2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Using Physics-Informed LSTM for Autonomous Vehicle Modelling

Mahmoud Selim[1,2], Jezdimir Milosevic[1], Sriharsha Bhat[1], Karl H. Johansson[2]

I. Introduction

Accurately modelling the dynamics of heavy duty vehicles such as trucks is essential for safe autonomous navigation. The dynamical model needs to capture complex system behaviour in various weather and road conditions as well as under different load configurations. This abstract outlines the integration of Physics-informed Long Short-Term Memory (PI-LSTM) networks as dynamical models within the context of motion planning and control for autonomous vehicles. By leveraging the predictive capabilities of LSTMs to model complex dynamics, and the generalizability imposed by adding the physics constraints in the loss function, we propose a framework for generating more efficient and reliable predictions that are tailored for motion planning and control.

The system identification problem for vehicle modelling aims to solve the following ordinary differential equation:

$$\dot{x} = f(x,u) \tag{1}$$

Depending on the formulation of f, existing methods can be classified roughly into three main categories: classical methods, learning methods, and residual methods.

Classical (white-box) modelling solves the problem of system identification using physics-based principles. In general, these models rely on simplifying assumptions to model the system. Although simple and efficient, they suffer from some major problems. First, many properties of the vehicle (or the road) may be quite difficult to model. For example, modelling how the road friction and aerodynamic forces affect the system is not an easy task. Thus, these classical approaches may be inaccurate in modelling complex physical phenomena due to these assumptions. Second, these models contain many parameters which describe the system characteristics (e.g.: mass, moment of inertia, friction coefficient, etc...) that are either hard to identify or varying due to changes in the system configuration (different loads affect mass and inertia, and weather conditions affect road friction.)

Learning-based (black-box) methods model the system entirely from observations. These methods prove to be quite successful in modelling complex physical phenomena. For example, [1] used Gaussian Processes (GPs) in a MPC framework to control a quadrotor. However, GPs are known to scale badly with the number of data points. Neural networks have also been used in system modelling. For example, [2] modelled a quadrotor dynamics in the context of motion control, and [3] used LSTMs to predict the movement of battleships in extreme sea states. Although these methods model the system well, how they generalize to different data distributions remains an un-explored direction.

[1] Scania CV AB, Södertälje, Sweden (mahmoud.selim, jezdimir.milosevic, sriharsha.bhat)@scania.com
[2] KTH Royal Institute of Technology, Stockholm, Sweden (mase2, kallej)@kth.se

Fig. 1. Tractor and semi-trailer truck. Courtesy of Scania CV AB.

Residual (grey-box) methods combine external knowledge and observations to enhance the learning as well as the generaliazability of the identified models. For example [4] used physics-inspired temporal convolutions to learn aggressive maneuvers of a quadrotor in various flight regimes, and [5] used a nominal (physics-based) model and trained a neural network to learn the residuals of this nominal model. In this work, we explore further in this direction and use Physics-Informed LSTMs (PI-LSTMs) as dynamical models in the context of planning and control.

II. Methodology

In this work, the goal is to approximate Eq. 1 using physics-informed LSTMs, and leverage past states and future inputs to predict the vehicle's full dynamic state in a fixed time horizon T. More formally, the system future states' trajectory can be given by

$$\widehat{X}_{+} = f_{NN}(X_{-}, U, \theta_{NN}) \tag{2}$$

where $X_{-} = [x_1, x_2, \ldots, x_n]$ is the matrix of past system states, $\widehat{X}_{+} = [x_{n+1}, x_{n+2}, \ldots, x_{n+T}]$ is the matrix of future system states predicted by the neural network, $U = [u_0, u_1, \ldots, u_{n+T-1}]$ is the matrix of the system inputs, θ is the neural network parameters, and F_{NN} is the neural network function. We approximate f by the encoder decoder LSTM-based architecture shown in Fig. 2. The idea is to provide past state-input trajectory to the encoder LSTM to output an encoded state. The encoded state is then used to initialize another LSTM (the decoder.) The decoder is also provided

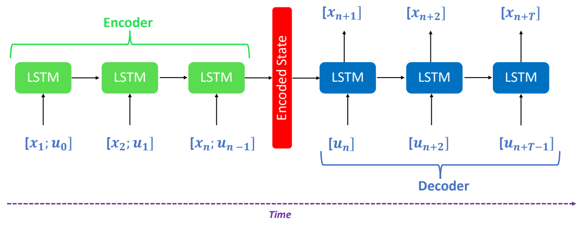

Fig. 2. Our architecture consists of two LSTMs: an encoder (green) and a decoder (blue). The encoder takes past state-input trajectory and outputs an encoded state vector (denoted in red). The decoder is then initialized using this state and predicts future states.

with future actions as input, and its goal is to predict the system's future states over a fixed time horizon T.

