Physics-Guided Approach with Transfer Learning in Vehicle Lateral Dynamics

Fabien Lionti
Université de Côte d'Azur - INRIA
2004 Rte des Lucioles, 06902 Valbonne
fabien.lionti@inria.fr

Nicolas Gutowski
Université d'Angers - LERIA
2 Bd de Lavoisier, 49000 Angers
nicolas.gutowski@univ-angers.fr

Sébastien Aubin
Direction Générale de l'Armement
Route de Laval - 49245 Avrillé
sebastien.aubin@intradef.gouv.fr

Philippe Martinet
Université de Côte d'Azur - INRIA
2004 Rte des Lucioles, 06902 Valbonne
philippe.martinet@inria.fr

***Abstract*— Simulating long state variable trajectories using neural networks models is a complex problem that requires auto-regressive prediction. In this context, poor generalization capabilities increase the risk of error accumulation over time. Physics-guided machine learning is an emerging field that combines the strengths of physics-based modeling with data-driven approaches by incorporating physical knowledge into machine learning models. The aim of this paper is to tackle the problem of long-horizon simulation using neural network models that exploit prior knowledge about underlying physical processes under noisy and disturbed data acquisition. This paper introduces a robust methodology for training a multi-layer perceptron (MLP) model using noisy and disturbed system measurements. Our approach leverages a custom loss function designed to minimize auto-regressive prediction errors across extended time horizon, ensuring alignment between predicted trajectories and actual state variable measurements. We further enhance model performance through transfer learning, utilizing simulated data generated from a pre-existing dynamical model where parameters have been estimated using Bayesian inference techniques allowing to leverage prior knowledge in the transfer-learning process. Comprehensive evaluation demonstrates the robustness of our method on the case of the lateral dynamic of vehicle. We assessed our methodology on real-world measurements and show its competitiveness over traditional methods.**

I. Introduction

Accurate dynamical models are essential for performing numerical simulations, which are crucial tools in various engineering tasks. Numerical simulations enable engineers to predict and analyze the behavior of complex systems under different conditions without the need for extensive real world experiments. This capability is essential for designing and optimizing engineering systems, performing safety assessments, and developing control strategies [1]. However, obtaining accurate dynamical models that can provide reliable long-horizon simulations remains a challenging problem [2]. The complexity of real-world systems, coupled with uncertainties in model parameters and external disturbances, often leads to significant deviations between simulated and actual system behaviors over extended time horizon due to poor generalization capabilities and increasing error propagation over time. Physics-based approaches to dynamical system modeling involve deriving equations that describe system behavior based on first principles. However, designing these models manually can be tedious, as it requires a deep understanding of the involved physical principles. To make the problem tractable, these models often rely on simplifying assumptions that introduce biases, limiting their accuracy and applicability in real-world scenarios, especially for long-horizon simulations. Despite these limitations, Physic-based approaches have the advantage of relying on small amounts of data to estimate a limited number of parameters that are clear, interpretable, and fully describe the model behavior. On the other hand, learning-based approaches represent a general-purpose method for modeling dynamical systems. These methods have high modeling capabilities and can capture complex system behaviors with relatively few assumptions. This flexibility allows them to adapt to a wide range of applications and system dynamics. However, this flexibility also comes with a significant risk of overfitting [3] and generating solutions that do not adhere to physical principles outside the training dataset domain. Obtaining such models requires datasets that cover the entire state and control space of the system. This dependence on exhaustive datasets can be a limitation in scenarios where data is scarce or expensive to obtain. The field of physics-guided machine learning [4] offers a promising approach by combining the strengths of both physics-based modeling and data-driven methods. This synergy aims to achieve accurate dynamical models, offering improved performance in scenarios with small data or unbalanced distributions. However, efficiently merging both sources of information can be challenging, as significant biases in prior knowledge can contribute to degrade the overall performance of the final model. Each previously mentioned modeling method relies on data that, in a real-world context, involves dealing with noisy, disturbed, and unevenly spread measurements. These challenges make it difficult to capture accurate system behavior. Consequently, the scope of this paper is twofold: first, to propose a robust methodology for training a neural networks architecture on noisy and disturbed data acquisition effectively; and second, to ensure enhanced generalization capabilities for long-horizon simulations by leveraging prior knowledge about the

underlying physics in context of the vehicle lateral dynamic. The contributions presented in this paper are as follows:

1) An approach is introduced to integrate prior physical knowledge into the neural network training process using transfer learning techniques. This enables the transfer of knowledge from a source dataset, obtained from simulations of a known physical model, to a target dataset comprising real-world data.
2) A fully-differentiable auto-regressive loss function is proposed during the fine-tuning process to ensure robust training of the neural network.
3) Simulation data is generated using a physics-based model with parameters previously estimated from real-world measurements. Bayesian inference techniques are employed to incorporate prior knowledge at the parameter level of the physical model during estimation.
4) A comprehensive evaluation is provided demonstrating that the proposed method outperforms existing approaches in long-term simulation.

