Pontifícia Universidade Católica
do Rio de Janeiro

Lucas Castro Sousa

Nonlinear identification and predictive control of vehicle dynamics

Tese de Doutorado

Thesis presented to the Programa de Pós–graduação em Engenharia Mecânica, do Departamento de Engenharia Mecânica da PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Mecânica.

Advisor: Prof. Helon Vicente Hultmann Ayala

Rio de Janeiro
February 2023

PUC-Rio - Certificação Digital Nº 1912774/CA

Pontifícia Universidade Católica
do Rio de Janeiro

Lucas Castro Sousa

Nonlinear identification and predictive control of vehicle dynamics

Thesis presented to the Programa de Pós–graduação em Engenharia Mecânica da PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Mecânica. Approved by the Examination Committee:

Prof. Helon Vicente Hultmann Ayala
Advisor
Departamento de Engenharia Mecânica – PUC-Rio

Prof. Marco Antonio Meggiolaro
Departamento de Engenharia Mecânica – PUC-Rio

Prof. Ricardo Teixeira da Costa Neto
IME

Prof. Elias Dias Rossi Lopes
IME

Prof. Roberto Zanetti Freire
PUCPR

Prof. Mauro Speranza Neto
Pesquisador Autônomo

Rio de Janeiro, February the 14th, 2023

PUC-Rio - Certificação Digital Nº 1912774/CA

All rights reserved.

Lucas Castro Sousa

Holds a bachelor's degree in Mechanical Engineering (2016) from the Amazonas State University (UEA), and a master's degree in Mechanical Engineering (2018) from the Military Institute of Engineering (IME). His research interests include vehicle dynamics, system identification, machine learning, and predictive control techniques.

PUC-Rio - Certificação Digital Nº 1912774/CA

Bibliographic data

Sousa, Lucas Castro

Nonlinear identification and predictive control of vehicle dynamics / Lucas Castro Sousa; advisor: Helon Vicente Hultmann Ayala. – 2023.

145 f: il. color. ; 30 cm

Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Mecânica, 2023.

Inclui bibliografia

1. Engenharia Mecânica – Teses. 2. Redes neurais artificiais. 3. Veículos autônomos. 4. Modelos híbridos. 5. Controle preditivo. 6. Identificação de sistemas. 7. Dinâmica de veículos. I. Ayala, Helon Vicente Hultmann. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Mecânica. III. Título.

CDD: 621

PUC-Rio - Certificação Digital Nº 1912774/CA

To my family, for their support
and encouragement.

Acknowledgments

First, I would like to thank my family for their support during this difficult journey, always encouraging me and keeping me focused on making this dream come true. A very special thanks to my mother, the person I admire most, who has always encouraged me to improve myself and pursue higher education.

To my advisor, prof. Helon Vicente Hultmann Ayala, for the guidance and continuous support on the development of this thesis. My sincere thanks also go to the people who encouraged and supported me through the development of this research.

To the professors of the Mechanical Engineering Department of PUC-Rio, for the quality of teaching, which was essential for developing this work.

Finally, to the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ) and the Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) for the scholarship and financial support to this research. This study was partly supported by the CNPQ under grant 141360/2019-4, and the FAPERJ under grants E-26/201.358/2022 (272597) and E-26/210.314/2019 (249246). In addition, this study was also financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

PUC-Rio - Certificação Digital Nº 1912774/CA

Abstract

Sousa, Lucas Castro; Ayala, Helon Vicente Hultmann (Advisor).
Nonlinear identification and predictive control of vehicle dynamics. Rio de Janeiro, 2023. 145p. Tese de Doutorado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Automated vehicles must travel in a given environment detecting, planning, and following a safe path. In order to be safer than humans, they must be able to perform these tasks as well or better than human drivers under different critical conditions. An essential part of the study of automated vehicles is the development of representative models that are accurate and computationally efficient. Thus, to cope with these problems, the present work applies artificial neural networks and system identification methods to perform vehicle modeling and trajectory tracking control. First, neural architectures are used to capture tire characteristics present in the interaction between lateral and longitudinal vehicle dynamics, reducing computational costs for predictive controllers. Secondly, a combination of black-box models is used to improve predictive control. Then, a hybrid approach combines physics-based and data-driven models with black-box modeling of the discrepancies. This approach is chosen to improve the accuracy of vehicle modeling by proposing a discrepancy model to capture mismatches between vehicle models and measured data. Results are shown when the proposed methods are applied to systems with simulated/real data and compared with approaches found in the literature, showing an increase of accuracy (up to 40%) in terms of error-based metrics while having lesser computational effort (reduction by up to 88%) than conventional predictive controllers.

