

Deep learning of vehicle dynamics*

M. Szécsi** B. Györök** Á. Weinhardt-Kovács**
G.I. Beintema* M. Schoukens* T. Péni** R. Tóth***

* *Control Systems Group, Eindhoven University of Technology,
Eindhoven, The Netherlands*

** *Systems and Control Laboratory, HUN-REN Institute for
Computers Science and Control, Budapest, Hungary*

Abstract: Recent advances in deep-learning-based identification of dynamic systems have resulted in a new generation of approaches utilizing state-space neural models with innovation noise structure, improved reformulation of multiple shooting, batch optimization, and a subspace identification-inspired form of encoders. The latter is used to learn a reconstructability map to estimate model states from past inputs and outputs. By using the SUBNET approach, which belongs to the state-of-the-art of these methods, we show how to effectively use these approaches to identify reliable vehicle models from data both in continuous and discrete time, respectively. We showcase the approach on the identification of the dynamics of a Crazyflie 2.1 nano-quadcopter and an F1tenth electric car both in a high-fidelity simulation environment, and in case of the electric car, on real measured data. The results indicate that new-generation of deep-learning methods offer efficient system identification of vehicle dynamics in practice.

Copyright © 2024 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: Nonlinear system identification, Deep learning, Automotive system identification and modelling, UAVs.

1. INTRODUCTION

Due to the significant increase of performance requirements and growing complexity of vehicles, model-based control methods gaining a dominant role in the vehicle industry, especially in motion control and path planning of autonomous ground and aerospace vehicles. However, to reach high-performance maneuvering capabilities, such methods require an accurate personalized model of the vehicle, which is a challenging task given their often nonlinear motion dynamics (tire behavior, aerodynamic effects, attitude dynamics, etc.). Moreover, modern vehicles have a wide array of digital and mechatronic components for which manufacturers do not disclose all details required for modeling and even existing models of subcomponents require coefficient estimation to match the specific characteristics of each vehicle and their change over time.

Hence, it is attractive to use data-driven modeling to capture the relevant vehicle dynamics directly from operational data. Recent advances in deep-learning-based identification of dynamic systems have resulted in a new generation of nonlinear identification approaches utilizing state-space neural models with innovation noise structure (Beintema et al., 2023a), improved reformulation of multiple shooting (Ribeiro et al., 2020), batch optimization, computationally efficient truncated loss functions (Forcione and Piga, 2021), and encoders (Beintema et al.,

2023a; Masti and Bemporad, 2021). The latter is used to learn a reconstructability map to estimate model states from past inputs and outputs. These methods have been shown to be superior in model performance compared to traditional nonlinear identification methods, such as NAR(MA)X, block-oriented, etc., as they build on an efficient fusion of identification methods and deep learning. Furthermore, the estimated models are obtained in a *state-space* (SS) form directly applicable for control design, e.g., see Hoekstra et al. (2023), compared to other approaches of deep-learning based vehicle modeling in terms of LSTMs (Amiruddin et al., 2020), transformers (Saviolo and Loianno, 2023), or end-to-end methods (Hermansdorfer et al., 2020), etc. As a state-encoder is also co-estimated with the model, the method also does not require measured state information such as in (Bansal et al., 2016).

By using the SUBNET approach (Beintema et al., 2023a), which belongs to the state-of-the-art new generation of methods, we show how to effectively use them to identify reliable vehicle models from data both in continuous and discrete time, respectively. We intend to state practical guidelines about the method, then we showcase the approach on the identification of the dynamics of a nano-quadcopter in a high-fidelity simulation environment and a small-scale car using measurements. Furthermore, we discuss various vehicle-specific details, e.g. training data generation, to enhance the applicability of the approach.

The paper is organized as follows. In Sect. 2, the identification problem and the SUBNET approach are introduced. This is followed by providing general guidelines for SUBNET-based vehicle dynamics identification in Sect. 3. Then we demonstrate the capabilities of the method by identifying the dynamics of a Crazyflie 2.1 quadcopter using simulation data in Sect. 4, and the dynamics of an F1tenth small-scale car using measured data in Sect. 5. Fi-



*This project has received funding from the European Defence Fund programme under grant agreement number No 101103386 and has also been supported by the Air Force Office of Scientific Research under award number FA8655-23-1-7061. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the granting authority can be held responsible for them.

nally, conclusions on the discussed identification approach are drawn in Sect. 6.

