2024 IEEE 27th International Conference on
Intelligent Transportation Systems (ITSC)
September 24- 27, 2024. Edmonton, Canada

Motion Planning Using Physics-Informed LSTMs for Autonomous Driving

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

***Abstract*— Developing accurate models for the behavior of heavy-duty vehicles such as trucks and buses is crucial for ensuring their safe navigation. It is important that these models accurately reflect the vehicle's performance across different weather conditions, road types, and cargo loads. This paper presents the use of Physics-Informed Long Short-Term Memory (PI-LSTM) networks as dynamic models tailored for sampling-based motion planning, which is relevant for the navigation of autonomous vehicles. By combining the LSTM's capability to predict nonlinear vehicle dynamics with physics-based constraints incorporated into the loss function, our planner will be able to generate motion plans that are not only safer due to improved dynamic accuracy but also efficient and capable of full parallel execution on a GPU to significantly enhance planning speed. Evaluating our model using real data from vehicle tests on different road surfaces and driving maneuvers, we see that PI-LSTMs capture vehicle behavior with a significantly lower error than traditional modelling techniques.**

Fig. 1: Tractor and semi-trailer truck in low friction conditions. Courtesy of Scania CV AB.

I. Introduction

Autonomous trucks and buses promise to revolutionize the transportation industry by enhancing safety through reduced human error, increasing efficiency through optimized routing and fuel consumption, improving overall traffic flow, and more importantly, addressing the shortage of drivers for heavy duty vehicles. One of the significant challenges in autonomous truck navigation is handling low friction surfaces, such as icy roads in winter, which can lead to dangerous situations such as jackknifing (it is the condition where a truck's trailer skids and forms an acute angle with the cab) or trailer swings which can lead to loss of control and accidents. These conditions demand advanced path planning strategies that take into account the reduced traction and increased stopping distances. To mitigate these risks, accurate vehicle models that simulate the truck's response to such hazardous conditions are essential. These models must accurately reflect the physics of vehicle-road interaction under low friction conditions, enabling the path planning algorithm to preemptively adjust routes, speeds, and driving strategies to prevent loss of control. However, one of the challenges that persist is the development of reliable and accurate vehicle models that can accurately predict vehicle behavior in real-time, adjust to changing conditions, and learn from new data to continually improve performance.

Vehicle modeling stands as a cornerstone in the development and deployment of autonomous vehicles, particularly for trucks and buses, which navigate a myriad of operational challenges daily. At its core, vehicle modeling involves the creation of mathematical models that simulate the real-world dynamics and behaviors of vehicles under various conditions. This is crucial for autonomous vehicles, as these models inform and guide their decision-making processes, ensuring that they can navigate safely, efficiently, and reliably. For heavy-duty vehicles like trucks and buses, which play a vital role in transportation and logistics, the stakes are even higher. These vehicles operate under a wide range of load configurations, traverse long distances across diverse terrains, and must adhere to stringent safety standards due to their potential impact in the case of an accident. Therefore, the accuracy and robustness of vehicle models directly influence the effectiveness of autonomous systems in predicting and responding to environmental changes, traffic conditions, and potential hazards.

Accurately modelling the dynamics of heavy duty vehicles such as trucks or buses is essential for safe navigation. The dynamical model needs to capture the complex system behaviour in various weather and road conditions as well as under different load configurations. This work outlines the integration of Physics-informed Long Short Term Memory networks (PI-LSTMs) as data driven dynamics models for autonomous vehicles within the context of sampling-based motion planning.

A. Related Work

1) Motion Planning: Motion (Path) planning for heavy-duty autonomous vehicles, such as trucks, involves devising a route that avoids collisions amidst static and dynamic obstacles, a task critical for their successful operation in complex

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

2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC) | 979-8-3315-0592-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ITSC58415.2024.10920067

environments. Numerous algorithms have been developed to address these challenges effectively. For instance, the Artificial Potential Field (APF) [1] method employs a potential function across the configuration space to guide the vehicle's movement, although it sometimes results in entrapment in local minima. Conversely, grid-based algorithms like A* and D* are renowned for their ability to identify an optimal path, termed "resolution optimal," since the granularity of map discretization affects the determination of such paths. However, the computational resources and memory required for these algorithms escalate exponentially with increases in map size and the complexity of the vehicle's state space.