The paper is organized as follows: Section II reviews the related work and section III sheds light on the proposed method. Section IV depicts how its empirical evaluation was conducted while section V discusses our experimental evaluation. Furthermore, Section VI provides a discussion on limitations. Finally, a section for conclusion and perspectives brings the paper to a close.

II. Related work

System identification is a longstanding problem in the field of systems engineering. In the case of nonlinear dynamical systems, it involves solving ill-posed inverse problems due to the multiplicity of likely parameter sets for physical models [5]. Identifying these parameters from data often involves optimizing least squares problems [6], which is equivalent to finding the set of parameters that maximizes the likelihood of aligning predicted state variable trajectories with system observations under the Gaussian error. In the case of poorly informative data, the multiplicity of solutions can lead to parameter estimates that do not extend beyond the range of measurements. To address this challenge, Bayesian inference methods [7] have been widely used, as they allow the incorporation of prior knowledge about parameter distributions and act as constraints to maintain physically meaningful estimates in limited initial data. Approximate Bayesian inference methods, such as the Approximate Bayesian Computation (ABC) algorithm [8], are likelihood-free techniques that provide a straightforward way to estimate posterior probabilities of model parameters when dealing with intractable likelihood functions due to nonlinear dynamical systems and non-Gaussian noise uncertainties. These methods have demonstrated their efficacy in various parameter estimation problems, ranging from modeling the lateral dynamics of vehicles [9] to cases of calibrating image simulations in cosmology [10].

Learning-based methods offer a flexible approach to directly learning complex dynamical phenomena with minimal assumptions about the underlying process. MLPs are known as universal function approximators [11] and have been successfully used for system dynamics identification [12]. However, this flexibility requires a sufficient quantity of data covering all state and control spaces of the system to ensure generalization capabilities. Recently, ResNet architectures [13] have been recognized for their similarity to Euler forward numerical integration methods for differential equations [14]. This analogy offers an opportunity to optimize neural network architectures over multi-step time integrations using automatic differentiation frameworks.

To improve the generalization performance of learning-based methods, physics-guided machine learning aims to incorporate prior physical knowledge into machine learning models through various mechanisms. One classical approach is to directly learn a model of discrepancies between handcrafted models and system measurements [15]. This approach enhances performance by learning the missing physics corresponding to the discrepancies. Other methods rely on physics-informed loss functions that constrain the learning process to adhere to physical principles, such as energy conservation, as demonstrated in Hamiltonian neural networks [16]. Direct incorporation of physical knowledge into neural networks has been investigated in [17], where the form of the Lagrangian equation is integrated into the neural network architecture, showing improved performance over standard architectures. Transfer learning [18] involves leveraging knowledge gained from one task learned from a source dataset to enhance learning on a related task with a target dataset. This typically entails reusing features or representations learned from the source task to aid learning on the target task, which may have less data available but related characteristics. This approach has been investigated in the context of robotics to adapt simulation domains to real-world applications [19] showing improved learning speed in context of robotic haptic problems.

III. Proposed method

A. Overview

In the context of Transfer Learning, the method proposed in this paper leverages a source dataset $\mathcal{D}_{\text{src}}$ and a target dataset $\mathcal{D}$. The approach transfers knowledge from simulation data $\mathcal{D}_{\text{src}}$, generated using a nonlinear dynamical bicycle model, to a target task of learning lateral vehicle dynamics based on real-world data $\mathcal{D}$. This is achieved through a robust training formulation, comprising three key steps:

- The first step involves estimating parameters θ from the bicycle model $f(x_t, u_t, \theta)$, where $x_t \in \mathbb{R}^n$ denotes the state and $u_t \in \mathbb{R}^m$ denotes the control applied at a time step t. θ represents a set of fixed parameters that need to be estimated, but prior beliefs about their distribution are available. This parameter estimation is performed using Bayesian inference, leveraging the system measurement dataset $\mathcal{D}$ and the prior physical model of the lateral dynamics f along with its associated parameter range beliefs $P(\theta)$.