2. DEEP-LEARNING-BASED SYSTEM IDENTIFICATION

2.1 Problem setting

We consider the representation of the vehicle dynamics in the form a *continuous-time* (CT) nonlinear SS equation:

$$\dot{x}(t) = f_c(x(t), u(t)) \quad (1a)$$

where $t \in \mathbb{R}$ is the continuous time, $x(t) \in \mathbb{R}^{n_x}$ is the collection of the motion states of the vehicle, $u(t) \in \mathbb{R}^{n_u}$ is the input in terms of controlling steering and propulsion of the vehicle, and $f_c : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ is a nonlinear bounded function. Based on digital actuation and data acquisition, we assume that the input satisfies $u(t) = u[k]$ for all $kT_s \leq t < (k+1)T_s$, where $u[k]$ is a *discrete time* (DT) signal with $k \in \mathbb{Z}$ and $T_s \in \mathbb{R}_+$ is the sampling time, corresponding to a *zero-order-hold* (ZOH) setting. Furthermore, measurements of the motion states are available in DT in terms of

$$y[k] = h(x[k]) + e[k] \quad (1b)$$

where $y[k] \in \mathbb{R}^{n_y}$ is the measured output signal, $h : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y}$ is a possibly nonlinear bounded output map, and e_k an i.i.d. white noise process representing measurement noise. Under the ZOH setting, the dynamics represented by (1) can be also described directly in DT in the form:

$$x[k+1] = f_d(x[k], u[k]), \quad (2a)$$

$$y[k] = h(x[k]) + e[k], \quad (2b)$$

where $f_d : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ denotes the DT state-transition function. In the remainder of the paper, we will resort to denoting CT signals as u_t and DT signals as u_k for the sake of readability.

Using a data sequence $\mathcal{D}_N = \{(y_k, u_k)\}_{k=1}^N$ measured by (1), we aim to identify models of the system in CT form

$$\hat{x}_{t+T_s} = \text{ODEsolve} \left\{ \frac{1}{\tau} f_\theta(\cdot, u_t), \hat{x}_t, T_s \right\}, \quad (3a)$$

$$\hat{y}_t = h_\theta(\hat{x}_t), \quad (3b)$$

and directly in DT form

$$\hat{x}_{k+1} = f_\theta(\hat{x}_k, u_k), \quad (4a)$$

$$\hat{y}_k = h_\theta(\hat{x}_k), \quad (4b)$$

where $\hat{x}_k \in \mathbb{R}^{n_x}$ is the model state, $\hat{y}_k \in \mathbb{R}^{n_y}$ is the model output, and $f_\theta : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ with $h_\theta : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_y}$ are feedforward multi-layer artificial neural networks¹ with $\theta \in \mathbb{R}^{n_\theta}$ collecting the activation weights as model parameters, $\tau \in \mathbb{R}_+$ is a normalization parameter, while ODEsolve corresponds to a chosen fixed-step numerical integration scheme, e.g., *4-step Runge-Kutta* (RK4), to compute the state-transition of the CT dynamics in the time interval $t \in [kT_s, (k+1)T_s]$. Under these considerations, both (3) and (4) represent recurrent neural networks, also called as ANN-SS models.

2.2 The SUBNET method

In the literature, many approaches have been introduced to identify ANN-SS models, e.g., (Masti and Bemporad,

¹ Consider that each hidden layer is composed from m activation functions $\phi : \mathbb{R} \rightarrow \mathbb{R}$ in the form of $z_{i,j} = \phi(\sum_{l=1}^{m_{i-1}} \theta_{w,i,j,l} z_{i-1,l} + \theta_{b,i,j})$ where $z_i = \text{col}(z_{i,1}, \dots, z_{i,m_i})$ is the latent variable representing the output of layer $1 \leq i \leq q$. Here, $\text{col}(\cdot)$ denotes the composition of a column vector. For f_θ with q hidden-layers and linear input and output layers, this means $f_\theta(\hat{x}_k, u_k) = \theta_{w,q+1} z_q(k) + \theta_{b,q+1}$ and $z_0(k) = \text{col}(\hat{x}_k, u_k)$.

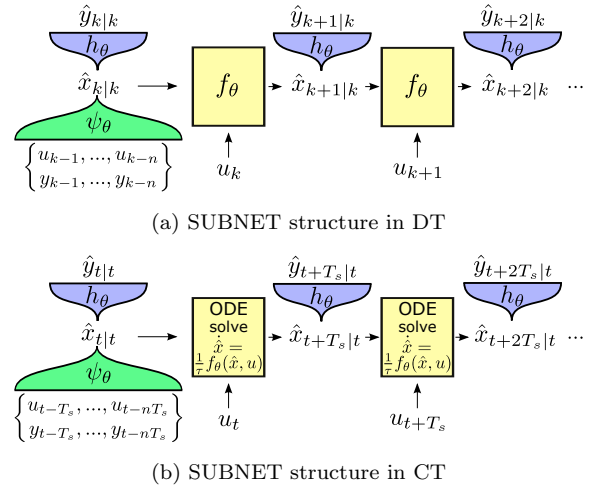


Fig. 1. SUBNET structure: the subspace encoder ψ_θ estimates the initial state at time k based on past inputs and outputs, then the state is propagated through f_θ and h_θ multiple times until a simulation length T .