Sampling-based motion planning techniques, such as Rapidly-exploring Random Trees (RRT) [2] and Probabilistic Roadmaps (PRM), have been widely adopted in robotics for navigating complex environments. However, their efficiency and effectiveness are often limited by the accuracy of the underlying dynamical models. Considering autonomous driving, the receding horizon nature of sampling-based motion planning requires rolling out the vehicle dynamical models several times over long prediction horizons. Traditional vehicle models usually rely on simplifying assumptions and may not adequately capture the intricacies of real-world dynamics, leading to suboptimal planning. In this work, we propose using Physics-Informed light-weight LSTM networks to learn the vehicle dynamics from real-world data and enhance dynamical modeling within this domain. We also adapt the traditional RRT implementation to be able to generate high fedility dynamically-feasible paths in parallel leverging the power of GPUs, hence, improving the accuracy as well as the speed of the algorithm.

2) System Identification: The system identification problem for vehicle modelling has been extensively studied in the literature. It aims to identify a mathematical function f that well expresses the system dynamics. The general form of the system equations can be expressed as follows:

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

where:

- $x \in \mathbb{R}^n$ represents the state vector of the system, including both the primary vehicle and its articulated components. The state vector typically encompasses positions, orientations, velocities, and angular velocities.
- $u \in \mathbb{R}^m$ denotes the control input vector, which might include steering angles, throttle positions, and braking forces.
- $\dot{x}$ is the time derivative of the state vector, representing the system's dynamics.
- $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a nonlinear function mapping the current state and control inputs to the rate of change of the state.

Depending on the formulation of f, existing methods be classified roughly in three main categories: classical (white-box) methods, learning-based (black-box) methods, and residual (grey-box) methods.

Classical (white-box) modelling solves the problem of system identification using first (physics-based) principles. In general, these models rely on simplifying assumptions to model the physical system. Although simple and efficient, traditional models face significant challenges. Accurately modeling vehicle or road properties, such as friction and wind resistance, is complex, leading to simplifications that work for low speeds or typical driving conditions but fail to capture more intricate physical phenomena. Furthermore, these models depend on numerous parameters—like mass, moment of inertia, and friction coefficients—that are not only difficult to measure but also vary with changes in system configuration, such as different loads or weather conditions affecting road friction.

Learning-based (black-box) methods on the other hand, model the system entirely from observations (data). These methods treat the modelling task as a time-series [3] or a multi-step prediction [4] problem. They prove to be quite successful in modelling complex physical phenomena. For example, [5]–[7] used Gaussian Processes (GPs) in a MPC framework to control a quadrotor. However, GPs are known to scale badly with the number of dimensions. With the rise of deep learning, there have been other attempts to use data driven methods. Neural networks have proven to be quite successful in modelling system dynamics [8]–[12]. For example, [13] used neural networks to learn quadrotor dynamics in the context of control, [14] used temporal convolutions for quadrotor motion prediction. LSTMs [15] have proved to be quite successful in dynamic modelling. For example, [16] used LSTMs to predict battle ships movement in extreme sea states. Although these methods model the system well, how they generalize to different data distrubtions remains an un-explored direction.

Residual (grey-box methods) combine external knowledge as well as data models to enhance the learning as well as the generaliazability of the models. For example [17] used physics-inspired temporal convolutions to learn agressive maneuvers of a quadrotor in various flight regimes. [18] used the nominal model and trained a neural network to learn the residuals of this nominal model. It's worth mentioning that most of the previous work has been on the application of these methods in the context of control. In this work, we explore further in this direction and use Physics-Informed LSTMs as dynamical models in the context of planning for autonomous vehicles.

B. Contributions and Organization

The contributions of this work are as follows:

1) We develop a physics-informed LSTM (PI-LSTM) dynamical model for modelling vehicle dynamics.
2) We compare the developed model against other baselines using real-world data collected in different terrain and extreme weather conditions.
3) We evaluate the applicability of the PI-LSTM in a sampling-based motion planning framework to generate more accurate and more efficient plans.

The organization of this paper is as follows: in section II we go through the preliminaries and problem formulation. In section III we explain the methodology of our framework, in section IV we explain the experimental details and discuss the results. Finally, in V we conclude our work and point out future directions for our work.

II. Preliminaries & Problem Formulation

A. Path Planning For Autonomous Vehicles

Planning with dynamic models is essential for generating dynamically feasible paths that adhere closely to the physical constraints of real-world systems. In sampling-based planners, an extensive number of action samples are generated, forming a large set of trajectories. These trajectories are integrated using dynamic models to compute possible states, which are then evaluated using a cost function to determine the effectiveness of each proposed action. Unfortunately, the models most commonly in use today are overly simplistic, including basic point mass models and simple kinematic or dynamic models, which do not adequately capture the complex dynamics needed for accurate planning when we consider low-friction surfaces and adverse weather. These models lack the fidelity necessary for precise and reliable predictions. To overcome these limitations, there is a pressing need for high-fidelity dynamic models that not only offer more accurate depictions but also leverage ongoing data collection to continuously improve and update the models over time, ensuring they remain effective in the face of changing conditions and advanced planning requirements. This process underscores the need for robust evaluation metrics and the ability to quickly generate and assess multiple trajectories to ensure safe and effective navigation. Next, we state a nominal model for truck and trailer combinations that is a baseline for our work.

