for the first order model is less than for the second order model for both vehicles, indicating that the first order model is a better trade-off of accuracy and complexity [Fig. 2(b)]. Overall fit quality for the Jeep Renegade is lower than for the Lancia Delta dataset and there is a less clear improvement provided by the road gradient signal. Although second order models produce similar fit qualities, the cross-correlations for the second order torque-brake-gradient model are noticeably worse than for the first order model.

To further investigate the choice of first order vs. second order, we split each dataset into four sections. The state-space model was identified for each of the four sections, and

TABLE 3. Four way state-space model validation (VAF metric).

	Lancia dataset:				Jeep dataset:			
1^{st} order	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Model 1	**97.8**	83.9	94	88.9	**89.2**	82.2	90.3	86.1
Model 2	95.8	**96**	97.2	96.7	76.8	**84.8**	82.4	91.8
Model 3	95	94.6	**96.8**	96.2	87.7	80.3	**90.5**	84.3
Model 4	97.5	96.1	97.9	**97.8**	70.3	80.3	77.7	**91.2**
2^{nd} order	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Model 1	**97.3**	80.4	92.7	85.4	**93.9**	85.1	93	84.6
Model 2	96.5	**95.8**	96.7	95.9	51.9	**77.7**	73.3	48
Model 3	96.6	90.5	**95**	91.2	-49.3	-28.4	**-4.1**	31.9
Model 4	96.1	94.1	92.9	**96.3**	-49.3	-16.9	-11.7	**38.7**

Model 1 is trained on Data 1, Model 2 on Data 2, etc. Values in bold are the VAF metrics for the models validated on their own training data.

then the VAF fit metric was computed against the remaining three sections, giving 12 validation fits for each model order/vehicle (Table 3). The quality of the validation fits were analysed to find a mean VAF with a standard error (computed by bootstrap). For the Lancia Delta, the mean validation fit was 94.5(2.4) for the first order model and 92.4(2.9) for the second order model. For the Jeep Renegade, the mean validation fit was 82.5(3.5) [first order] and 26(26.2) [second order]. This indicates that the first order model generalises better than the second order model for both vehicles. The second order model was particularly poor for the Renegade for two of the four datasets, hence the very low, and highly variable validation VAF.

For the Lancia Delta, the linear first order model with torque-brake-gradient inputs (u1, u2, u3) was identified as

$$A = \begin{bmatrix} -0.0008098 \end{bmatrix} \tag{14}$$

$$B = \begin{bmatrix} 0.00000137 & -0.0000294 & -0.002256 \end{bmatrix} \tag{15}$$

$$C = \begin{bmatrix} 3995 \end{bmatrix} \tag{16}$$

$$D = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \tag{17}$$

For the Jeep Renegade, the linear first order model (incorporating torque, brake and gradient in analogy with the Lancia model) was

$$A = \begin{bmatrix} -0.03062 \end{bmatrix} \tag{18}$$

$$B = \begin{bmatrix} 0.0000245 & -0.0000198 & -0.0004249 \end{bmatrix} \tag{19}$$

$$C = \begin{bmatrix} 2047 \end{bmatrix} \tag{20}$$

$$D = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \tag{21}$$

Figs. 4 & 5 show the results of the state-space model (middle graphs, in blue). It took approximately 8 seconds of compute time on an Intel i9 CPU (using a single core) to identify one state-space model and 100 seconds to identify 12 models and select the best one.

The neural network model required a total of 24 minutes to train. Results are given in the bottom two graphs in Figs. 4 & 5 (pink).

The overall Lancia Delta VAF, on a reserved set of independent validation data, for the nonlinear physical model was 93.4% (RMSE=2.06 m/s), for the linear first order model it was 97.9% (RMSE=1.12 m/s), and for the neural network model was 98.6% (RMSE=1.04 m/s). The reader will notice that, for the state-space and neural network models (Fig. 4, blue and pink graphs), the VAF for the validation dataset is marginally higher than that for the training set, contrary to the usual expectation that the model should fit best the data on which it was trained. Inspection of the validation set shows that it has fewer sharp acceleration and deceleration events, suggesting the reason for the slightly improved validation fits. For the Jeep Renegade, the overall validation VAF was 80.1% (RMSE=4.01 m/s) for the physical model, 90.3% (RMSE=2.76 m/s) for the state-space model, and 97.3% (RMSE=1.46 m/s) for the neural network model.

Simulation comparison of the nonlinear physical, linear first order and neural network models to each other and the observed data demonstrated that firstly, all models were accurate predictors of car velocity (Figs. 4 & 5), and that secondly the data driven models (state-space and neural network) were superior in accuracy to the nonlinear model for both vehicles (Figs. 6 & 9). Of the two data driven models, the neural network slightly outperformed the first order state-space model, but at a cost of approximately an order of magnitude more training time.