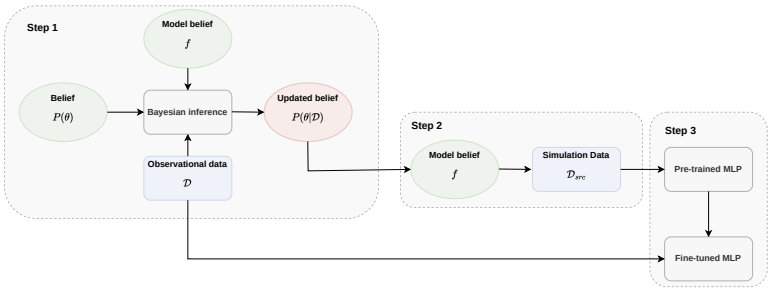

Fig. 1: Description of the proposed method

- The second step involves generating simulation data $\mathcal{D}_{src}$ using f with the set of parameters that maximize the posterior $P(\theta|\mathcal{D})$.
- This dataset $\mathcal{D}_{src}$ is then used to pretrain a MLP which is then fine-tuned on measurement data $\mathcal{D}$ using an autoregressive training formulation.

B. Physical model of the vehicle lateral dynamic

The non-linear dynamical bicycle model $f(x_t, u_t, \theta)$ [20] depends on a state vector x_t composed of the lateral speed v_y and the yaw rate $\dot{\psi}$. The control vector u_t includes the longitudinal speed v_x and the front steering angle δ. The model behavior depends on a set of parameters θ, which includes variables such as the vehicle's mass m expressed in kg, the center of gravity position defined by the front axle l_f and rear axle l_r distances in meters, the moment of inertia I_z expressed in $kg.m^2$, and the unitless Pacejka coefficients for the front $\{B_f, C_f, D_f, E_f\}$ and rear $\{B_r, C_r, D_r, E_r\}$ tires [21]. This model, depicted in Figure 2, assumes that the vehicle is symmetric and operates under steady-state conditions with no load transfers during maneuvers and at a constant velocity v_x. It also assumes that the car body motion is static, neglecting pitch and roll dynamics.

$$\dot{v}_y = \frac{1}{m}\left(F_{yf}\cos(\delta) + F_{yr} - m v_x \dot{\psi}\right) \tag{1}$$

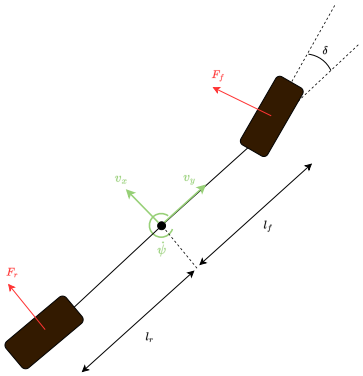

Fig. 2: Non-linear dynamical bicyle model

$$\ddot{\psi} = \frac{1}{I_z}\left(F_{yf}\cos(\delta) - F_{yr}\right) \tag{2}$$

Tire forces F_{y_k} with index $k \in \{f, r\}$ corresponding to front and rear wheel are provided by the Pacejka tire model :

$$F_{y_k} = D_k \sin(C_k \tan^{-1}(B_k\alpha_k - E_k(B_k\alpha_k - \tan^{-1}(B_k\alpha_k)))) \tag{3}$$

The tire model input depends on the tire slip angle α_f and α_r. Tire slip angles are computed with the following set of equations :

$$\alpha_f = \delta - \tan^{-1}\left(\frac{v_y + l_f\dot{\psi}}{v_x}\right) \tag{4}$$

$$\alpha_r = -\tan^{-1}\left(\frac{v_y - l_r\dot{\psi}}{v_x}\right) \tag{5}$$

C. Parameter identification

The first step of the proposed method involves identifying the parameters θ of the bicycle model using Bayesian inference. This approach allows to combine prior information $P(\theta)$ about the parameters with a likelihood function $P(\mathcal{D}|\theta)$, which represents the probability of observing the dataset $\mathcal{D}$ given the parameters θ. This combination yields the posterior distribution $P(\theta|\mathcal{D})$, which reflects updated belief about the parameters after taking the data into account.

$$P(\theta|\mathcal{D}) = \frac{P(\mathcal{D}|\theta)P(\theta)}{P(\mathcal{D})} \tag{6}$$