2021; Forgiione and Piga, 2021). One recent approach that provides computationally efficient estimation of (3) and (4) under statistical consistency guarantees is the SUBNET approach (Beintema et al., 2023a,b). The SUBNET method, depicted in Fig. 1, uses a T -step ahead prediction cost over multiple subsections of the data, which is tractable and stable to optimize compared to the conventional one-step-ahead prediction error cost. This is formalized for DT as follows:

$$V_{\mathcal{D}_N}^{\text{enc}}(\theta) = \frac{1}{N_{\text{sec}}} \sum_{k \in \mathcal{I}_{N_{\text{sec}}}} \frac{1}{T} \sum_{l=0}^{T-1} \|y_{k+l} - \hat{y}_{k+l|k}\|_2^2, \quad (5a)$$

$$\hat{x}_{k+l+1|k} = f_\theta(\hat{x}_{k+l|k}, u_{k+l}), \quad (5b)$$

$$\hat{y}_{k+l|k} = h_\theta(\hat{x}_{k+l|k}) \quad (5c)$$

where $\mathcal{I}_{N_{\text{sec}}} \subseteq \{n+1, \dots, N-T+1\}$ s.t. $|\mathcal{I}_{N_{\text{sec}}}| = N_{\text{sec}}$ are the starting times of the sections and the pipe $(|)$ notation is used to denote time indexing inside of the section in terms of (current index|start index). In the CT version, k is replaced with t , l with $T_s l$ and (5b) becomes $\text{ODEsolve} \left\{ \frac{1}{\tau} f_\theta(\cdot, u_{t+T_s l}), \hat{x}_{t+T_s l|t}, T_s \right\}$. As prediction over the sections can be computed in parallel, this enables the use of powerful batch optimization algorithms such as Adam for training. Furthermore, SUBNET co-estimates an “encoder” function ψ_θ together with f_θ and h_θ , which provides a direct estimate of the initial state $\hat{x}_{k|k}$ for each section in (5) based on past input and output data, i.e.,

$$\hat{x}_{k|k} = \psi_\theta(u_{k-n}^{k-1}, y_{k-n}^{k-1}) \quad (5d)$$

where $u_k^{k+t} = [u_k^\top \dots u_{k+t}^\top]^\top$ for $t \geq 0$ and y_k^{k+t} is defined similarly. In fact, ψ_θ corresponds to learning a reconstructability map of the state from past IO samples, where n needs to be chosen at least the lag (n_x in the SISO case) of the NL system, see (Beintema et al., 2023a) for details. In case full state measurements are available, i.e., $y_k = x_k + e_k$, n should be chosen to be at least 1.

3. GENERAL GUIDELINES

In case of many vehicles, the position and heading angle do not explicitly influence the system dynamics. As a result, the kinematic relations concerning these states can be separated from the rest, and the overall network

structure of the SUBNET approach can be simplified by focusing the identification on the velocity dynamics and augmenting the resulting model with first-principle kinematic relations. In the case of some vehicles, e.g. aircrafts, where the altitude influences the aerodynamics, such separation can only be applied in a limited sense.

As the extrapolation capabilities of ANN models are also limited, it is important to provide sufficiently excited training data to explore the domain of interest in the state- and input space and bring out all essential system features. In Sect. 4 and 5 we present suitable trajectories for data generation in case of quadcopters and cars, respectively.

The SUBNET structure involves certain hyperparameters, the selection of which is system-dependent. Based on Beintema et al. (2023a), the following guidelines are available:

Model order: reasonable to start with twice as many states as in the first-principle dynamics, and then tune this value according to the validation loss. SUBNET structures with larger model order can capture more detailed dynamical behavior, which the first-principle models may not contain, but selecting n_x too large can cause overfitting.

Encoder length: should be chosen as the lag of the underlying system, which corresponds to the minimal number of past IO samples required to reconstruct the state in the conditional mean sense. For general nonlinear systems, reconstructability can require lags up to $2 \cdot n_x$.

Section length: T should be a few times the largest characteristic time scale $\lambda_{\max}$ of the data-generating system.

ANN structure: for simplicity, the structure of f_θ and h_θ can be kept identical, thus the same number of hidden layers (hl_{din}) and the same number of neurons per layer (n_{din}) can be applied. Similar notations can be introduced for ψ_θ as well (hl_{enc} and n_{enc}). It is recommended to start with $hl_{\text{din}} = hl_{\text{enc}} = 2$ and $n_{\text{din}} = n_{\text{enc}} = 64$.

Learning rate and batch sizes: $N_{\text{batch}} = 256 - 512$ is the baseline and the Adam optimizer with a fixed $\alpha = 10^{-3} - 10^{-4}$ learning rate is recommended. In case of over-fitting, ℓ_2 regularization is advised during training.