Visual inspection of Fig. 6 gives an indication of the variation of the accuracy of the simulated velocity with respect to the measured velocity. To further investigate the quality of the models at different speeds we performed an analysis of the RMS simulation errors for the Lancia validation dataset. We used the three models (choosing the first-order torque-brake-gradient model as the best state-space example) and computed the root-mean squared velocity errors for each gear. Fig. 7(a) shows the result of this analysis. The state-space and neural network models have similar accuracy in 3rd gear and above (i.e. at higher speeds) and out-perform the physical model. In 1st and 2nd gear; the neural network model is better than both other models.

In a similar manner, we computed the RMS velocity error by grouping the data into 5 m/s wide bins. Fig. 7(b) shows the result for the Lancia, which has the same form as Fig. 7(a). Fig. 7(c) gives the error vs. speed result for the Jeep validation dataset.

To understand what quantity of training data is required to be able to compute an accurate model, we investigated models trained on subsets of the full Lancia Delta training dataset of 2500 s. We trained the physical, and the first-order torque-brake-gradient state-space models on sub-divisions of these data. We divided the data into 2, 4, 8, 16 or 32 equal length subsets for training. In order to provide sufficient persistence of excitation to find a good model, we found it was necessary to exclude data subsets for which the vehicle velocity varied by less than 10 m/s, such as those for which the vehicle was cruising on fast, extra-urban roads. Shorter subsets were more likely to be excluded according to this criterion. We excluded 0 of the 1/4 length subsets, 2 of the 1/8 subsets, 5 of the 1/16 subsets and 21 out of the 1/32 length subsets. We then

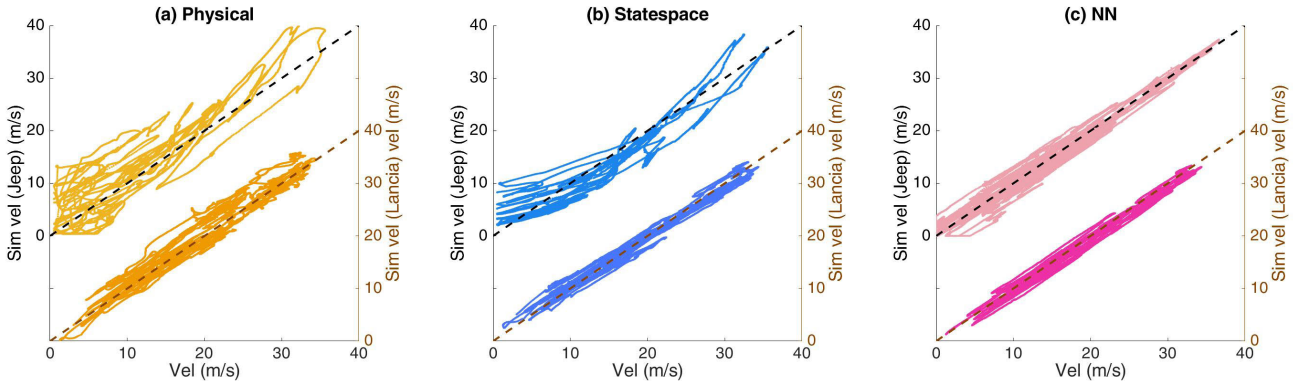

FIGURE 6. Plotting the simulated velocity versus the actual velocity. The closer the points lie to a straight line, the better the model fit. These graphs [for physical (a), state-space (b) and neural network (c)] each show data for the Jeep Renegade (left y axis) and Lancia Delta (right axis). The graphs indicate that for both vehicles the physical model performs poorly compared with the state-space and neural network models. They also suggest that the data quality was superior for the Lancia Delta, as for each model trained against the data for this vehicle, the simulated vs. actual velocity is clustered most closely to the straight line. Graph (b) suggests a non-linearity in the data for the Jeep Renegade, due to the apparent dishing of the data points.