B. System Equations for Articulated Vehicle Dynamics

The dynamics of an articulated vehicle (in our case a truck and trailer) are governed by a set of differential equations that takes into account the vehicle's mass distribution, external forces, and control inputs to model the system's evolution over time

For our system, we model the vehicle as a single track tractor-trailer model as shown in Fig. 2 that is built upon [19]. The specific form of the function f depends on the detailed modeling of the vehicle dynamics, including considerations for the vehicle's mass matrix $\mathbf{M}(x)$ and force vector $F(x,u)$, which can be articulated as:

$$\mathbf{M}(x)\dot{x} = F(x,u), \tag{2}$$

where $\mathbf{M}(x)$ is the mass matrix accounting for the inertia of the vehicle and its load distribution, and $F(x,u)$ represents the sum of external and internal forces acting on the system. The exact equations for $\mathbf{M}$ and F as well as the parameters of the model are expanded in the Appendix. Solving these equations allows the prediction of vehicle behavior under various conditions, vital for the design and analysis

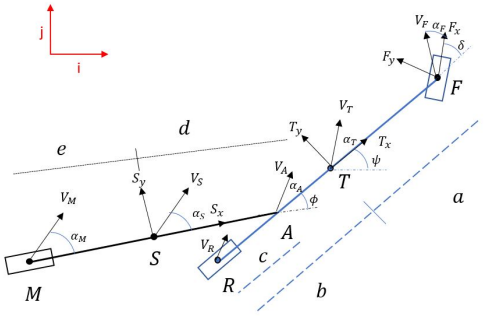

Fig. 2: A mathematical model for a truck, denoted blue, and a trailer, denoted black. The system is modelled using a single track bicycle model, which lumps together the left and right wheels into a unified wheel along the center-line for simplified analysis. The trailer connects to the tractor at the articulation point, denoted as point A. Additionally, tire models are integrated to accurately simulate the forces acting upon the vehicle, enhancing the model's precision in depicting real-world dynamics.

of autonomous systems. However, for precise system state predictions, it is necessary to consider additional vehicle components such as gearboxes, transmission systems and other complex modules that induce nonlinearities. These elements are intricate and typically not suited for real-time planning applications. Furthermore, accurately modeling the interaction between vehicle tires and varying road surfaces, particularly in low-friction scenarios, poses significant challenges due to the variability in friction coefficients, which depend on different tire materials. The primary objective of this work is to develop a representation that is physics-aware, scalable, rapidly adaptable to new vehicle types, and efficient for planning purposes.

III. Methodology

In this work, our objective is to approximate the system equation (Eq. 1) using Physics-Informed Long Short-Term Memory Networks (PI-LSTMs), utilizing historical state information and future inputs to predict the vehicle's full dynamic state over a fixed time horizon T. The predicted future states of the system are described by:

$$\widehat{X}_+ = f_{NN}(X_-, U, \theta_{NN}), \tag{3}$$

where $X_- = [x_1, x_2, \ldots, x_n]$ represents the matrix of past system states, $\widehat{X}_+ = [\widehat{x}_{n+1}, \widehat{x}_{n+2}, \ldots, \widehat{x}_{n+T}]$ denotes 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 system inputs, and θ_{NN} are the parameters of the neural network function f_{NN}.

To enhance the prediction capabilities of our model, we incorporate a history of past states through an encoder LSTM architecture. This method leverages the concept of time delay-embedding, based on Takens' embedding theorem [20], which suggests that a rich feature space can be constructed using delayed embeddings of time series measure-

ments. This approach, widely used in the analysis of chaotic systems, allows for a more comprehensive reconstruction of state spaces, providing a deeper contextual basis for predictions. Additionally, Recurrent Neural Networks (RNNs) and LSTMs, particularly in their continuous-time form, have demonstrated universal approximation capabilities for dynamical systems in state-space model representations [21], [22]. It has been shown that any finite-time trajectory of a given n-dimensional dynamical system can be approximated by the internal state dynamics of an RNN with n output units, supplemented by hidden units and suitably chosen initial conditions.

The system employs an encoder-decoder LSTM architecture, as depicted in Fig. 3. The encoder LSTM receives the past state-input trajectory and outputs an encoded state representation. This encoded state then initializes a second LSTM, the decoder, which is also supplied with future actions. The primary objective of the decoder is to predict the system's future states within the fixed time horizon T. We note that having a fixed time horizon, which is more suited for motion planning, allows for the training of more effective LSTMs compared to using a variable time horizon.