To perform this inference task, a variant of the Approximate Bayesian Computation (ABC) method known as ABC-SMC (Sequential Monte Carlo) [22] is employed. This method is based on an iterative refinement a population of particles representing system parameter. The initial population is sampled from the prior distribution of the parameters and iteratively filtered based on a distance criterion ρ. At each iteration p, a new set of particles is obtained by resampling accepted particles from the previous iteration using an importance sampling scheme. New particles are then obtained by randomly perturbing the resampled particles using a kernel distribution. Following this principle, the particles progressively converge towards an approximation

of the posterior distribution. The role of the distance ρ is to iteratively select parameters that are likely to correspond to parameters from the posterior. For small ϵ, the sets of parameters generate simulation data $\mathcal{D}_\theta$ with the model f that is close to $\mathcal{D}$, approximating the posterior:

$$P(\theta|\mathcal{D}) = \lim_{\epsilon \to 0} P(\rho(\mathcal{D}, \mathcal{D}_\theta) < \epsilon \ |\theta) \tag{7}$$

where ϵ represents the final threshold in a sequence of P decreasing thresholds, denoted as $\{\epsilon_0, \epsilon_1, \ldots, \epsilon_{P-1}, \epsilon\}$. At each iteration, these thresholds are applied to guide the selection of parameter sets. Here, $\mathcal{D}_\theta$ refers to the set of predicted trajectories $\{\hat{x}_0^n, \hat{x}_1^n, \ldots, \hat{x}_T^n\}$, while $\mathcal{D}$ represents the corresponding set of measured state variable trajectories $\{x_0^n, x_1^n, \ldots, x_T^n\}$. The index n denotes different state variable trajectories, each spanning T time steps generated using the forward Euler integration scheme. The constant Δ_t is the time step size used in the integration :

$$\begin{cases} \hat{x}_{t+1}^n = \hat{x}_t^n + f(\hat{x}_t^n, u_t^n, \theta)\Delta_t & (8) \\ \hat{x}_0^n = x_0^n & (9) \end{cases}$$

The distance metric ρ is defined as the Root Mean Squared Error (RMSE) between predictions and observations over T steps of trajectories, averaged over all N trajectories:

$$\rho(\mathcal{D}, \mathcal{D}_\theta) = \frac{1}{N} \sqrt{\frac{1}{T} \sum_{t=0}^{T} \left(\frac{\hat{x}_t^n - x_t^n}{\sigma_x} \right)^2} \tag{10}$$

To avoid favoring any particular state variable, the distance is standardized by σ_x, the standard deviation computed over each state variable of $\mathcal{D}$.

D. Transfer Learning

We define the MLP architecture as follows:

$$\dot{x}_t = \phi(x_t, u_t, \beta) \tag{11}$$

where β represents the parameters of the MLP model. To generate simulation data for the pre-training phase, fifteen particles corresponding to the last population P of the ABC-SMC that minimize ρ are retained. Those particles corresponds to the subset of parameters θ_{MAP} that maximizes the posterior distribution $P(\theta|\mathcal{D})$. Using these selected particles a simulation dataset $\mathcal{D}_{src}$ is generated. This is done by randomly sampling initial state variables x_t and control variables u_t from $\mathcal{D}$, and then generating one step ahead predictions x_{t+1} using the function f with a randomly sampled θ from the set θ_{MAP}. The dataset $\mathcal{D}_{src}$ thus captures the variability in predictions from f that corresponds to uncertainty in maximum a posteriori estimate. During both pre-training and fine-tuning of the MLP architecture, the optimization is based on the following loss function:

$$\mathcal{L}(\beta) = \frac{1}{T} \sum_{t=0}^{T} \left(\hat{x}_t - x_t \right)^2 \tag{12}$$

In the pre-training phase, as $\mathcal{D}_{src}$ does not include disturbances the MLP is optimized for a single time step by setting $T = 1$. In the fine-tuning phase, the MLP starts from the parameterization obtained from the pre-training phase. During this phase, the training process involves optimizing the auto-regressive trajectory generation using ϕ based on the forward Euler integration scheme over multiple time steps T. This process leverages automatic differentiation over involved numerical integration operations as described in [14].

IV. Experimental setting

A. Data acquisition

Measurement dataset $\mathcal{D}$ were acquired from a Renault Premium truck using Correvit S-Motion DTI optical sensors from Kistler to measure longitudinal speed v_x and lateral speed v_y. Yaw rate $\dot{\psi}$ was measured using an Alma inertial measurement unit. Data were collected at various speeds ranging from 15 to 80 km/h on a test track by executing different maneuvers, including slalom, avoidance maneuvers, and constant radius turns. The total duration of the recorded data is approximately 20 minutes.