Keywords

Artificial neural networks; Autonomous vehicles; Hybrid models; Predictive control; System identification; Vehicle dynamics.

PUC-Rio - Certificação Digital Nº 1912774/CA

Resumo

Sousa, Lucas Castro; Ayala, Helon Vicente Hultmann. **Identificação não-linear e controle preditivo da dinâmica do veículo**. Rio de Janeiro, 2023. 145p. Tese de Doutorado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Os veículos automatizados devem trafegar em determinado ambiente detectando, planejando e seguindo uma trajetória segura. De modo a se mostrarem mais seguros que seres humanos, eles devem ser capazes de executar essas tarefas tão bem ou melhor do que motoristas humanos sob diferentes condições críticas. Uma parte essencial no estudo de veículos automatizados o desenvolvimento de modelos representativos que sejam precisos e computacionalmente eficientes. Assim, para lidar com esses problemas, o presente trabalho aplica métodos de inteligência computational e identificação de sistemas para realizar modelagem de veículos e controle de rastreamento de trajetória. Primeiro, arquiteturas neurais são usadas para capturar as características do pneu na interação entre a dinâmica lateral e longitudinal do veículo, reduzindo o custo computacional em controladores preditivos. Em segundo lugar, uma combinação de modelos caixa-preta é usada para melhorar o controle preditivo. Em seguida, uma abordagem híbrida combina modelos baseados na física e orientados por dados com modelagem de caixa-preta das discrepâncias. Essa abordagem é escolhida para melhorar a precisão da modelagem de veículos, propondo um modelo de discrepância para capturar incompatibilidades entre modelos de veículos e dados medidos. Os resultados são mostrados quando os métodos propostos são aplicados a sistemas com dados simulados/reais e comparados com abordagens encontradas na literatura, mostrando um aumento de precisão (até 40%) em termos de métricas baseadas em erro, com menor esforço computacional (redução de até 88%) do que os controladores preditivos convencionais.

Palavras-chave

Redes neurais artificiais; Veículos autônomos; Modelos híbridos; Controle preditivo; Identificação de sistemas; Dinâmica de veículos.

PUC-Rio - Certificação Digital Nº 1912774/CA

Table of contents

PUC-Rio - Certificação Digital Nº 1912774/CA

PUC-Rio - Certificação Digital Nº 1912774/CA

PUC-Rio - Certificação Digital Nº 1912774/CA

List of figures

PUC-Rio - Certificação Digital Nº 1912774/CA

PUC-Rio - Certificação Digital Nº 1912774/CA

PUC-Rio - Certificação Digital Nº 1912774/CA

List of tables

PUC-Rio - Certificação Digital Nº 1912774/CA

PUC-Rio - Certificação Digital Nº 1912774/CA

List of Abreviations

AGV – Autonomous ground vehicle
ANN – Artificial Neural Network
ARX – AutoRegressive with eXogenous inputs
CPU – Central Processing Unit
CG – Center of Gravity
DoF – Degree of Freedom
ELU – Exponential Linear Unit
EMPS – Electro-Mechanical Positioning System
FR – Free-Run simulation
IPM – Interior-Point Method
LMPC – Linear Model Predictive Control
LPV – Linear Parameter-Varying
LTV – Linear Time-Varying
MPC – Model Predictive Control
ML – Machine Learning
MLP – Multilayer Perceptron
NARX – Nonlinear AutoRegressive with eXogenous inputs
NLP – Nonlinear Programming
NMPC – Nonlinear Model Predictive Control
NN – Neural Network
ODE – Ordinary Differential Equation
OSA – One-Step-Ahead
PD – Proportional-Derivative
PEM – Prediction Error Method
PI – Proportional-Integral
QP – Quadratic Programming
RAM – Random Access Memory
RBF – Radial Basis Function
RBFNN – Radial Basis Functions Neural Network
RMSE – Root Mean Squared Error
SQP – Sequential Quadratic Programming
SYSID – System Identification
VAF – Variance-Accounted-For metric