A. Training of the Network.

The training of the network involves optimizing its parameters to minimize a specific loss function. To ensure that the model generalizes beyond its training data, physics-based constraints are integrated into the loss function, which is defined as:

$$\underbrace{L}_{\text{Total Loss}} = \underbrace{\sum_{i=N+1}^{N+T} ||x_i - \widehat{x}_i||_2^2}_{\text{Data loss}} + \lambda \underbrace{\sum_{i=N+1}^{N+T} ||\nabla f_{NN,i} - f_i||_2^2}_{\text{Physics Loss}} \quad (4)$$

where T is the trajectory length, λ weighs the importance of the physics-based model fidelity within the overall loss, $\nabla f_{NN,i}$ is the gradient of the neural network at time step i of the receding horizon with respect to its input, and f_i is the system physics as denoted in Eq. 1. It's important to mention that a higher λ value is used when the nominal model is highly reliable, allowing the optimization to focus on adherence to physical laws. Conversely, a lower λ emphasizes data-driven exploration in the model's learning process.

Ultimately, the objective is to use gradient descent techniques to identify the neural network parameters θ_{NN} that minimize the loss function:

$$\theta_{NN} = \arg\min_{\theta} L. \quad (5)$$

This formulation allows for precise, data-informed predictions while adhering to fundamental physical principles, ensuring both accuracy and robustness in the model's predictive performance.

B. Integration With Motion Planning

The developed PI-LSTM model fits into a new modular approach to path planning that can run easily on a GPU. More specifically, Our proposed planner in Fig. 4 consists of three consecutive modules: a trajectory sampler that samples actions for a whole trajectory, a PI-LSTM neural dynamics model, and a cost function that calculates the cost for each trajectory separately. The planning can be done as follows: first, we sample a large number of action samples (tens of thousands of trajectories of actions.) In our case, each action sample is a tuple consisting of longitudinal acceleration and front wheel steering angle. Then, using our PI-LSTM dyamical model we predict the next states for each of the trajectories using these sampled actions as input to our PI-LSTM. It's worth mentioning this whole planner is designed to run on a GPU in an end to end manner. Trajectories of action samples are passed as a batch and the predicted states are generated simultaneously. Then, the planner evaluates these trajectories and selects the best trajectory. The evaluation of the trajectories is done using a cost function applied to the predicted states of the receding horizon. This cost function takes into consideration typical motion planning metrics such as safety, violation of traffic rules, comfort, and progress towards the goal. Moreover, these trajectories have accurate state predictions and dynamically feasible paths thanks to how the PI-LSTM network was developed. In this paper we evaluate the performance of the PI-LSTM in the context of future integration into this planning framework.

IV. Experimental Setup and Results

A. Experimental Setup

In our experiments, we aim to learn an accurate dynamical model of a 37.5 ton, 17m long Scania truck and trailer (see Fig. 1). Data is collected both by driving the vehicle autonomously and using the assistance of a professional safety driver. State feedback signals are obtained from inertial and navigation sensors from tests conducted on dry asphalt, wet roads, snow and ice in winter conditions. In addition, we process the recorded state trajectories using a 4th order Butterworth lowpass filter with a cutoff frequency of 5 Hz to eliminate high-frequency noise. We scale all input features to the neural network between [-1, 1] to ensure that the model attributes equal importance to each data component. Our dataset comprises 10 hours of vehicle trajectory data collected from tests conducted in 2023. To ensure robustness and diversity, we selected four segments, each spanning two weeks, for model evaluation. These segments encompass a wide range of driving scenarios, including forward movement, left and right turns, ascents and descents, U-turns, roundabouts, and varying weather conditions such as rain and snow. Additionally, to further assess model performance under low-friction driving conditions, the testing dataset is augmented with additional winter test data gathered from Scania's proving ground in northern Sweden during winter 2024.

All learning and system identification experiments were run on a workstation with a single Quadro RTX 4000 GPU and an Intel Xeon 12 core CPU. The training of the neural network as well as the other baselines was done using PyTorch in Python while both testing and evaluation were