Derivative normalization: in CT, $\hat{x}$ and its derivative can have different magnitudes which can lead to difficulties in the estimation of f_θ . Hence, a derivative normalization factor $1/\tau$ is introduced, which can be selected as

$$\tau \approx \frac{(7.5 \pm 2.5) \cdot \lambda_{\max}}{0.03}. \quad (6)$$

4. IDENTIFICATION OF THE CRAZYFLIE 2.1 QUADCOPTER

4.1 Quadcopter simulator

First, we show the applicability of the SUBNET approach for the identification of a simulation model of the Crazyflie nano quadcopter. This model is based on the following coordinate frames: inertial frame F_i with *north-east-down* (NED) coordinates, vehicle frame F_v , and body frame F_b , which is fixed to the vehicle. The transformation from F_i to F_v is a translation, and from F_v to F_b a rotation, because the axes of F_v and F_i are considered to be parallel. The translational dynamics of the quadcopter are characterized by

$$m\ddot{r} = R_b^v \begin{bmatrix} 0 \\ 0 \\ -F \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}, \quad (7a)$$

where $r = [x \ y \ z]^\top$ is the position, $\dot{r} = [v_x \ v_y \ v_z]^\top$ is the lateral velocity in the inertial frame, m is the mass,

F is the collective thrust of the propellers, and g is the gravitational acceleration. $R_b^v \in \text{SO}(3)$ is the rotation matrix between the vehicle and body frames, where $\text{SO}(3)$ denotes the 3D special orthogonal group. The rotational dynamics are described based on Euler's equations as

$$\dot{R}_b^v = R_b^v \hat{\omega}_b, \quad (7b)$$

$$\dot{\omega}_b = J_b^{-1} (\tau - \omega_b \times J_b \omega_b), \quad (7c)$$

where $\omega_b = [p \ q \ r]^\top$ is the angular velocity of the vehicle in the body frame, J_b is the inertia matrix, and $\tau = [\tau_x \ \tau_y \ \tau_z]^\top$ is the torque vector in F_b produced by the propellers. The notion $\hat{\cdot}$ stands for the projection: $\mathbb{R}^3 \rightarrow \text{SO}(3)$ ensuring that $\hat{x}y = x \times y$ for all $x, y \in \mathbb{R}^3$, where $\times$ corresponds to the vector product. To achieve a more concise representation of (7), the rotation matrix R_b^v is mapped to Euler angles: roll, pitch, and yaw denoted as ϕ , θ and ψ respectively. To reproduce the dynamics of a Crazyflie 2.1, the parameters are set to: $J_x = J_y = 1.4 \times 10^{-5} \text{ kg} \cdot \text{m}^2$, $J_z = 2.17 \times 10^{-5} \text{ kg} \cdot \text{m}^2$, corresponding to the diagonal values in the diagonal inertia matrix and the off-diagonal values are 0, $m = 0.027 \text{ kg}$, and $g = 9.8 \text{ m/s}^2$.

4.2 Data generation

Data-generation based on simulation of (7) with a sampling frequency² $f_s = 48 \text{ Hz}$ is used for training and validation data, and 48, 96, and 192 Hz for the test data using RK4 numerical integration. As the system is unstable, flight-trajectories of (7) are generated by using a gain-scheduled *linear quadratic regulator* (LQR) controller, designed w.r.t. the local linearizations of (7) over its operating regime. The LQR is scheduled based on a desired state trajectory that is calculated for a x - y - z - ψ defined flight-path reference by taking advantage of the differential flatness property of the system (see Mellinger and Kumar (2011) for details). As we aim to discover the nonlinear behavior of the quadcopter, we designed flight paths that involve large values of roll and pitch angles, as for smaller angles the dynamics are close to linear.

Training data: To discover the nonlinear characteristics of the system, we generated the following flight-path references

$$[x_{\text{ref}}(t) \ y_{\text{ref}}(t) \ z_{\text{ref}}(t)]^T = R_\theta(t) R_\phi(t) e_x s(t), \quad (8a)$$

$$s(t) = r_d \sin(2\pi\omega_r t - \frac{\pi}{2}) + r_d, \quad (8b)$$

where $R_\theta(t)$ and $R_\phi(t)$ are $\text{SO}(3)$ matrices corresponding to rotation around the e_z and e_y axes respectively. The rotation angles are generated by $\phi(t) = 0.3\pi \sin(2\pi\omega_\phi t)$, and $\theta(t) = 2\pi\omega_\theta t$, while $\psi_{\text{ref}}(t)$ has been kept 0. Here t denotes time, while r_d , ω_r , ω_ϕ and ω_θ are the parameters of the trajectory. Multiple flights have been recorded under the choices of r_d, ω_ϕ and ω_θ detailed in Table 1, with r_d expressed in meters, and the angular velocity values expressed in Hz. Furthermore, for sufficient excitation, white Gaussian signals have been injected into the control inputs with standard deviations of $\sigma_F = 0.07 \text{ N}$ for the collective thrust and $\sigma_\tau = [2 \ 2 \ 0.2]^\top \cdot 10^{-4} \text{ N} \cdot \text{m}$ for the torque input. As measurement noise, Gaussian white noise has been added to the recorded states of the system. The variance of this noise has been chosen to achieve an aggregated SNR of 35 dB, 25 dB, and 15 dB.