PUC-Rio - Certificação Digital Nº 1912774/CA

Part I

General Introduction

PUC-Rio - Certificação Digital Nº 1912774/CA

1
Contextualization and goals

PUC-Rio - Certificação Digital Nº 1912774/CA

Autonomous ground vehicles (AGVs) are likely to revolutionize the transportation system since they reduce pollution and mobility costs [1, 2]. Moreover, increase efficiency and safety by reducing human interference responsible for 94% of vehicle crashes [3]. Motivated by this, autonomous vehicles must operate under different and variable circumstances safer than human drivers. Though some autonomous commands (e.g., autopilot in airplanes and ships) have been used since the 1920s, their application in ground vehicles is considered a recent problem since autonomous commands, applied in AGVs, were first identified in the 1980s and gaining new importance due to the Defense Advanced Research Projects Agency Grand Challenge [4]. In recent years, AGVs have been a research trend in the automotive field. Several automotive companies and technical institutes have been producing technological advances in developing intelligent autonomous vehicles [5].

As with many technological revolutions, the vehicle's capabilities will arrive in stages. Thus, conveyances will ascend a ladder of automated functions through the next years leading to an autonomous vehicle that can drive itself. The steps given to the progress in the implementation of AGVs may lead users, in general, to need clarification regarding the gradations of autonomy. Therefore, one of the first procedures along those lines was the six different levels outlined by SAE-J3016 [6] (Fig. 1.1), in which, for an automated vehicle, the levels of autonomy can be defined:

– Level 0: The human operator is responsible for controlling the vehicle. The presence of automated systems may indicate warnings;

– Level 1: Adaptive Cruise Control, Parking and lane keep assistance, and other automated assistance may be available;

– Level 2: The vehicle system executes important functions such as acceleration, braking, and steering. In addition, the automated system can be deactivated upon takeover by the driver;

– Level 3: The automated vehicle is able to perform auto-pilot within specific conditions, such as freeways. When the automated features request, the human operator must drive;

- Level 4: The system can control the vehicle in any direction except under severe weather. Moreover, the human operator must enable the automated system when it is safe;
- Level 5: The system allows the vehicle full autonomy under any condition. In addition, the driver's attention and intervention are not required anymore;

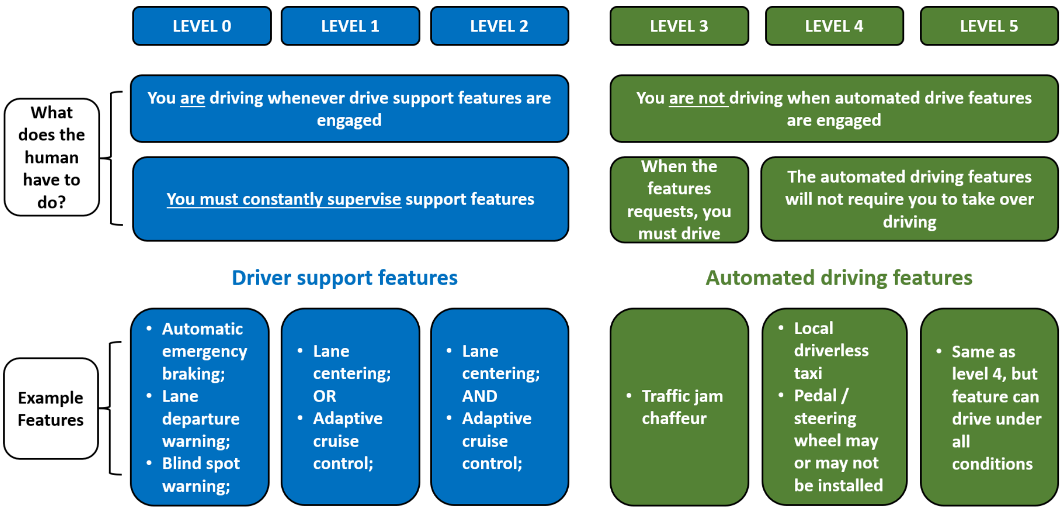

Figure 1.1: Levels of driving automation according to SAE-J3016.