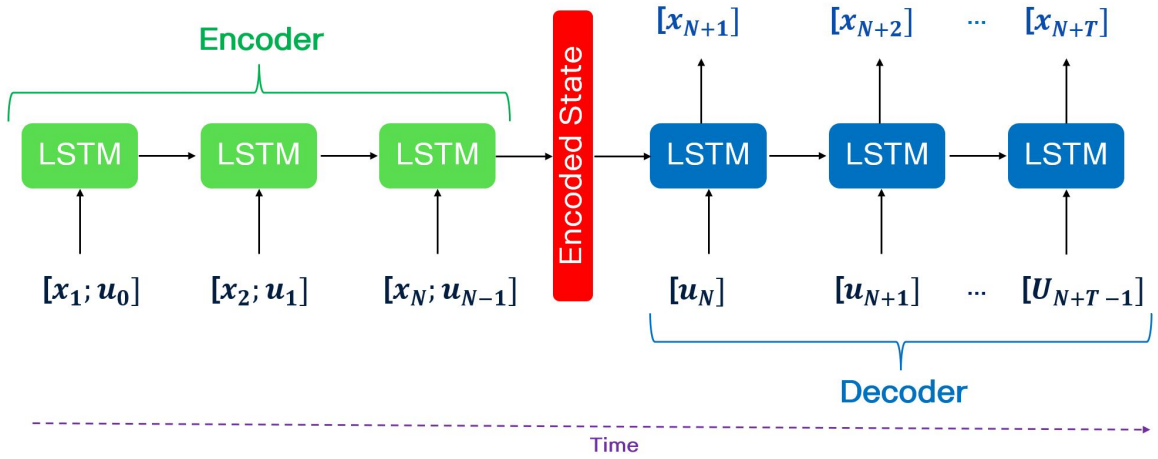

Fig. 3: Our architecture consists of two LSTMs: an encoder (green) and a decoder (blue). The encoder takes past state-input trajectory and outputs an encoded state vector (denoted in red). The decoder is then initialized using this state and predicts future states.

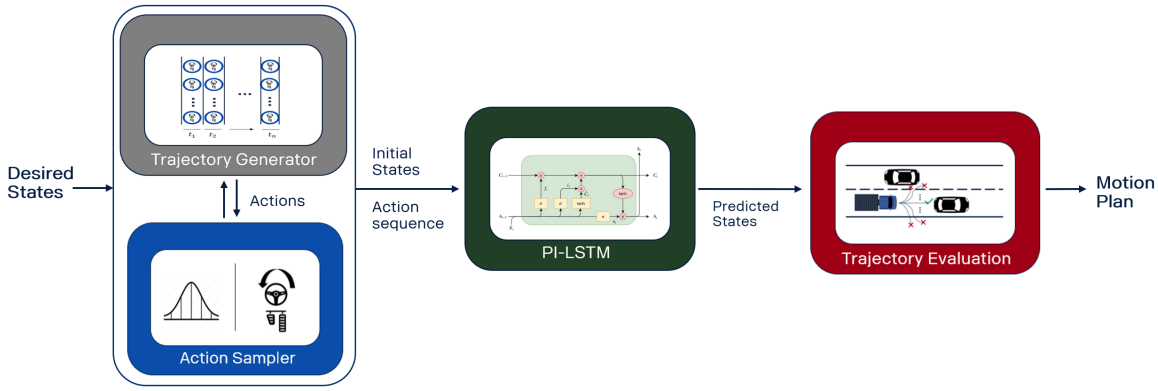

Fig. 4: The proposed planner architecture is a modular architecture and can be executed on a GPU efficiently. It consists of an action sampler that samples actions (acceleration, and steering). A trajectory sampler queries this action sampler to generate full length trajectories. These trajectories are then passed to our PI-LSTM to predict the future states. Finally, a cost function evaluates all the trajectories and selects the best one.

done in C++ using TensorRT to optimise the performance of the neural network.

Both the encoder and the decoder each had a hidden state vector with a dimension of 32. The state vector is of 5 dimensions. So, 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. Here, a history of 1 second was seen to be sufficient (longer history trajectories did not improve the system performance further). The system equation used in the loss term in Eq. 4 is a single track tractor-trailer model found in [19]. For the training parameters, we train the model for 1000 epochs 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. The training loss is shown in Fig. 5.

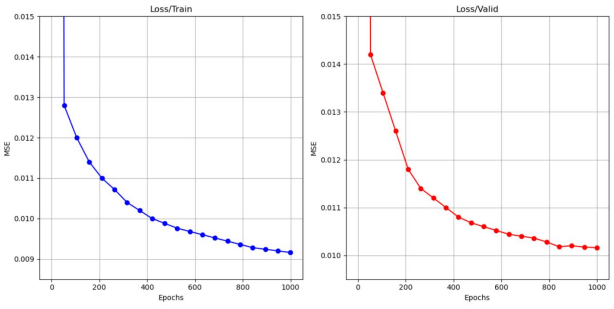

Fig. 5: Train (left) and validation (right) loss for our PI-LSTM.

B. Experimental Results

Our experiments are designed to address the following key questions:

TABLE I: Experimental Results.