B. Comparison and parameter setting

Comparisons are provided between the transfer-learning method, the bicycle model, an MLP trained solely on the training measurements, and an MLP that learned the discrepancy of the bicycle model as presented in [23]. The dataset $\mathcal{D}$ consists of 50% of the available measurements allocated for training and estimation for each method. The remaining measurements are divided into 30% for validation and 20% for testing. Each set was created by randomly selecting samples. Each method is evaluated using different combinations of parameters, including MLP complexities defined by the MLP depth $\times$ number of neurons per hidden layer, specifically $\{4 \times 8, 4 \times 16, 4 \times 24\}$. Additionally, various integration time in secondes for the training loss are considered $\{0.05, 0.5, 1\}$. For all MLP training, the Adam optimizer is used with a learning rate of 10^{-3}.

C. Evaluation metric

Evaluation of each compared methods involves computation of the RMSE between measured and predicted state variable trajectories over a ten-second time horizon. Predicted trajectories are generated autoregressively with a time step size $\Delta_t = 0.05$, corresponding to $T = 200$ forward Euler steps. To compute RMSE, predictions of the state variables start from an initial state x_0, which corresponds to the initially measured state of the test set sequence. Predictions are then produced using the same control commands u_t as the one in the test set measurements, the RMSE is defined as:

$$\text{RMSE} = \frac{1}{M} \sqrt{\frac{1}{T} \sum_{t=0}^{T} (\hat{x}_t^m - x_t^m)^2} \tag{13}$$

M is set to 20 and represents the number of trajectory in the test set.

D. Prior parameter setting

For bicycle parameter estimation using ABC-SMC methods, the prior distribution for each parameter in θ is modeled as an independent Gaussian probability density function. Table I lists the mean μ and standard deviation σ values associated with each Gaussian prior distribution, corresponding to each parameter in θ vector. The mass parameter $m = 18781kg$ has been directly measured using a scale These parameters are based on domain expert knowledge in vehicle dynamics and are validated against external sources. For the Pacejka tire coefficients, the prior means are set as $B \geq 0$, $C \geq 0$, $D \geq 0$, and $E \leq 0$ to avoid unrealistic tire model behavior, in line with the guidelines from [20]. The prior for the mass parameter m is determined from direct measurements of the vehicle. For the moment of inertia parameter I_z, the prior mean is calculated using the prism method, as described in [24].

V. Results

A. Bicycle model performance

The aim of this section is to evaluate the quality of the parameter estimate of the lateral dynamic bicycle model using the ABC-SMC algorithm with different integration time. Table II presents the performance of the top fifteen parameter sets θ_{MAP}, which maximize the posterior distribution $P(\theta|D)$. The performance of these parameters is evaluated in terms of the average RMSE, computed for each of the fifteen parameter sets, and for each state variable, v_y and $\dot{\psi}$. The results demonstrate that longer integration times used to compute the distance metric lead to a reduction in RMSE and greater stability, as evidenced by the decrease in the associated standard deviation. Increasing the number of integration steps allows for comparing trajectories over longer time horizons using ρ, resulting in a distance metric that is less sensitive to noise fluctuations as the signal-to-noise ratio improves with longer trajectories. However, increasing the number of steps requires greater computing power to compute ρ for each tested particle at each iteration of the ABC-SMC algorithm.

B. Transfer-learning performances

The previous section provides information about the predictive accuracy and associated variability related to the estimated parameters of the bicycle model. The final performances of both the transfer-learning and discrepancy modeling methods are dependent on the quality and variability of predictions of the bicycle model with assessed with parameters from θ_{MAP}. To take into account this dependence in the final results, context of the transfer-learning approach, the performances of the fine-tuned MLP, depends on pre-trained on data simulated from f with parameters from the set θ_{MAP}. For the discrepancy modeling the experiments are repeated fiftneen times, each with a different parameterization from of θ_{MAP} the bicycle model contained in this set. Table III shows the performance in terms of RMSE for each state variable, considering different integration times and MLP complexities. It compares MLPs with random initialization, MLPs with transfer learning starting from different parameterizations β obtained from repeated pretraining processes using θ_{MAP}, and similarly, repetitions of the discrepancy learning method based on different parameterizations of the bicycle model. For each compared method, as in the previous section, increasing integration time is associated with a reduction in the RMSE and its variability, enhancing the robustness of the loss function against noise fluctuations. Additionally, increasing MLP complexity from 8 to 16 and 24 neurons per hidden layer is associated with improved performance for each method. The transfer learning method shows a smaller RMSE and standard deviation over each state variable. Discrepancy modeling shows higher RMSE compared to the learning-based method, which is our baseline as it represents the case where no additional information other than dataset $\mathcal{D}$ is used. A hypothesis for this loss of performance is an unstable gradient during training due to the dependence between bicycle model predictions that are fed into the MLP, which is responsible for learning to correct its predictive bias. As these steps involve auto-regressive dependence, backpropagating the gradient through this scheme is likely to lead to predictions from the MLP that are outside the classical range of states with physical meaning for the bicycle model, leading to an unstable gradient. The Figure 3 shows the evolution of the averaged RMSE over each trajectory of the test set for each state variable, for increasing lengths of autoregressive prediction. This is done for the parameter set for each method that yielded the lowest RMSE for each state variable in Table III. For the learning-based, transfer learning, and discrepancy methods, we observe an increase in error at the beginning of the simulation, followed by a slight attenuation and stabilization over time of the RMSE on the yaw rate state variable. This pattern seems to indicate an error accumulation due to random fluctuations in the measured trajectory that cannot be learned by any method. Over time, the properties of these random fluctuations remain constant,