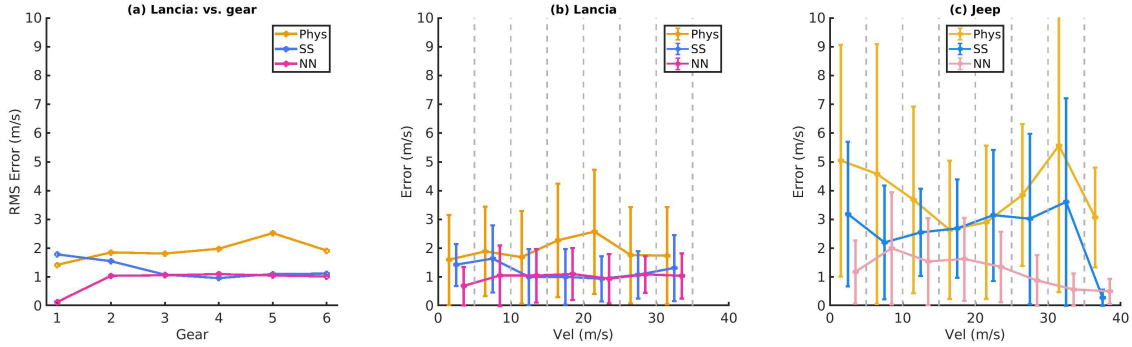

FIGURE 7. Model accuracy vs. vehicle speed. a) RMS velocity error vs. gear for physical (Phys), state-space [tbg1o] (SS) and neural network (NN) model simulations. b) & c) RMS velocity error vs. measured velocity for Lancia (b) and Jeep (c). The velocity error is the measured velocity subtracted from the simulated velocity computed from the data shown in Fig. 6. Velocity errors are then grouped by their corresponding measured velocities, in 5 m/s wide bins. The graphs show the RMS velocity error (points) and velocity error standard deviation (bars). The position of the points for Phys and NN datasets have been offset in velocity by −1, +1 m/s for clarity only. Grey lines show the bin edges).

trained the models on the remaining subsets and for each model, we simulated the usual validation dataset to compute a VAF. For the neural network model, we sub-divided the data using a different method. Because it must find the parameters for the gear model (as shown in Fig. 3, the neural network learns one sub-model per gear), it requires input data for the vehicle operating in all gears. (In contrast, the physical and state-space models use the post-gearbox torque which incorporates the known gear ratios). Thus, the data was divided into 10 s sections, then (for example) to create the 1/4 length data subset, every 4th 10 s section was added (subsets that still did not include examples for all gears were also excluded). Fig. 8 shows the mean VAF for each subset length, with error bars showing the standard deviation. The data-driven models are seen to maintain a good accuracy, with little change in VAF until the training set length drops below about 600 s (for the neural network the limit is the size of the examples for the first gear, which is the one with least data). The variability of the physical models is much greater than that of the data-driven models, especially for small training set sizes.

To verify that longitudinal-lateral coupling is weak, and that data-driven models trained on datasets which incorporate road bends can provide good prediction accuracy, we took the neural network Lancia Delta model, which was trained on the full-length training dataset, and validated it on subsets of the data which included i) only roads with bends (radius < 500 m) or ii) only roads with sharp bends (radius < 50 m). Validating on the 13.3% of the data with mild and sharp bends gave a VAF of 98%. Validating on the 3.4% with only sharp bends gave a reduced VAF of 79.1%. The model appears to lose accuracy only in sharp bends.

IV. SUMMARY OF RESULTS

In this paper, we have compared several mathematical models of longitudinal car vehicle dynamics using large-scale real-world driving experimental data for two different vehicles, specifically:

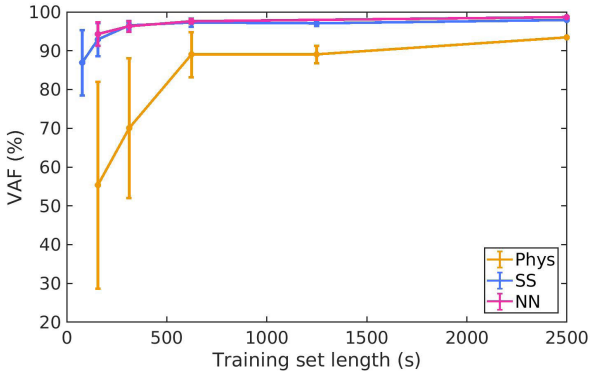

FIGURE 8. Variation-accounted-for (VAF) metric giving simulation accuracy for physical (Phys), state-spate (SS - tbg1o) and neural network (NN) models trained on differing set sizes. SS and Phys models were trained on halves, quarters, eighths, etc of the original training dataset. However, if a given fraction of the data did not have at least 10 m/s of variability in the velocity, that fraction was discarded and not used for training. The NN model was trained on data that had been subdivided into ten second sections and then interleaved to make, say, 4 training datasets each a quarter of the length of the original dataset. The reason for this approach was to ensure that the NN model received data from driving conducted in all gears. The number of VAFs computed at each training set length varied; for the full length of 2500 s, there is only one VAF (and is the value reported in the main results). For 625 s four VAFs were computed; for 312 s, up to 8 VAFs were computed etc. The mean VAF is plotted, with error bars giving the standard deviations. The graph indicates that approximately 10 minutes of training data is required as a minimum to get a good model.

- a nonlinear physical model
- linear state-space models, focusing on first and second order, obtained by system identification techniques
- a neural network model whose design is guided by physical knowledge

The key results are that the data-driven models improve on the accuracy of the nonlinear physical model for both vehicles (Fig. 9). The most accurate model for each vehicle was the neural network, which slightly out-performed the state-space model for the Lancia Delta and was a significant improvement for the Jeep Renegade data.