In general, the architecture for AGVs is typically composed of three basic modules or layers: sensing and perception, robust route planning, and a trajectory tracking [7, 8] (Fig. 1.2). The sensing and perception layer may provide updated data for the system to identify the actual time, location, and environment around the vehicle. Thus, output data is prepared for the system to process. The planning layer uses the data produced by the sensing and perception layer to indicate a feasible and safe trajectory to be followed. Then, the trajectory tracking layer contains the control strategies (including actuator control of each subsystem) to allow the vehicle to follow a reference path.

Focusing on the latter, trajectory tracking control is an essential part of the study of AGVs. It aims to keep the vehicle on a pre-established trajectory, reduce accidents, and improve the traffic flow [9]. However, path-tracking is arduous for ground vehicles since these vehicles may be subject to high slip conditions due to curvy trajectories, obstacles, and environmental conditions, which may directly affect the performance of this type of control [10]. Thus, many works have dealt with the problem of AGV trajectory tracking control [4, 7]. Geometric and kinematic controllers, for example, are popular strategies due to their simplicity and stability and inspired several configurations found in

PUC-Rio - Certificação Digital Nº 1912774/CA

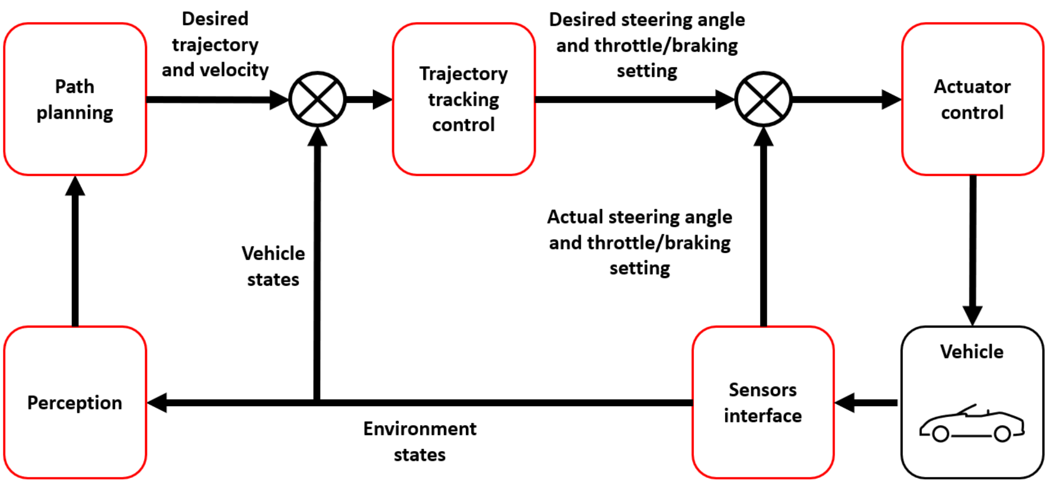

Figure 1.2: Overview regarding the autonomous vehicle system.

the literature. Among them, follow the carrot [11], its extension, the well-known Pure pursuit [12–15], as well as the Stanley method [13, 16, 17]. These latter methods have been a standard benchmark to validate new proposed controllers [7]. In addition, these controllers are useful for applications mainly at low speed, becoming popular due to reduced computational cost [18]. However, these controllers usually do not consider dynamic aspects. They, therefore, may be unsuitable for particular circumstances under urban or racing tracks, mainly when uncertainties and limits of handling are easily achieved.

The vehicle-environment interaction can be better understood by using Vehicle Dynamics. According to [19], vehicle dynamics can be divided into longitudinal, vertical, and lateral. Longitudinal dynamics act to verify the behavior of the vehicle when subjected to acceleration and braking forces. The vertical dynamics deals with the ride, focusing on the comfort of vehicle occupants during vibrations at low frequencies (generally between 0 and 25 Hz) when the vehicle is subjected to excitation from the base [20, 21]. Lateral dynamics, in turn, deals with the stability and maneuverability of the vehicle. However, this work addresses only studies regarding lateral and longitudinal dynamics. With this view, dynamic-based controllers are a potential solution for accurate path-tracking control. Common approaches include Sliding Model control [22, 23], H∞ control [24, 25], Game-based control [26, 27], and Model-based controllers [9, 28, 29].