² $f_s = 48 \text{ Hz}$ has been selected based on the time constants of the system to achieve a realistic behavior in the simulation.

Table 1. Parameters for the training (upper), validation (lower left), and test (lower right) trajectories, executed for 100, 100, and 10 sec.

#	ω_r	r_d	ω_ϕ	ω_θ	#	ω_r	r_d	ω_ϕ	ω_θ
1	0.20	0.5	0.0092	0.12	10	0.7	0.6	0.0087	0.12
2	0.25	0.45	0.11	-0.95	11	0.71	0.6	0.0088	0.12
3	0.33	0.45	0.012	-0.095	12	0.74	0.55	0.096	0.12
4	0.35	0.5	0.092	0.12	13	0.75	0.55	0.0089	0.12
5	0.45	0.45	0.010	-0.095	14	0.8	0.5	0.0087	0.12
6	0.45	0.5	0.0092	0.12	15	0.84	0.45	0.0085	0.12
7	0.55	0.5	0.0092	0.12	16	0.841	0.45	0.010	0.085
8	0.65	0.5	0.090	0.12	17	0.842	0.45	0.011	0.085
9	0.66	0.5	0.092	0.12	18	0.844	0.45	0.011	-0.095

#	ω_r	r_d	ω_ϕ	ω_θ	#	A	B	f_r
1	0.78	0.45	0.01	0.12	1	0.25	0.3	0.7

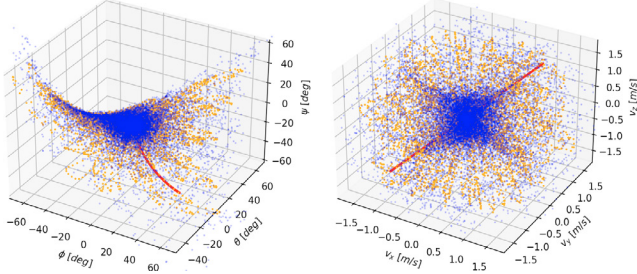


Fig. 2. Visualization of gathered data. Training, validation, and test sets are shown in blue, orange, and red.

Validation data: Trajectories have been generated similarly to the training data, with parameters outlined in Table 1, and added measurement noise.

Test data: Flight trajectories with

$$\begin{bmatrix} x_{\text{ref}}(t) \\ y_{\text{ref}}(t) \\ z_{\text{ref}}(t) \end{bmatrix} = \begin{bmatrix} -A \cos(2\pi f_r t) + A - vt \\ -A \cos(2\pi f_r t) + A + vt \\ -B \cos(2\pi f_r t) + B \end{bmatrix}, \quad (9)$$

have been generated with $v = 0.05$, defining the translational velocity perpendicular to the sinusoidal movement, A and B characterizing the amplitude, and f_r being the frequency of the oscillations, all specified in Table 1. The datasets are shown in Fig. 2. It is visible, that the system is excited well, and the validation set covers the same region as the training set. The test data is chosen to evaluate the performance of the model in parts of the state and input space less densely populated with training data points.

4.3 Training and results

The selected structures and hyperparameters are presented in Tables 2 and 3. Due to the assumed full state measurement, $n = 1$. By analyzing the validation losses, we have found that $n_x = 15$ results in better model performance compared to $n_x = 9$ of the first-principle model, which can be explained by immersion effects (Beintema et al., 2023a). Furthermore, as training on noise-free data resulted in overfitting, we employed ℓ_2 regularization of the parameters θ in (5a) with a weight $\lambda = 10^{-6}$.

Performance of the models is evaluated using *normalized root mean square* (NRMS) error of the simulated model response computed and averaged on all n -sample long sequences of the test data. As the CT SUBNET model can operate with various sampling frequencies, its performance is evaluated on test data sets with 48, 96, and 192 Hz sampling rates to showcase the generalization capability of the model. For a baseline, simulation results of the *linear time invariant* (LTI) model, corresponding to the local

Table 2. SUBNET model structure

Vehicle	Model	hl_{din}	n_{din}	hl_{enc}	n_{enc}	n_x	n	τ
Crazyflie	DT	2	64	2	64	15	1	-
	CT	2	64	2	64	15	1	5
F1tenth	DT	2	64	2	64	9	12	-
	CT	4	64	2	64	6	12	50

Table 3. Optimization parameters

Vehicle	Model	α	N_{batch}	T	Samples	Epochs	Time [h]
Crazyflie	DT	10^{-4}	512	80	138,720	3000	18
	CT	10^{-4}	512	80	138,720	1500	36
F1tenth	DT	10^{-3}	256	40	8,376	1000	1/4
	CT	10^{-3}	256	40	8,376	2000	2

Table 4. NRMS test losses of model estimates

Test	DT	CT				LTI
Freq	48 Hz	48 Hz	96 Hz	192 Hz	192 Hz	48 Hz
Step	160	160	160	320	160	640
Time	3.3 s	3.3 s	1.7 s	3.3 s	0.8 s	3.3 s
None	5.4%	5.2%	4.2%	6.0%	4.4%	7.4%
35 dB	5.9%	6.0%	5.0%	6.6%	6.1%	7.1%
25 dB	6.3%	6.3%	7.0%	7.1%	8.3%	7.9%
15 dB	12.1%	7.5%	8.2%	8.3%	10.9%	8.9%

linearization of the true NL model (7), are also provided. Linearization has been performed around hovering, that is, $x_* = 0$ and $u_* = [-mg \ 0 \ 0 \ 0]^T$.