Model accuracy for all three models was lower for the Jeep than for the Lancia. The torque signal from the Jeep's ECU is generally noisier, contributing to the increased scatter in the upper plots in Fig. 6. Secondly, there is a nonlinear trend in the upper plot of Fig. 6(b) for the state-space model of the Jeep. This non-linearity is consistent with the quadratic drag term in the physical model: the linear state-space model is over-estimating the speed when the Jeep is travelling fast. This could be attributable to the less aerodynamic profile of the Jeep vehicle (an SUV) in comparison with the Lancia Delta car. Indeed, the physical model finds the drag parameters, k_D to be greater for the Jeep than for the Lancia. Notably, the neural network model is able to learn the non-linearity in the Jeep data.

V. DISCUSSION

The aim of the work was to assess the advantages of each form of modelling within the context of developing autonomous vehicle control systems. Autonomous vehicles will require accurate vehicle dynamics models both for control applications and as forward models for prediction to aid decision making processes. Thus, the focus of the work is prediction accuracy, rather than parameter estimation. Experimental data used in the study was drawn from normal driving over circa 100 km of roads. This dataset was particularly useful as it reflects the type of data that would be continuously available to an autonomous vehicle during its operational life, that might be used to update data-driven models.

A. APPLICATION OF DATA DRIVEN MODELS

The models described in this work are all 'forward' (i.e. predictive) models. Given control input, they produce a predicted output, $v(t)$. The use of such models is common in control systems, even though for control it is the 'inverse model' that is required; a desired trajectory is the input and control signals are the required output. Nonetheless, the exploitation of forward models is important in several control schemes which incorporate both forward and inverse data-driven models. For examples, see [39] (their Fig. 8) which shows neural network models being used as inverse/forward pairs and the 'feedforward inverse control' system, which feeds an estimate of the disturbance back to the controller as an adaptive signal. For comprehensive discussions of adaptive control, see [40], [41].

Exploiting a neural network model of the real plant to find the neural network for the inverse model can be carried out by training the network that cancels the forward network. This can be i) more efficient than learning a neural network controller via direct interaction with the plant, allowing algorithms such as gradient descent to be applied; and ii) safer, as it is possible to test actions that would be dangerous. Furthermore, the training of the network that cancels the forward model is an unsupervised learning problem (once the forward model is learned) and, by controlling the set of training data (which trajectories we are really interested in producing well) it is possible to control on which domain the inverse model is accurate. Inverting the forward model to find the inverse can even be analytically computed for the linear state-space models explored in this work.

The resulting data-driven inverse model is, also, somewhat equivalent to model predictive control in the sense that (after training is completed) when used for control it takes a future desired trajectory as input and returns the control commands that, fed into the plant model, drive it onto the desired trajectory. If the inverse model is then transferred to controlling the real plant, the accuracy of the plant model becomes important (the real plant will otherwise diverge from the model), which is the main motivation of this paper.

Another use for forward models is in online model selection. If there is significant mismatch between the model's prediction and sensory data, the control system could select between alternative inverse models [42]. For example if the car runs on a slippery road the predictions of a model trained on a dry, well-maintained road will fail. The agent may have multiple forward models (one for slippery roads; one for dry