Model-Based Predictive Control (MPC) has become an attractive control method since it allows for handling system constraints and future forecasts [30]. This method applies optimal control actions to the system within a limited horizon using optimization mechanisms [31]. The MPC approach is

PUC-Rio - Certificação Digital Nº 1912774/CA

generally used when linear design methods fail to ensure adherence to the project requirements. The advantages of MPC control include its ability to handle different variables and constraints [30] and provide a nonlinear control law concerning path-following errors, even when applied to linear systems [28]. These characteristics lead to performance gains in closed-loop system operation. In addition, the imposition of constraints on the vehicle states can ensure comfort and safety for the passengers [32]. However, a setback for the MPC approach is its dependence on an accurate vehicle model [4, 33]. Although a complex and reliable dynamic vehicle model can guarantee satisfactory path-tracking in terms of accuracy, the computational effort required by the MPC solver may not be suitable for real-time implementation.

The parameters commonly used to derive linear and nonlinear dynamic models are difficult to determine with precision and to tune correctly, which may lead to uncertain results [34], specifically for autonomous vehicles that require vehicle models for control design. The traditional dynamic modeling needs insight into propulsion and nonlinear tire effects besides the vehicle behavior, which are relatively complicated to control, implement in real-time, and analyze in terms of stability [35]. For ground vehicles, one of the main difficulties is the modeling of tire-surface interactions considering both road and off-road vehicles, where, for the latter, deformable soils are composed of different unstable particles and commonly result in a considerable deviation between predicted and measured data [36].

In this way, Computational Intelligence (CI) and System Identification (SYSID) data-driven models emerge as a potential solution since they may represent complex systems from measured data, providing a suitable trade-off between uncertainties throughout the system and accuracy [18, 37]. System Identification is the science of developing mathematical models of physical systems using experimental data [38]. In the control field, system identification is focused on modeling physical processes that describe the behavior of a given system, being an effective tool in the advanced control strategies [39]. The system identification models may be classified as white, gray, and black-box models regarding the prior information about the system [40]. Moreover, identified models may also be classified as linear and nonlinear according to the adherence to the superposition condition [41]. The white-box modeling requires the full knowledge of the law governing the system, and the measured data is commonly used for model validation. On the other hand, black-box models are derived from a technique in which it is assumed that no prior information regarding the system is available [40]. Gray-box models stand between the previous models. Due to the complexity and uncertainties regarding tire-

PUC-Rio - Certificação Digital Nº 1912774/CA

road interaction modeling and other nonlinear aspects, white-box modeling is challenging for vehicle applications. Therefore black-box models become a potential solution for designers. Among them, models considered suitable for this approach include higher-degree polynomial functions, fuzzy systems, and artificial neural networks (ANNs) [42].

Typical CI and SYSID studies related to vehicle dynamics generally obtain longitudinal dynamics [37, 43–45], estimation of some lateral dynamic parameters such as the side slip angle[46, 47] (requiring a dynamic model), and the proper lateral dynamics [36] including driver steering model [48]. Other approaches include the tire study, which contributes most to ground vehicle dynamics [49]. Tires are the link between the vehicle and the environment, capable of transmitting forces that influence vehicle motion. Therefore, the knowledge of friction aspects on the contact patch is crucial for the vehicle safety systems and in improving noise emission and fuel economy [50]. Moreover, one of the essential topics of vehicle research has been the design of traction/braking control systems since the loss of adhesion between the tire and the surface leads to vehicle instability.

Previous studies have used different methodologies to approximate tire models as [51], which obtained different curve fittings to the traditional Pacejka tire model[1] using rational functions, expansions in a series of Chebyshev polynomials, and a series of rational orthogonal functions. In [52], the authors applied a multilayer feed-forward neural network to build an intelligent tire. The peak value of the tire-road friction curve to control the slip using linear methods is the main topic of the work presented by [53]. In [54], a neural network tire model is applied to predict both longitudinal and lateral forces on the tire by estimating the Pacejka model parameters. Reference [55] applied Artificial Neural Networks to predict Pacejka model parameters.