The results³ are depicted in Table 4. While the DT and CT SUBNET methods show comparable performance, the latter has a slight edge and performs better across all noise levels. The distinction is especially notable at 15 dB SNR, showing that estimation in CT has better robustness against high SNR. The accuracy of the CT model deteriorates with increasing sampling frequency, its performance remains remarkable, with only 2 – 2.5% NRMS error increase at four times the sampling frequency. Responses of the estimated models are also displayed along a 160-step long simulation segment in Fig. 3 for both SUBNET structures. In Fig. 3, $|e_x/\sigma_x|$ stands for the absolute error of position x normed with the standard deviation of the measured x samples and the other subplots are similarly computed. The results show consistently low errors, except for the positions, which can be attributed to the integration process, where small errors in lateral velocity can result in quickly accumulating errors.

5. IDENTIFICATION OF THE F1TENTH CAR

F1tenth is a 1/10 scale model of an electric car, which has been mainly developed as a test platform for various automotive applications (Agnihotri et al., 2020). In this section, we present how to identify the dynamic relations of such a car, which can be useful for possible model-based control methods. In contrary to Sect. 4, measurements from a real F1tenth are used instead of simulated data.

5.1 Modelling considerations

One of the most common simplifications of car-like vehicles is the single-track model, often called the bicycle model (Paden et al., 2016). Multi-body dynamic models can be also considered (Althoff et al., 2017), but these higher

³ An attempt was made to compare the results with other well-known methods, as in Sect. 5. Due to the size of the dataset, running the selected benchmark algorithms was not feasible, and no subset of the data was found where the identification results were reasonable.

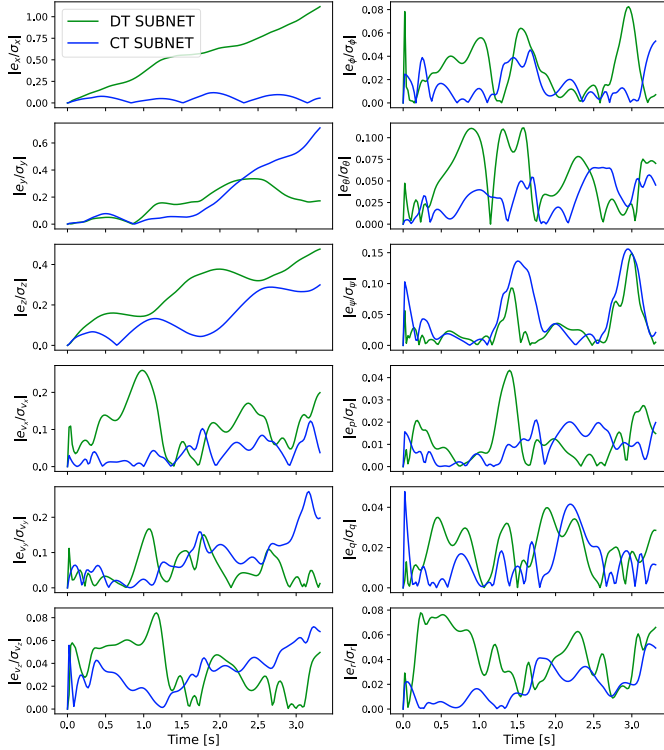


Fig. 3. Comparison of state errors along a test trajectory.

degree-of-freedom models are often too complicated to handle. Moreover, the mentioned first-principle methods, despite their complexity, are highly inaccurate for extreme maneuvering situations. We intend to show that black-box models offer a reliable and convenient way of identifying complicated vehicle dynamics through our experimental study with an F1tenth and the SUBNET approach.

The F1tenth car is displayed in Fig. 4. A brushless DC motor drives all four wheels of the vehicle, and the steering is controlled by a servo motor. A *Vedder Electronic Speed Controller* (VESC) unit is used to drive the actuators. The VESC unit also has a built-in IMU and speed sensor. For measuring the position and the orientation of the vehicle, an OptiTrack motion capture system is used that is capable of high-accuracy measurement of these variables based on passive markers placed on the car.