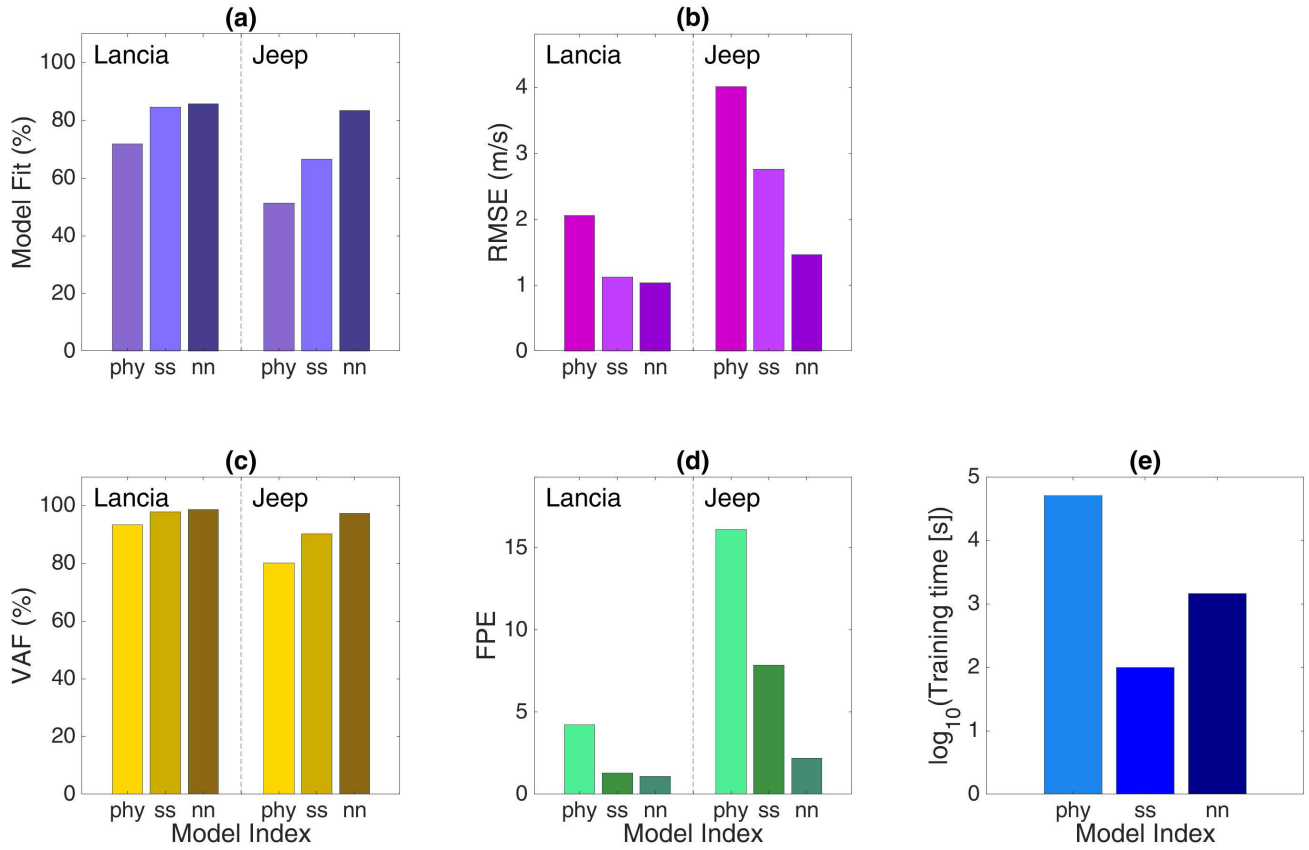

FIGURE 9. (a)–(d). Model fit, RMS error (RMSE), variance accounted for (VAF) and Akaike's final prediction error metrics (FPE) for the physical (phy), first-order state-space (ss) and neural network (nn) models. For the Lancia Delta data (left bars), the physical model is out-performed by the alternatives for each metric. The neural network model performs slightly better than the state-space model, though at a cost of an increased training time. For the Jeep Renegade data, the neural network model still outperforms the physical model by a substantial margin, but the state-space model is comparable with the physical model in terms of accuracy; (e) Model training time. Note the log scale and note also that training was carried out on three separate computer systems and so this graph provides a guide, rather than a direct comparison of the three methods.

roads) and estimate which condition holds. If no existing model matches sensory data then a new model could be created.

Finally, forward models provide the ability to carry out internal simulations, which is considered to be at the root of thought. Indeed the simulation theory of cognition postulates that thoughts are sequences of simulated actions and simulated perception at different levels of abstractions [43], [44].

B. VEHICLE DIFFERENCES

The differences in the parameters found by the physical model for the two vehicles warrants discussion (Table 2). While it is possible to explain the difference in the torque parameter, k_τ, by an additional gearing in one vehicle, the magnitude of the differences in the other parameters is less easy to dismiss. Although the increased drag parameter, k_D, is consistent with the less aerodynamic design of the Jeep SUV, it is hard to believe that the rolling resistance (k_r) of the Lancia is orders of magnitude greater than the Jeep. More likely is that this is simply the best fitting parameter for a system which is failing to capture some unmodelled effects and throws into doubt the interpretability argument in favour of the physical model.

Interestingly, k_D and k_r can also be obtained by inspection of the trained neural network, thanks to its structured nature. Indeed, as already explained, branch 1 (Fig. 3) learns the combined aerodynamic and rolling resistance (respectively by means of its weight and bias). Hence, inspection of the trained parameters reveals the sub-model parameters. For the Delta, the network estimates are $k_D = 0.2854$ (kg/m) and $k_r = 0.01208$, which are similar to Table 2. Instead, for the Renegade they are $k_D = 0.4209$ (kg/m) and $k_r = 0.00657$, which are more plausible. It is likely that the neural network model, having a more flexible nonlinear structure, also correctly fits the parameters. Hence, the interpretability argument is reversed in favour of the network in this case.

C. EXTENDING THE DOMAIN

The study focused only on the ordinary driving domain; good driving conditions; safe, legal speeds and comfortable accelerations [32]. It is worth considering the approach that would be necessary to extend these models to data collected in more

extreme conditions, such as driving in an icy environment, hard, emergency braking or track driving at high speed. For the physical model, it would be necessary to re-engineer the model, introducing additional terms and refined models for tyre forces, which in turn might call for complex and more costly experiments [45]. In more extreme driving, it would also be necessary to consider selecting between prediction models (and their inverted control counterparts), depending on conditions [42]. Also, the assumption of de-coupled lateral-longitudinal dynamics would need to be examined carefully; the introduction of a lateral physical model significantly increases the complexity of physical vehicle dynamic modelling. If this assumption were to fail for the physical model, it would also fail for the data-driven models. However, for these models, the solution is more straightforward; the additional control inputs are incorporated into the modelling, but the methodology remains the same, albeit with a requirement for a wider range of conditions in the training set, i.e. for more experiments in the targeted domain (extra experiments are also needed for more parameter estimation of the analytical models).