Mainly aiming at the path-tracking problem based on system identification, a few studies have been made regarding two-wheeled ground vehicles [56], and considering four-wheeled vehicles [3, 18]. Therefore, path-tracking based on system identification is undoubtedly a topic that needs exploration, as will be mentioned in the specialized literature review (Chapter 2). The system identification method is based on an iterative task considering experimental or simulated data. Fig. 1.3 shows an overview of the system identification procedure based on four stages. In Stage 1: Acquire data, the main objective is the data acquisition which can be obtained from simulated or experimental tests. This stage is crucial for the purpose of system identification since

[1]Mathematical representation widely used to simulate the longitudinal/lateral tire forces as a function of longitudinal/lateral slip. Further information in Section 3.1.2.

PUC-Rio - Certificação Digital Nº 1912774/CA

whether input signals are introduced in the system producing unexpected and non-informative output data, then the resultant model will fail to represent the system accurately. Therefore, poorly designed experiments/tests may lead to unsatisfactory and inadequate data acquisition, leading to badly identified models.

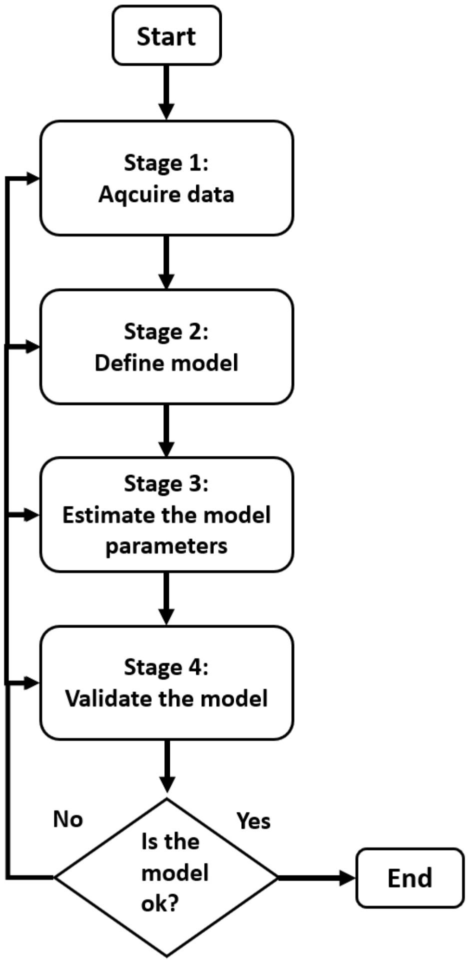

Figure 1.3: Overview regarding the system identification method, adapted from [57].

In Stage 2: Define model, one has to choose the model configuration, particularly linear and nonlinear models. When linear, the model configuration (model order) has to be defined. Then, the prediction approaches may be applied to derive and build the identified linear models. A significant advantage of using linear models is their simple application together with control approaches since they are easier to interpret and make direct relations with real physical systems. On the other hand, under certain conditions in which nonlinear aspects dominate the system, the accuracy of linear models fails, and nonlinear models are required. In addition, if the linear model can perform the prediction with accuracy based on validation metrics, there is no reason to

PUC-Rio - Certificação Digital Nº 1912774/CA

choose a nonlinear model. Suppose one opts to choose a nonlinear model. In that case, structure and orders should be set using a family of models with structure and model order variations, commonly by trial-and-error tasks.

Next, in Stage 3: Estimate the model parameters, once the model configuration, including structure and order, becomes possible, the estimation of the model parameters by using, generally, an optimization problem. Considering ANNs, this procedure is commonly referenced as learning or training steps. The optimization task relies on the minimization of the One-Step-Ahead (OSA) or Free-run prediction residuals (detailed in Section 4.1.1).