Training of the network. To train the network, we need to optimize its parameters with the goal of minimizing the loss function, and since we need to make sure that the network generalizes outside its training data, we add physics constraints as well. Thus, the loss function is as follows:

$$\underbrace{L}_{Total\ Loss} = \underbrace{||\dot{X}_+ - \widehat{X}_+||_2^2}_{data\ loss} + \lambda \underbrace{||\nabla f_{NN} - f||_2^2}_{Physics\ Loss} \tag{3}$$

where λ denotes how much we trust the physics-based (the nominal) model. If the nominal model is of high fidelity, we set λ to a high value, otherwise, we set λ to a small value to allow the optimization process to explore the data landscape more and give it higher importance. Note that the system model goes into the physics loss. However, it is removed for brevity. Then, the goal is to use gradient descent to find the neural network parameters θ that minimizes the loss function:

$$\theta_{NN} = \arg\min_{\theta} L \tag{4}$$

III. Experimental Setup and Results

We run our system against real world data collected from a truck with a trailer (see Fig. 1.) Training data consisted of 9 hours, and test data consisted of 1 hour and was collected throughout 2023. The training data was then split into training and validation with percentages of 90% and 10% respectively. All experiments were run on a workstation with a single Quadro RTX 4000 GPU and an Intel Xeon 12 core CPU. Both the encoder and the decoder had a hidden state vector with a dimensionality of 32 each. Since the state vector is of 5 dimensions, a 5-neuron feed-forward layer has also been added as a final prediction layer in the decoder. The goal is to predict the states of the system in a 10-second horizon using the future inputs and the state-action history trajectory. We found that a history of 1 second is enough and the system didn't improve any further by providing longer history trajectories. The system was sampled at 50 Hz. For data processing, only a low pass filter was applied to the signals before the training process. The system equation used in the loss term in Eq. 3 is a double track tractor-trailer model found in [6]. For the training parameters, we train the model for 1000 epoch with a batch size of 512. The learning rate is set to 0.003 and is decreased by one-third every one-third of the training length.

	PI-LSTM (ours)	PI-TCN	Nominal
R^2 Score	**.87**	.74	.58
MSE Loss	**.01**	.12	1.26

We compare our model against a nominal model and a temporal-convolution physics-inspired neural network (PI-TCN) [4]. For the training loss, only the MSE (Mean-Squared Error) is used, while for evaluation, all the methods were evaluated using both mean-squared error as well as R^2 score as given by Eq. 5:

$$R^2 = 1 - \frac{\sum(x_+ - \widehat{x}_+)}{\sum(x_+ - \mu)} \tag{5}$$

where μ is the mean of the groud truth. It is worth mentioning that we use R^2 as it provides a normalized version of MSE that doesn't depend on the scale of the target values. Results (Table. III) show the average MSE as well as R^2 score for both training and test data for all different methods. The proposed approach outperforms the other baselines in both MSE & R2 Score.

IV. Conclusion & Future Work

This work demonstrates the potential of using LSTMs to enhance the accuracy and efficiency of dynamical models in the context of planning and control. By accurately modeling complex dynamics, LSTMs enable more reliable and optimized predictions, paving the way for their broader adoption in industrial and research applications. For future work, we study the interpretability of data-driven dynamical models as well as their application in other domains such data-driven simulations.

References

[1] C. Peng and Y. Yang, "Trajectory tracking of a quadrotor based on gaussian process model predictive control," in *2021 33rd Chinese Control and Decision Conference (CCDC)*, pp. 4932–4937, IEEE, 2021.

[2] S. Bansal, A. K. Akametalu, F. J. Jiang, F. Laine, and C. J. Tomlin, "Learning quadrotor dynamics using neural network for flight control," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 4653–4660, IEEE, 2016.

[3] J. del Águila Ferrandis, M. S. Triantafyllou, C. Chryssostomidis, and G. E. Karniadakis, "Learning functionals via lstm neural networks for predicting vessel dynamics in extreme sea states," *Proceedings of the Royal Society A*, vol. 477, no. 2245, p. 20190897, 2021.

[4] A. Saviolo, G. Li, and G. Loianno, "Physics-inspired temporal learning of quadrotor dynamics for accurate model predictive trajectory tracking," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10256–10263, 2022.

[5] N. Mohajerin and S. L. Waslander, "Multistep prediction of dynamic systems with recurrent neural networks," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3370–3383, 2019.

[6] A. de Souza Mendes, M. Ackermann, F. Leonardi, and A. de Toledo Fleury, "Yaw stability analysis of articulated vehicles using phase trajectory method," in *Proceedings of DINAME 2017: Selected Papers of the XVII International Symposium on Dynamic Problems of Mechanics 17*, pp. 445–457, Springer, 2019.