5.2 Data collection

The measured outputs have been selected based on the bicycle model: (x, y) position of the center of gravity in the reference plane (X, Y) , φ heading angle, that is measured from the X axis, v_ξ longitudinal velocity, v_η lateral velocity, and ω angular velocity (yaw rate). The two control inputs of the car are the motor voltage in terms of PWM percentage d and the steering angle δ . In line with Sect. 3, the kinematics of the vehicle are known, thus we only need to capture the velocity dynamics (v_ξ, v_η, ω) as the position and orientation can be computed using the kinematic single track model:

$$\dot{x} = v_\xi \cos \varphi - v_\eta \sin \varphi, \quad (10a)$$

$$\dot{y} = v_\xi \sin \varphi + v_\eta \cos \varphi, \quad (10b)$$

$$\dot{\varphi} = \omega. \quad (10c)$$

With the discussed measurement setup, all outputs and the two input signals are measured directly. To achieve robust identification results, two different trajectories have

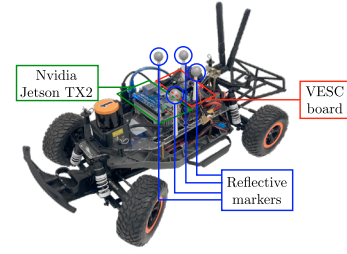


Fig. 4. F1tenth autonomous small-scale electric vehicle.

been used for training the network. A lemniscate-shaped trajectory has been selected, because by following it, the heading angle traverses the whole operational domain, and the resulting motion has quick changes in the velocity measures. The second trajectory has been chosen to be a circle-shaped path because it is also a typical maneuver for this type of vehicle.

Measurement data has been collected with a sampling frequency of $f_s = 40$ Hz. To acquire data with various motor PWM inputs, the velocity references have been varied for both trajectories ranging from 0.45 m/s to 1 m/s with step increments of 0.05 m/s, resulting in a total of 24 measurement records. Half of these measurements have been separated into training and test sets. Both trajectories with alternating reference velocities have been included in all data sets. Before concatenating the measurement signals for the training data set, 20% of the measured data was randomly selected to form a validation data set.

5.3 Hyperparameter selection

Identification of a model of the car dynamics has been performed with both the DT and the CT SUBNET methods. As previously discussed, the ANN model has been considered with two inputs and three outputs (as the position and orientation are calculated from the velocities). To find the optimal T and τ parameters, the time constants have been determined by exciting the car with step inputs and the outputs have been fitted by a first-order LTI model. The largest approximated time constant is $\lambda_{\max} = 0.282$ s. Based on the identified time constants, the values of T and τ have been determined according to Sect. 3.

Based on the validation losses, the model order n_x has been optimized to be 9 for the DT and 6 for the CT case. The encoder lag has been chosen to be $n = 12$, and the parameters of f_θ , h_θ , and ψ_θ have been selected according to the guidelines in Sect. 3. The *hyperbolic tangent* ($\tanh$) activation function has been chosen for all three neural networks, but using *rectified linear* (relu) activation functions has given similar results as well. For training the models, the Adam optimizer has been applied. The hyperparameter selection for the network structure and the optimizer are shown in Table 2 and Table 3.

5.4 Results

The resulting models are evaluated based on the validation and test errors on the full data sets using the NRMS error metric. Also, similar to Sect. 4, an average NRMS error is calculated on all 150-timestep (3.75 seconds) long segments of the test data. The results are summarized in Table 5, with other well-known nonlinear identification methods for comparison. With both the DT and CT SUBNET methods, the estimated models have achieved outstanding test results showcasing superior performance compared to

Table 5. Validation and test errors.

Model	Valid. NRMS	Test NRMS	150-step NRMS
DT SUBNET	6.54%	8.53%	5.08%
CT SUBNET	7.45%	8.99%	5.12%
PNL-SS	-	15.92%	218.06%
NARX-net	-	17.99%	36.97%

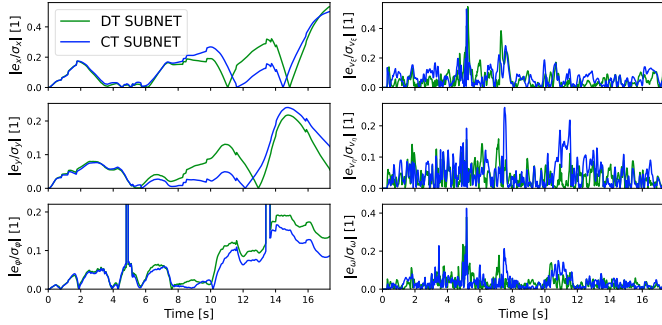


Fig. 5. Comparison of state errors along a test trajectory.

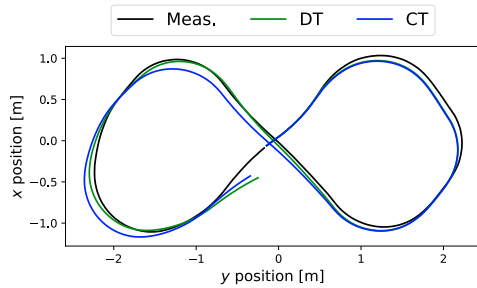


Fig. 6. Comparing the generated trajectories.

the benchmark algorithms. The PNL-SS model (Decuyper et al., 2021) produced especially inaccurate results in the 150-step simulations, because this method does not estimate the initial model state, thus $\hat{x}_0 \approx 0$ was assumed for all sections. A similar problem is visible for the NARX-net⁴, although this method uses measurements of past IO data, it is not as effective as the SUBNET structure.