D. PERSISTENCY OF EXCITATION

The focus of this study is on ordinary driving on public roads, which consequently means there is an imperfect fulfilment of the usual requirement in system identification to fully excite the inputs, i.e. the system lacks full 'persistency of excitation'. Given that the data collected is from normal, legal driving on public roads, it is clearly impossible to excite the inputs across their full ranges. For example, we do not have data for conditions corresponding to hard braking manoeuvres or for full acceleration in all gears and at all speeds.

However, we allow for partial excitation of the system for two reasons:

Firstly, the results indicate that the identification process is still successful, in agreement with other studies of persistency of excitation [46]–[48].

Secondly, we seek to compare predictive models in real-life conditions typical of normal, safe driving. If predictive models can be trained from normal driving data, manufacturers will incur lower development costs because they will not need to conduct the work at dedicated track facilities. Such models could even be trained on-line, by automated driving agents or driver assistance systems. Thus, it is worthwhile to make a comparison of the accuracy of different models trained on these datasets.

E. SUMMARY

We find that the use of data-driven models may be encouraged within the vehicle control community. Linear state-space models, which have advantages for use in control and state estimation due to their linear structure, may be favoured as long as the vehicle operates linearly in the driving conditions in which it will be deployed. The neural network model was most accurate, offering a systematic approach to the learning of models that better describe vehicle dynamics in nonlinear conditions. The recommendation of this paper is therefore to test a vehicle with both linear state-space and neural network data-driven models to evaluate which will fit best. For autonomous vehicle applications, the ability to use the simplest, and most quickly identified, state-space models may motivate the specification of low-drag vehicle bodies for driverless cars. On the other hand, neural network models may be the building block on which a complete sensorimotor strategy can be bootstrapped. In other words, neural network models can be learned (by a driving agent or by a human supervisor designer) and used to produce lower and higher-level control loops in way that resembles human hierarchical control, as explained in [49].

VI. FUTURE WORK

A limitation of this investigation is that it focused on longitudinal vehicle dynamics only, and providing no modelling of the vehicles lateral dynamics. The lateral dynamics might lead to more severely nonlinear behaviour, necessitating further nonlinear modelling to adequately describe the car behaviour for full path control. In future work, we plan to extend the comparison of physical and data-driven models to the coupled longitudinal-lateral vehicle dynamics.

ACKNOWLEDGMENT

The authors would like to thank A. Saroldi and Centro Ricerche Fiat for their contribution in collecting the data and Francesco Biral for useful discussion.

REFERENCES

[1] V. Milanes, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 296–305, Feb. 2014.

[2] S. E. Li, Q. Guo, S. Xu, J. Duan, S. Li, C. Li, and K. Su, "Performance enhanced predictive control for adaptive cruise control system considering road elevation information," *IEEE Trans. Intell. Vehicles*, vol. 2, no. 3, pp. 150–160, Sep. 2017.

[3] S. Sheikholeslam and C. A. Desoer, "Longitudinal control of a platoon of vehicles with no communication of lead vehicle information: A system level study," *IEEE Trans. Veh. Technol.*, vol. 42, no. 4, pp. 546–554, Nov. 1993.

[4] G. Guo and W. Yue, "Autonomous platoon control allowing range-limited sensors," *IEEE Trans. Veh. Technol.*, vol. 61, no. 7, pp. 2901–2912, Sep. 2012.

[5] S. Wen, G. Guo, B. Chen, and X. Gao, "Cooperative adaptive cruise control of vehicles using a resource-efficient communication mechanism," *IEEE Trans. Intell. Vehicles*, vol. 4, no. 1, pp. 127–140, Mar. 2019.

[6] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.

[7] M. Da Lio, F. Biral, E. Bertolazzi, M. Galvani, P. Bosetti, D. Windridge, A. Saroldi, and F. Tango, "Artificial co-drivers as a universal enabling technology for future intelligent vehicles and transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 244–263, Feb. 2015.

[8] A. Bisoffi, F. Biral, M. Da Lio, and L. Zaccarian, "Longitudinal jerk estimation of driver intentions for advanced driver assistance systems," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 4, pp. 1531–1541, Aug. 2017.

[9] R. Hult, F. E. Sancar, M. Jalalmaab, A. Vijayan, A. Severinson, M. Di Vaio, P. Falcone, B. Fidan, and S. Santini, "Design and experimental validation of a cooperative driving control architecture for the grand cooperative driving challenge 2016," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1290–1301, Apr. 2018.