Parameter	I_z	l_f	l_r	B	C	D	E
μ	150000	5	5	7.1	3	3	-2
σ	50000	1	1	2	2	2	1

TABLE I: Prior distribution parameters. The mean (μ) and standard deviation (σ) are provided for each parameter.

Integration Time (s)	v_y	$\dot{\psi}$
0.05	0.1416 ± 0.0144	0.0116 ± 0.0022
0.50	0.1354 ± 0.0060	0.0101 ± 0.0011
1.00	$\mathbf{0.1185 \pm 0.0051}$	$\mathbf{0.0083 \pm 0.0009}$

TABLE II: RMSE computed over 10-second trajectories from the test set for various training integration times. RMSE values are provided for the lateral speed v_y and yaw rate $\dot{\psi}$ state variables. Uncertainties are given as $\pm$ one standard deviation.

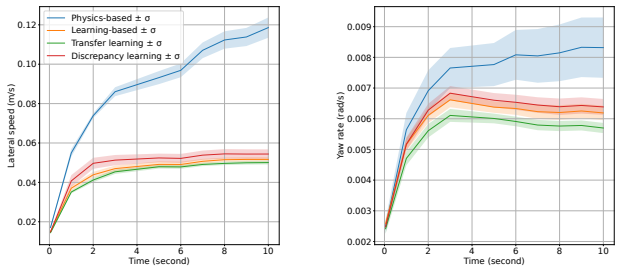

Fig. 3: Evolution of RMSE for each state variable v_y and $\dot{\psi}$ over 10 secondes trajectories for each compared method

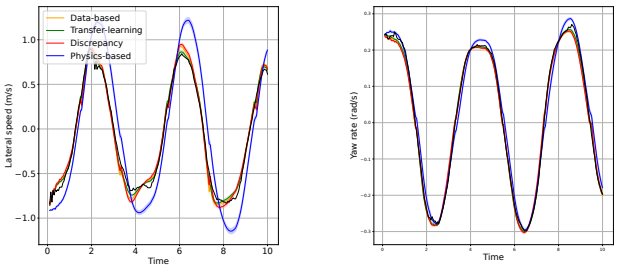

Fig. 4: Trajectory simulation of 10-second duration for a slalom path. Transparent areas correspond to $\pm 0.1\sigma$ around the predicted mean

and the predictions of each method are valid up to a constant prediction bias, implying RMSE stabilization over time. In the case of the lateral speed state variable, we observe a similar pattern, but the RMSE tends to be in slightly increase tendency at the end of the trajectory, indicating error accumulation over time. For the bicycle model, due to its inability to capture all phenomena present in the measurements, error accumulate without any perceptible stabilization. Figure 4 presents the trajectory simulated over a temporal horizon of 10 seconds for each method, focusing on the v_y and $\dot{\psi}$ state variables during a slalom maneuver. The bicycle model demonstrates degraded predictions for each state variable, with a significant bias in v_y. This degradation as observed in 3 is attributed to the model's lack of degrees of freedom, as it always assumes steady-state behavior, failing to account for lateral mass transfer. In contrast, our transfer learning method, which leverages knowledge from the bicycle model, shows improvement during the high-dynamic phases of the slalom trajectory, particularly in scenarios involving mass transfer, compared to purely learning-based and discrepancy modeling methods. The Table IV shows the relative performance gains in percentage with respect to the learning-based method. The relative gain in performance is computed as follows for each state variable: $(1 - \frac{RMSE}{RMSE_{\text{baseline}}}) \times 100$, where $RMSE_{\text{baseline}}$ corresponds to the smallest obtained RMSE of the learning-based method in Table III in bold. The physics-based model shows significant performance loss with respect to the baseline due to previously exposed limitations. Although its predictive accuracy is far outside the range of the learning-based method, the integration of this imperfect knowledge offers a way to significantly improve the performance of the transfer-learning method. The gain in performance for transfer-learning is dependent on the quality of the predictive performance of the physics-based method. For the state variable v_y, improvement is less significant than for $\dot{\psi}$ due to the strong bias of the bicycle model on the lateral speed. The discrepancy method shows a global decrease in performance for each state variable with respect to the baseline, related to the difficulty of making this method stable in an auto-regressive training context. Implementation of our method and data are available at https://github.com/N9TT-9G0A-B7FQ-RANC/IEEE_RAL