A single lemniscate trajectory with 0.8 m/s reference velocity has been selected to illustrate the simulation performance of the models. The absolute errors of the simulation on this particular trajectory are shown in Fig. 5 for both SUBNET structures. Remarkably, the identified models provide an accurate simulation of the measured reality for 16 seconds. Although, some large peaks are visible on the heading error at some time instances. This is mainly because φ is wrapped to $[-\pi, \pi)$ and when transitions from $-\pi$ to π occur, temporary inaccuracies are present. The generated trajectories are also shown in Fig. 6. It is worth noting that the main source of the growing error of the position is that these values have been numerically integrated from the velocity signals, which has resulted in accumulating errors in the simulated trajectory.

6. CONCLUSION

In this paper, the capabilities of the new generation of deep-learning-based system identification schemes, such as the SUBNET approach, have been demonstrated on the identification of the dynamics of both an aerial and a ground vehicle. First, we have proposed general guidelines for SUBNET-based identification of vehicle dynamics, then following these recommendations we have achieved

state-of-the-art estimated model accuracy for both systems, proving the effectiveness of the identification procedure. Furthermore, the SUBNET method delivered nonlinear state-space models to enable the design of advanced controllers for the identified vehicles both in terms of predictive and feedback solutions, opening the door for high-performance personalized control of vehicles in practice.

REFERENCES

- Agnihotri, A., O'Kelly, M., Mangharam, R., and Abbas, H. (2020). Teaching autonomous systems at 1/10th-scale: Design of the f1/10 racecar, simulators and curriculum. In *Proc. of the 51st ACM Tech. Symp. on Comp. Sci. Edu.*, 657–663.
- Althoff, M., Koschi, M., and Manzing, S. (2017). CommonRoad: Composible benchmarks for motion planning on roads. In *Proc. of the IEEE Int. Veh. Symp.*, 719–726.
- Amiruddin, B.P., Iskandar, E., Fatoni, A., and Santoso, A. (2020). Deep learning based system identification of quadcopter unmanned aerial vehicle. In *Proc. of the 3rd Int. Conf. on Inf. and Com. Tech.*, 165–169.
- Bansal, S., Akametalu, A.K., Jiang, F.J., Laine, F., and Tomlin, C.J. (2016). Learning quadrotor dynamics using neural network for flight control. In *Proc. of the 55th IEEE Conf. on Dec. and Cont.*, 4653–4660.
- Beintema, G.I., Schoukens, M., and Tóth, R. (2023a). Deep subspace encoders for nonlinear system identification. *Automatica*, 156, 111210.
- Beintema, G., Schoukens, M., and Tóth, R. (2023b). Continuous-time identification of dynamic state-space models by deep subspace encoding. In *Proc. of the Int. Conf. on Learn. Rep.*, 1–15.
- Decuyper, J., Tiels, K., and Schoukens, J. (2021). PNLSS Toolbox 1.0. *IFAC-PapersOnLine*, 54(7), 374–378.
- Forgione, M. and Piga, D. (2021). Continuous-time system identification with neural networks: Model structures and fitting criteria. *Eu. Jour. of Cont.*, 59, 69–81.
- Hermansdorfer, L., Trauth, R., Betz, J., and Lienkamp, M. (2020). End-to-end neural network for vehicle dynamics modeling. In *Proc. of the 6th IEEE Cong. on Inf. Sci. and Tech.*, 407–412.
- Hoekstra, J.H., Cseppento, B., Beintema, G.I., Schoukens, M., Kollár, Z., and Tóth, R. (2023). Computationally efficient predictive control based on ANN state-space models. In *Proc. of the 62nd IEEE Conf. on Dec. and Cont.*, 6336–6341.
- Masti, D. and Bemporad, A. (2021). Learning nonlinear state-space models using autoencoders. *Automatica*, 129, 109666.
- Mellinger, D. and Kumar, V. (2011). Minimum snap trajectory generation and control for quadrotors. In *Proc. of the IEEE Int. Conf. on Rob. and Aut.*, 2520–2525.
- Paden, B., Čáp, M., Yong, S.Z., Yershov, D., and Frazzoli, E. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Tran. on Int. Vehic.*, 1(1), 33–55.
- Ribeiro, A.H., Tiels, K., Umenberger, J., Schön, T.B., and Aguirre, L.A. (2020). On the smoothness of nonlinear system identification. *Automatica*, 121, 109158.
- Saviolo, A. and Loianno, G. (2023). Learning quadrotor dynamics for precise, safe, and agile flight control. *An. Rev. in Cont.*, 55, 45–60.

⁴ Please refer to Matlab 2024a documentation for more details