[10] H. Pham, M. Tomizuka, and J. K. Hedrick, "Integrated maneuvering control for automated highway systems based on a magnetic reference/sensing system," Inst. Transp. Stud., Univ. California, Oakland, CA, USA, Tech. Rep. UCB-ITS-PRR-97-28, Jan. 1997. [Online]. Available: https://escholarship.org/uc/item/7hk255bn
[11] P. Petrov and F. Nashashibi, "Modeling and nonlinear adaptive control for autonomous vehicle overtaking," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1643–1656, Aug. 2014.
[12] V. Cerone, A. Chinu, and D. Regruto, "Experimental results in vision-based lane keeping for highway vehicles," in *Proc. Amer. Control Conf.*, vol. 2, May 2002, pp. 869–874.
[13] V. Cerone, M. Milanese, and D. Regruto, "Combined automatic lane-keeping and Driver's steering through a 2-DOF control strategy," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 1, pp. 135–142, Jan. 2009.
[14] V. Turri, A. Carvalho, H. E. Tseng, K. H. Johansson, and F. Borrelli, "Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 378–383.
[15] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 566–580, May 2007.
[16] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, "Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation," *Int. J. Robust Nonlinear Control*, vol. 18, no. 8, pp. 862–875, May 2008.
[17] E. Kayacan, "Multiobjective H_∞ control for string stability of cooperative adaptive cruise control systems," *IEEE Trans. Intell. Vehicles*, vol. 2, no. 1, pp. 52–61, Mar. 2017.
[18] G. V. Raffo, G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker, "A predictive controller for autonomous vehicle path tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 92–102, Mar. 2009.
[19] D. E. Smith and J. M. Starkey, "Effects of model complexity on the performance of automated vehicle steering controllers: Model development, validation and comparison," *Vehicle Syst. Dyn.*, vol. 24, no. 2, pp. 163–181, Mar. 1995.
[20] A. Mozaffari, M. Vajedi, and N. L. Azad, "A robust safety-oriented autonomous cruise control scheme for electric vehicles based on model predictive control and online sequential extreme learning machine with a hyper-level fault tolerance-based supervisor," *Neurocomputing*, vol. 151, pp. 845–856, Mar. 2015.
[21] K. J. Hunt, J. C. Kalkkuhl, H. Fritz, and T. A. Johansen, "Constructive empirical modelling of longitudinal vehicle dynamics using local model networks," *Control Eng. Pract.*, vol. 4, no. 2, pp. 167–178, Feb. 1996.
[22] M. Abtahi, "Intelligent identification of vehicle's dynamics based on local model network," *J. AI Data Mining*, vol. 7, no. 1, pp. 161–168, May 2018, doi: 10.22044/jadm.2018.5334.1642.
[23] J. Kalkkuhl, K. J. Hunt, and H. Fritz, "FEM-based neural-network approach to nonlinear modeling with application to longitudinal vehicle dynamics control," *IEEE Trans. Neural Netw.*, vol. 10, no. 4, pp. 885–897, Jul. 1999.
[24] S. J. Rutherford and D. J. Cole, "Modelling nonlinear vehicle dynamics with neural networks," *Int. J. Vehicle Design*, vol. 53, no. 4, pp. 260–287, Jan. 2010. [Online]. Available: https://www.inderscienceonline.com/doi/abs/10.1504/IJVD.2010.034101
[25] G. Garimella, J. Funke, C. Wang, and M. Kobilarov, "Neural network modeling for steering control of an autonomous vehicle," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. Vancouver, BC, USA: IEEE, Sep. 2017, pp. 2609–2615. [Online]. Available: http://ieeexplore.ieee.org/document/8206084/
[26] M. D. Lio, D. Bortoluzzi, and G. P. R. Papini, "Modelling longitudinal vehicle dynamics with neural networks," *Vehicle Syst. Dyn.*, pp. 1–19, Jul. 2019, doi: 10.1080/00423114.2019.1638947.
[27] M. D. Lio, A. Mazzalai, D. Windridge, S. Thill, H. Svensson, M. Yüksel, K. Gurney, A. Saroldi, L. Andreone, S. R. Anderson, and H. Heich, "Exploiting dream-like simulation mechanisms to develop safer agents for automated driving: The 'Dreams4Cars' EU research and innovation action," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–6.
[28] S. James and S. R. Anderson, "Linear system identification of longitudinal vehicle dynamics versus nonlinear physical modelling," in *Proc. UKACC 12th Int. Conf. Control (CONTROL)*, Sep. 2018, pp. 146–151.
[29] R. Rajamani, *Vehicle Dynamics and Control*. New York, NY, USA: Springer, 2011.
[30] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.
[31] O. Nelles, *Nonlinear System Identification*. Berlin, Germany: Springer-Verlag, 2001.
[32] P. Bosetti, M. Da Lio, and A. Saroldi, "On the human control of vehicles: An experimental study of acceleration," *Eur. Transp. Res. Rev.*, vol. 6, no. 2, pp. 157–170, Jun. 2014, doi: 10.1007/s12544-013-0120-2.
[33] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban, "An interior algorithm for nonlinear optimization that combines line search and trust region steps," *Math. Program.*, vol. 107, no. 3, pp. 391–408, Jul. 2006.
[34] L. F. Shampine and M. W. Reichelt, "The MATLAB ODE suite," *SIAM J. Sci. Comput.*, vol. 18, no. 1, pp. 1–22, Jan. 1997.
[35] W. E. Larimore, "Canonical variate analysis in identification, filtering, and adaptive control," in *Proc. 29th IEEE Conf. Decis. Control*, Dec. 1990, pp. 596–604.
[36] P. Van Overschee and B. De Moor, *Subspace Identification for Linear Systems: Theory, Implementation, Applications*. Norwell, MA, USA: Kluwer, 1996.
[37] L. Ljung, "Aspects and experiences of user choices in subspace identification methods," *IFAC Proc. Volumes*, vol. 36, no. 16, pp. 1765–1770, Sep. 2003.
[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[39] K. S. Narendra, "Neural networks for control theory and practice," *Proc. IEEE*, vol. 84, no. 10, pp. 1385–1406, Oct. 1996.
[40] G. L. Plett, "Adaptive inverse control of linear and nonlinear systems using dynamic neural networks," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 360–376, Mar. 2003.
[41] J. Porrill, P. Dean, and S. R. Anderson, "Adaptive filters and internal models: Multilevel description of cerebellar function," *Neural Netw.*, vol. 47, pp. 134–149, Nov. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608012003206
[42] X. Zhang and D. Göhlich, "Integrated traction control strategy for distributed drive electric vehicles with improvement of economy and longitudinal driving stability," *Energies*, vol. 10, no. 1, p. 126, Jan. 2017. [Online]. Available: https://www.mdpi.com/1996-1073/10/1/126
[43] G. Hesslow, "Will neuroscience explain consciousness?" *J. Theor. Biol.*, vol. 171, no. 1, pp. 29–39, Jul. 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022519384712094
[44] G. Hesslow, "Conscious thought as simulation of behaviour and perception," *Trends Cognit. Sci.*, vol. 6, no. 6, pp. 242–247, Jun. 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1364661302019137
[45] S. M. Erlien, J. Funke, and J. C. Gerdes, "Incorporating non-linear tire dynamics into a convex approach to shared steering control," in *Proc. Amer. Control Conf.*, Jun. 2014, pp. 3468–3473.
[46] R. Lozano-Leal, "Robust adaptive regulation without persistent excitation," in *Proc. Amer. Control Conf.*, Jun. 1989, pp. 2303–2308.
[47] C. S. Ludovico and J. C. M. Bermudez, "A recursive least squares algorithm robust to low-power excitation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 2, May 2004, p. II-673.
[48] S. A. Billings, Y. Guo, L. Guo, and H. Wei, "Identification of nonlinear systems with non-persistent excitation using an iterative forward orthogonal least squares regression algorithm," *Int. J. Model., Identificat. Control*, vol. 23, no. 1, pp. 1–7, 2015.
[49] M. D. Lio, D. Windridge, and A. Saroldi. (Dec. 2018). *Dream-Like Simulation Abilities for Automated Cars*. [Online]. Available: https://cordis.europa.eu/project/rcn/206390/factsheet/en