VI. Limitations

Although the integration of prior knowledge through physical models and their associated parameter beliefs in a transfer-learning context allows for improved predictive accuracy over a long-term prediction horizon, we did not assess the sensitivity of this improvement when using more complex physical models with better predictive capabilities, which can have impact on final performances. On the other side, we restricted the scope of our experimentation to MLP models that are designed to take only the current state and control as input, assuming a memoryless property of the underlying process. Recurrent neural network (RNN)

		Learning-based		Discrepancy		Transfer-learning	
Integration Time	Complexity	Vy	$\dot{\psi}$	Vy	$\dot{\psi}$	Vy	$\dot{\psi}$
0.05	8	0.07390 ± 0.00700	0.00825 ± 0.00035	**0.07115** ± 0.00667	**0.00782** ± 0.00043	0.07821 ± 0.02219	0.01074 ± 0.00678
	16	0.20725 ± 0.23938	0.03773 ± 0.06155	**0.07510** ± 0.02942	**0.01006** ± 0.00853	0.27206 ± 0.34704	0.02854 ± 0.03610
	24	0.15261 ± 0.22186	0.02597 ± 0.04075	**0.07917** ± 0.03554	**0.00976** ± 0.00576	0.10662 ± 0.06833	0.01417 ± 0.01019
0.5	8	0.05778 ± 0.00293	0.00682 ± 0.00028	0.06174 ± 0.00478	0.00667 ± 0.00027	**0.05451** ± 0.00235	**0.00634** ± 0.00032
	16	0.05424 ± 0.00105	0.00626 ± 0.00025	0.05978 ± 0.00400	0.00643 ± 0.00022	**0.05265** ± 0.00085	**0.00578** ± 0.00019
	24	0.05358 ± 0.00165	0.00609 ± 0.00031	0.05504 ± 0.00175	0.00641 ± 0.00036	**0.05142** ± 0.00148	**0.00589** ± 0.00030
1.0	8	0.05361 ± 0.00127	0.00669 ± 0.00028	0.05378 ± 0.00167	0.00663 ± 0.00013	**0.05252** ± 0.00204	**0.00637** ± 0.00035
	16	0.05109 ± 0.00082	0.00630 ± 0.00020	0.05483 ± 0.00133	0.00657 ± 0.00017	**0.05063** ± 0.00084	**0.00566** ± 0.00012
	24	0.05166 ± 0.00110	0.00618 ± 0.00023	0.05439 ± 0.00238	0.00638 ± 0.00025	**0.05004** ± 0.00094	**0.00569** ± 0.00016

TABLE III: RMSE computed over 10-second trajectories from the test set, for various training integration times (in seconds) and different MLP complexities (corresponding to the number of neurons per layer). RMSE are provided for v_y and $\dot{\psi}$ state variables for each compared method. Uncertainties are provided as ± one standard deviation.

	v_y	$\dot{\psi}$
Transfer-learning	+2.00	+7.06
Discrepancy	-5.26	-4.76
Physic-based	-131.19	-36.28

TABLE IV: Relative performance gains in percentage with respect to the learning-based method

architectures [25] have demonstrated wide range of success in time series forecasting and could present an additional layer of performance improvement due to the exploitation of more powerful inductive biases in conjunction with the use of prior knowledge through the same transfer-learning procedure described in this paper. In this context, RNNs, according to [26], require a careful memory state initialization procedure to guarantee stable long-term prediction, which is outside the scope of the issues addressed in this paper, but offers future research opportunities.

VII. Conclusion and Future Work

Accurate long-term simulation of dynamical systems is critical for studying, optimizing, controlling, and monitoring of process. In this work, we developed a robust methodology to address this challenge by leveraging both system measurements and prior knowledge about the system's physical behavior. Our method is based on transfer learning from a pretrained MLP on simulation data generated by a known physical model to a source dataset corresponding to real-world measurements of state variable evolution. The physical model includes a set of parameters estimated using Bayesian inference techniques that leverage additional prior knowledge about parameter distributions. The fine-tuning process involves an autoregressive function that improves the prediction accuracy of the final model by optimizing the MLP over many numerical integration steps, enhancing the robustness of the training against noise and disturbances in the measurements. Our experimental results on real-world measurements related to lateral vehicle dynamics demonstrate that our proposed method significantly outperforms purely data-driven models and traditional discrepancy modeling frameworks.