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

1) How accurate is our model compared to other baseline models?
2) Does incorporating physics-based loss enhance the model's ability to learn more accurate representations of the dynamics?
3) Is the incorporation of the system's history through Takens' embedding beneficial for improving model performance?
4) Considering the planner integration, is it feasible for the neural network to operate in real-time?

To assess the fidelity of our model, we compare our model against a nominal model, based on Eq. 2, and a temporal-convolution physics-inspired neural network (PI-TCN) [17]. 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. 6:

$$R^2 = 1 - \frac{\sum(x_+ - \widehat{x}_+)}{\sum(x_+ - \mu)} \tag{6}$$

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. I) 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 metrics. A qualitative result is also shown in Fig. 6. We plot the predictions of a 10-second trajectory for the longitudinal and lateral acceleration as well as the yaw rate. The PI-LSTM model matches the reference signals more closely than the other two methods. To assess the importance of the physics loss, we conducted experiments training our encoder-decoder LSTM both with and without incorporating the physics loss. We noticed that removing the physics loss results in a 0.15 lower R^2 in comparison to incorporating physics loss. We observed that including the physics loss leads to improved validation accuracies. Furthermore, we explored the effect of varying history sizes on the LSTM's performance. As shown in Fig, 7, one can note that with a history size of approximately 50 samples, the benefits plateau, beyond which no further improvements in network performance are observed. Finally, we also study the validation error with respect to the model size as shown in Fig. 8. We see a trend of diminishing returns as the network size grows. We conclude that either a hidden size of 32 or 64 is enough for our application.

The successful integration of our LSTM-based dynamic model into a sampling-based path planning framework hinges on two critical factors. The first factor is model accuracy. Our experiments demonstrate that our model significantly outperforms traditional nominal models, ensuring dynamic

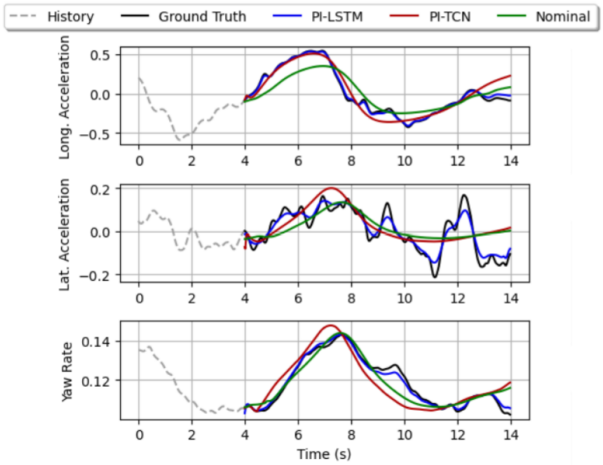

Fig. 6: 10-second open-loop prediction of the longitudinal and lateral accelerations (m/s^2) as well as the yaw rate (rad/s) for the three methods. PI-LSTM (blue) is better at predicting the states than the other methods.

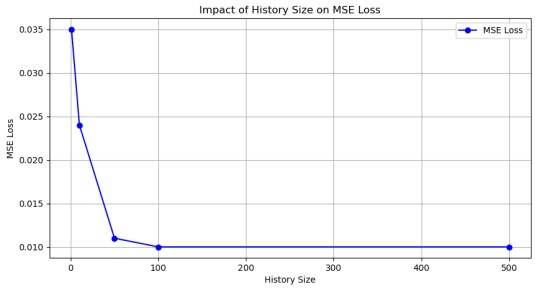

Fig. 7: Effect of history length in our PI-LSTM on the MSE loss

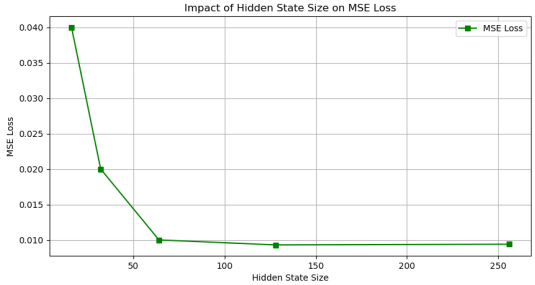

Fig. 8: Effect of varying hidden state sizes in our PI-LSTM on the MSE loss

feasibility across various driving and weather conditions. The second factor is the model's capability for real-time operation. For sampling-based motion planners, optimizing for throughput (the number of trajectories processed within a fixed time frame) is more crucial than latency (the processing time of a single trajectory). The performance of these planners typically improves with an increased number of samples or trajectories. We tested the model's inference speed for

real-time applicability, and our results show that we could process 500,000 trajectories, each spanning 12 seconds, at a rate of 20 Hz. This performance comfortably satisfies our criteria for real-time functionality, affirming the model's suitability for real-time path planning.