SEBASTIAN S. JAMES received the M.Phys. degree in physics from The University of Manchester, U.K., in 1996, the M.S. degree in physics from Colorado State University, USA, in 1998, and the Ph.D. degree in experimental physics from the University of Cambridge, in 2001. He worked in industry until 2013 and since then has been a Research Associate with the Departments of Psychology and Automatic Control and Systems Engineering, University of Sheffield.

SEAN R. ANDERSON received the M.Eng. degree in control systems engineering, in 2001, and the Ph.D. in nonlinear system identification and predictive control from the Department of Automatic Control and Systems Engineering, University of Sheffield, U.K., in 2005. From 2005 to 2010, he was a Research Associate with the Neural Algorithms Research Group, University of Sheffield. From 2010 to 2011, he was a Research Associate with the Department of Automatic Control and Systems Engineering, University of Sheffield, then became lecturer there, in 2012, and has been a Senior Lecturer, since 2015.

MAURO DA LIO (Member, IEEE) received the Laurea degree in mechanical engineering from the University of Padova, Italy, in 1986. He is currently a Full Professor of mechanical systems with the University of Trento, Italy. His research activity has involved modeling and optimal control of vehicle and spacecraft dynamics, and work on underwater robotics. Recently, his focus shifted to the modeling of human sensorimotor control. He has been involved in several EU framework programmes (PReVENT, SAFERIDER, interactIVe, VERITAS, adaptIVe, and NoTremor) and is the Coordinator of Horizon2020 Dream4Cars.