Finally, in Stage 4: Validate the model, the designer has to test the identified model and check it based on validation metrics and effectiveness through the desired application. One of the most common ways to check the model validation is using error-based metrics in which the data residuals from the prediction and the measured data are compared (detailed in Section 4.1.1). Once the validation is set, the designer can decide whether the model is valid. If the model is unsatisfactory, the designer should go backward in the methodology and re-apply some of the abovementioned stages.

As mentioned above, due to the nonlinear aspects of vehicle dynamics, modeling automotive subsystems remains a challenging task. On the one hand, physics-based models can provide accurate and feasible results when the designer knows the vehicle's properties but may require enormous computational effort. On the other hand, whether measured data is available, data-driven approaches can be used to derive identified models that represent the acquisition data. However, these procedures may face significant challenges, such as noise in the data and the inability to include and satisfy physical constraints [58, 59]. Thus, to cope with these issues, the present work also deals with a hybrid approach (Fig. 1.4) that combines vehicle modeling with a black-box modeling of the discrepancies. This approach is able to increase the accuracy and feasibility of vehicle modeling since the discrepancy model can capture mismatches between vehicle models and measured data. First, the vehicle model outputs are compared to the acquisition data to derive the discrepancy data. Then, the black-box approach is used to model the error or mismatches. Finally, the discrepancy data is summed to the vehicle's outputs to derive the hybrid approach's output. Hence, the new discrepancy data is expected to be less than the original discrepancy data.

PUC-Rio - Certificação Digital Nº 1912774/CA

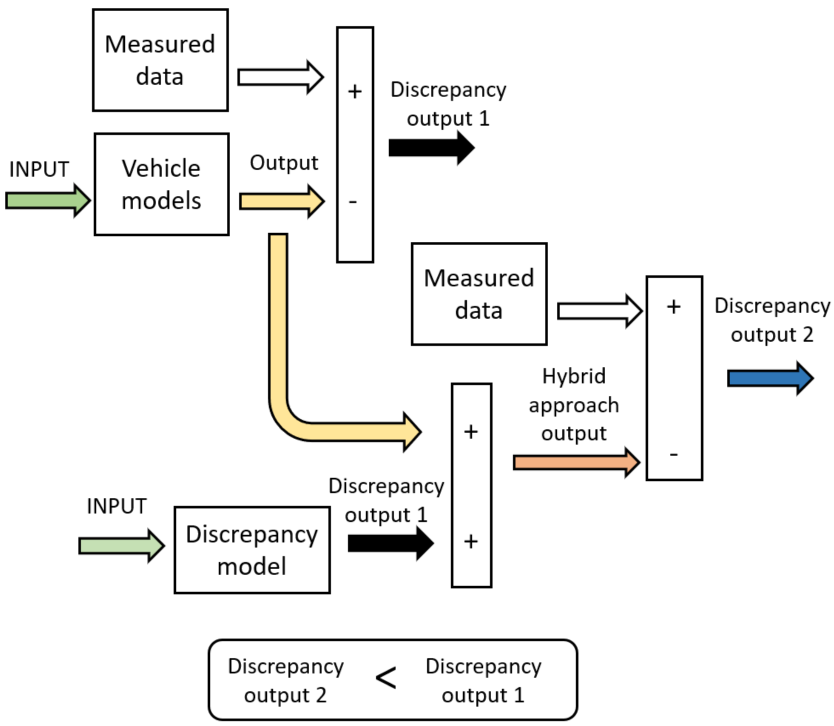

Figure 1.4: General regarding the hybrid approach: The vehicle models produce discrepancy data to be modeled by the discrepancy model. Next, the hybrid approach comprises the vehicle models' outputs summed with their respective identified discrepancy models.

PUC-Rio - Certificação Digital Nº 1912774/CA

1.1 Motivation

As highlighted before, the modeling, control, and simulation of AGVs face difficulties for implementation in real-time in the presence of nonlinearities. Thus, methods aiming to enhance the effectiveness of predictive modeling by means of accurate and statistically valid models are of interest to the vehicle dynamics community.

The present work deals with applying the Computational Intelligence methods and System Identification as accurate models for predicting tire modeling and longitudinal and lateral dynamics behaviors, leading to effectively performing predictive control.