Acknowledgment

The authors would like to express their gratitude for the financial support provided by the French Ministry of Defense through the Defense Innovation Agency (AID) and by the National Institute for Research in Digital Science and Technology (INRIA)

References

[1] W. Hu, T. Zhang, X. Deng, Z. Liu, and J. Tan, "Digital twin: A state-of-the-art review of its enabling technologies, applications and challenges," *Journal of Intelligent Manufacturing and Special Equipment*, vol. 2, no. 1, pp. 1–34, 2021.

[2] M. Jiménez-Guarneros, P. Gómez-Gil, R. Fonseca-Delgado, M. Ramírez-Cortés, and V. Alarcón-Aquino, *Long-Term Prediction of a Sine Function Using a LSTM Neural Network*. Cham: Springer International Publishing, 2017, pp. 159–173.

[3] V. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.

[4] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, pp. 422–440, 2021.

[5] L. Ljung, *System Identification: Theory for the User*, ser. Prentice Hall information and system sciences series. Prentice Hall PTR, 1999.

[6] W. J. H. Stortelder, *Parameter estimation in nonlinear dynamic systems*, ser. CWI tract ; 124. Amsterdam, Netherlands: Centrum voor Wiskunde en Informatica, 1998.

[7] E. L. Ionides, C. Bretó, and A. A. King, "Inference for nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 103, no. 49, pp. 18 438–18 443, 2006.

[8] J.-M. Marin, P. Pudlo, C. P. Robert, and R. J. Ryder, "Approximate bayesian computational methods," *Statistics and Computing*, vol. 22, no. 6, pp. 1167–1180, 2012.

[9] F. Lionti, N. Gutowski, S. Aubin, and P. Martinet, "Bayesian approach fornbsp;parameter estimation innbsp;vehicle lateral dynamics," in *Foundations of Intelligent Systems: 27th International Symposium, ISMIS 2024, Poitiers, France, June 17–19, 2024, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2024, p. 249–259.

[10] J. Akeret, A. Refregier, A. Amara, S. Seehars, and C. Hasner, "Approximate bayesian computation for forward modeling in cosmology," *Journal of Cosmology and Astroparticle Physics*, vol. 2015, no. 08, p. 043, aug 2015.

[11] A. Pinkus, "Approximation theory of the mlp model in neural networks," *Acta Numerica*, vol. 8, p. 143–195, 1999.

[12] Z. Ghahramani and S. Roweis, "Learning nonlinear dynamical systems using an em algorithm," in *Advances in Neural Information Processing Systems*, M. Kearns, S. Solla, and D. Cohn, Eds., vol. 11. MIT Press, 1998.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '16. IEEE, Jun. 2016, pp. 770–778.

[14] W. Weinan, "A proposal on machine learning via dynamical systems," *Communications in Mathematics and Statistics*, vol. 5, no. 1, pp. 1–11, Mar. 2017.

[15] J. Brynjarsdóttir and A. OHagan, "Learning about physical parameters: the importance of model discrepancy," *Inverse Problems*, vol. 30, no. 11, p. 114007, oct 2014.

[16] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[17] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho, "Lagrangian neural networks," 2020.

[18] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 1, p. 9, May 2016.

[19] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 4243–4250.

[20] H. Pacejka, *Tyre and Vehicle Dynamics*, ser. Automotive engineering. Butterworth-Heinemann, 2006.

[21] E. Bakker *et al.*, "A new tire model with an application in vehicle dynamics studies," *SAE Transactions*, vol. 98, pp. 101–113, 1989.

[22] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. P. H. Stumpf, "Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems," *Journal of the Royal Society Interface*, vol. 6, no. 31, pp. 187–202, February 2009.

[23] Z. Chen and D. Xiu, "On generalized residual network for deep learning of unknown dynamical systems," *Journal of Computational Physics*, vol. 438, p. 110362, 2021.

[24] D. D. Maclnnis *et al.*, "A comparison of moment of inertia estimation techniques for vehicle dynamics simulation," *SAE Transactions*, vol. 106, pp. 1557–1575, 1997.

[25] H. Salehinejad, J. Baarbe, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," *CoRR*, vol. abs/1801.01078, 2018.

[26] 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.