V. Conclusion & Future Work

This work highlights the potential of using LSTMs to enhance the accuracy and efficiency of dynamical models in planning, ultimately prioritizing overall safety. By precisely modeling complex dynamics, LSTMs enable more reliable and optimized predictions, leading to safer motion plans and facilitating their wider adoption in both industrial and research applications. Future work will focus on improving the interpretability of data-driven dynamical models, exploring more advanced architectures, and extending the application of data-driven dynamic models to other domains, such as data-driven simulations.

Acknowledgement

We would like to express our sincere gratitude to Vinnova Sweden for their generous financial support for the AllDrive project, which made this research possible. Their funding was instrumental in facilitating our study and achieving our research objectives.

References

[1] C. W. Warren, "Global path planning using artificial potential fields," in *1989 IEEE International Conference on Robotics and Automation*, pp. 316–317, IEEE Computer Society, 1989.

[2] S. M. LaValle, J. J. Kuffner, B. Donald, *et al.*, "Rapidly-exploring random trees: Progress and prospects," *Algorithmic and computational robotics: new directions*, vol. 5, pp. 293–308, 2001.

[3] J. C. Principe and J.-M. Kuo, "Dynamic modelling of chaotic time series with neural networks," *Advances in neural information processing systems*, vol. 7, 1994.

[4] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Precision flight control for a multi-vehicle quadrotor helicopter testbed," *Control engineering practice*, vol. 19, no. 9, pp. 1023–1036, 2011.

[5] 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.

[6] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with gaussian processes," in *2016 IEEE international conference on robotics and automation (ICRA)*, pp. 491–496, IEEE, 2016.

[7] G. Cao, E. M.-K. Lai, and F. Alam, "Gaussian process model predictive control of an unmanned quadrotor," *Journal of Intelligent & Robotic Systems*, vol. 88, pp. 147–162, 2017.

[8] J. Atuonwu, Y. Cao, G. Rangaiah, and M. Tadé, "Identification and predictive control of a multistage evaporator," *Control Engineering Practice*, vol. 18, no. 12, pp. 1418–1428, 2010.

[9] O. Nerrand, P. Roussel-Ragot, D. Urbani, L. Personnaz, and G. Dreyfus, "Training recurrent neural networks: Why and how? an illustration in dynamical process modeling," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 178–184, 1994.

[10] I. S. Baruch and C. R. Mariaca-Gaspar, "A levenberg–marquardt learning applied for recurrent neural identification and control of a wastewater treatment bioprocess," *International Journal of Intelligent Systems*, vol. 24, no. 11, pp. 1094–1114, 2009.

[11] A. Delgado, C. Kambhampati, and K. Warwick, "Dynamic recurrent neural network for system identification and control," *IEE Proceedings-Control Theory and Applications*, vol. 142, no. 4, pp. 307–314, 1995.

[12] V. A. Akpan and G. D. Hassapis, "Nonlinear model identification and adaptive model predictive control using neural networks," *ISA transactions*, vol. 50, no. 2, pp. 177–194, 2011.

[13] 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.

[14] S. Looper and S. L. Waslander, "Temporal convolutions for multi-step quadrotor motion prediction," in *2022 19th Conference on Robots and Vision (CRV)*, pp. 32–39, IEEE, 2022.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*, pp. 366–381, Springer, 2006.

[21] A. M. Schäfer and H. G. Zimmermann, "Recurrent neural networks are universal approximators," in *Artificial Neural Networks–ICANN 2006: 16th International Conference, Athens, Greece, September 10-14, 2006. Proceedings, Part I 16*, pp. 632–640, Springer, 2006.

[22] X. Chen, Y. Tao, W. Xu, and S. S.-T. Yau, "Recurrent neural networks are universal approximators with stochastic inputs," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

APPENDIX

NOMINAL DYNAMIC MODEL

The expansion of the terms representing the vehicle dynamics in Eq. 2 is provided here following the work present in [19].

The $\mathbf{M}$ matrix in Eq. (2) is a 4×4 matrix in our case and can be denoted as:

$$\mathbf{M}=\begin{pmatrix} M_{11} & M_{12} & M_{13} & M_{14}\\ M_{21} & M_{22} & M_{23} & M_{24}\\ M_{31} & M_{32} & M_{33} & M_{34}\\ M_{41} & M_{42} & M_{43} & M_{44}\end{pmatrix}$$

where:

$$\begin{aligned}
M_{11} &= (m_T+m_S)\cos(\psi+\alpha_T),\\
M_{12} &= -(m_T+m_S)V_T\sin(\psi+\alpha_T),\\
M_{13} &= m_S((b-c)\cos(\psi+\phi)+d\cos(\psi-\phi)),\\
M_{14} &= -m_S d\sin(\psi-\phi),\\
M_{21} &= (m_T+m_S)\sin(\psi+\alpha_T),\\
M_{22} &= (m_T+m_S)V_T\cos(\psi+\alpha_T),\\
M_{23} &= -m_S((b-c)\cos(\psi)+d\cos(\psi-\phi)),\\
M_{24} &= m_S d\cos(\psi-\phi),\\
M_{31} &= -m_S((b-c)\sin(\alpha_T)+d\cos(\alpha_T+\phi)),\\
M_{32} &= -m_S((b-c)V_T\cos(\alpha_T)+dV_T\cos(\alpha_T+\phi)),\\
M_{33} &= m_S((b-c)^2+2(b-c)d\cos(\phi)+d^2)+I_S+I_T,\\
M_{34} &= -m_S((b-c)d\cos(\phi)+d^2)+I_s,\\
M_{41} &= m_S d\sin(\alpha_T+\phi),\\
M_{42} &= m_S dV_T\cos(\alpha_T+\phi),\\
M_{43} &= -m_S(d^2+(b-c)d\cos(\phi))+I_S,\\
M_{44} &= m_S d^2+I_S.
\end{aligned}$$

TABLE II: Description of Symbols

Symbol	Description
m_T	Mass of the tractor [kg]
m_S	Mass of the semi-trailer [kg]
I_T	Moment of inertia of the tractor [kgm^2]
I_S	Moment of inertia of the semi-trailer [kgm^2]
a	Distance between the tractor center of mass and the front wheel [m]
b	Distance between the tractor center of mass and the rear wheel [m]
c	Distance between point A and the rear wheel of the tractor [m]
d	Distance between the trailer center of mass and the front wheel [m]
e	Distance between the trailer center of mass and the rear wheel [m]
δ	The steering angle of the front wheel [rad]
α_T	The slip angle of the tractor's front wheel [rad]
α_M	The slip angle of the trailer's rear wheel [rad]
ϕ	The angle between the center lines of the tractor and the trailer [rad]
ψ	The yaw angle of the tractor [rad]
V_T	The velocity at the tractors center of mass [m/s]
V_S	The velocity at the trailers front wheel
F_x	Longitudinal force at front axle [N]
F_r	Longitudinal force at rear axle [N]
F_s	Longitudinal force at semi trailer axle [N]

The parameters used in the system dynamics equation 2 are described in Table II. This table provides a detailed explanation of each symbol used in the equations, along with their corresponding physical interpretations and units.

The force vector $\mathbf{F}(x,u)=(F_1,F_2,F_3,F_4)^T$ is expressed as:

$$\begin{aligned}
f_1 &= F_{x,F}\cos(\psi+\delta)+F_{x,R}\cos\psi+F_{x,M}\cos(\psi-\phi)-\\
&\quad -F_{y,F}\sin(\psi+\delta)-F_{y,R}\sin\psi-F_{y,M}\sin(\psi-\phi)-\\
&\quad -m_S(b+c)\dot\psi^2\cos\psi-m_S d\left(\dot\psi-\dot\phi\right)^2\cos(\psi-\phi)\\
&\quad -(m_T+m_S)v_T\sin(\psi+\alpha_T)\dot\psi\\
f_2 &= F_{x,F}\sin(\psi+\delta)+F_{x,R}\sin\psi+F_{x,M}\sin(\psi-\phi)+\\
&\quad +F_{y,F}\cos(\psi+\delta)+F_{y,R}\cos\psi+F_{y,M}\cos(\psi-\phi)-\\
&\quad -m_S(b+c)\dot\psi^2\sin\psi-m_S d\left(\dot\psi-\dot\phi\right)^2\sin(\psi-\phi)-\\
&\quad (m_T+m_S)v_T\cos(\psi+\alpha_T)\dot\psi\\
f_3 &= F_{x,F}a\sin\delta+F_{x,M}(b+c)\sin\phi+F_{y,F}a\cos\delta-\\
&\quad -F_{y,R}b-F_{y,M}\left[(b+c)\cos\phi+(d+e)\right]-\\
&\quad -m_S(b+c)d\left(\dot\psi-\dot\phi\right)^2\sin\phi+m_S(b+c)d\dot\psi^2\sin\phi+\\
&\quad +m_S\left[(b+c)v_T\cos\alpha_T+dv_T\cos(\alpha_T+\phi)\right]\dot\psi\\
f_4 &= F_{y,M}(d+e)-m_S(b+c)d\dot\psi^2\sin\phi-m_S dv_T\cos(\alpha_T+\phi)\dot\psi.
\end{aligned}$$