These procedures also aim to produce less research effort since previous minor information regarding the nonlinearities and other vehicle properties is needed. These could lead researchers to spend more time analyzing the results than with tedious and financially costly procedures to obtain terrain and vehicle properties. Besides, the present work allows the development of future work related to different CI and SYSID algorithms and the implementation of different control approaches for the control of ground vehicles.

1.2 Objectives

The main goal of the present thesis is to apply CI and SYSID techniques in vehicle dynamics to allow the application of predictive control for path-tracking effectively.

According to the aforementioned main goal of this work, it is possible to enumerate the specific goals as follows.

- To approximate the tire interaction by means of Machine Learning, in particular, Radial basis function (RBF) and multilayer perceptron (MLP) neural networks;
- To compare the learned model with Pacejka's tire model and experimental data for efficient model predictive control;
- To identify lateral dynamics of a ground vehicle using AutoRegressive with eXogenous inputs (ARX) models;
- To elaborate on a novel Multi-ARX-MPC (MARX-MPC) control framework for adopting a unique cost function that considers multiple vehicle data-driven models;
- To compare the MARX-MPC with conventional model predictive controllers, in particular, linear (LMPC) and nonlinear MPC (NMPC);
- To obtain a more accurate vehicle model using a hybrid approach of vehicle models (data-driven and physics-based models) and discrepancies modeled by black-box models;
- To compare the proposed hybrid approaches with the conventional vehicle models presented in the literature and establish their importance;

PUC-Rio - Certificação Digital Nº 1912774/CA

2
Literature review and contributions

The aim of the present chapter is to present the literature review regarding MPC and vehicle models, together with the original contributions and the outline of the present thesis. The first part deals with the literature review of MPC controllers applied to trajectory tracking control which will be the focus of two of the proposed contributions. Secondly, a literature review regarding data-driven and physics-based vehicle models, which will be part of all the original contributions, is established. Next, the original contributions are stated and described. Lastly, the outline of the present thesis is presented.

2.1
Model predictive control for trajectory tracking

In the context of AGVs, the MPC methodology has been used for trajectory-tracking tasks since it deals with representative system models, constraints, and future predictions based on that model [9, 28]. Moreover, the MPC control proved effective for robust control with a reasonable computational cost. Furthermore, passenger comfort and vehicle handling can be improved by implementing an MPC control due to the possibility of using multiple states and constraints to ensure stability and security [32]. Another factor contributing to the widespread use of MPC control is its robustness against system uncertainties.

The MPC for trajectory tracking control has been generally applied by using two different approaches, i) NMPC, designed using a nonlinear vehicle dynamics model, tends to enhance path-tracking, capturing nonlinear vehicle characteristics, and ii) LMPC, generally designed by applying simplified vehicle dynamics through linearization processes at specific operating points, using the linear approximation of the model [33]. Considering the NMPC approach, the application and formulation of a Nonlinear Programming (NLP) problem is expected [60, 61]. Then, to solve this problem, techniques such as the Sequential Quadratic Programming (SQP) [62, 63] and the Interior-Point Method (IPM) [64] were developed. However, due to the complexity commonly involved during the NLP process, linearization techniques have been proposed, enabling Quadratic Programming (QP). Thus, the system

PUC-Rio - Certificação Digital Nº 1912774/CA

(kinematic or dynamic model) is changed to a Linear Time-Varying (LTV) system [60, 65]. It is important to point out that the tire aspect becomes essential when linearizing the vehicle's dynamics. When using linear tire modeling, the tire force approximation turns the slip angles limited to large slip angles [60]. In addition, constraints are commonly used to keep slip angles in the linear envelope [66, 67]. Based on the principle of MPC controllers, a linear time-varying model predictive controller (LTV-MPC) considering cornering characteristics is designed and optimized in [68]. Ignoring the effects of vertical, roll, and pitch motions, the authors established a three-degree-freedom model of vehicle monorail, considering the Pacejka Formula. The results show that the controller has good self-adaptability under rough terrains. In [69], a linear time-varying model predictive controller (LTV-MPC) is designed through local linearization, tracking the desired vehicle velocity and the reference path.

Even though the LTV procedure can reduce the computational effort, it requires performing the successive linearization process online. Then, to further reduce the computational burden in the process, a Linear Parameter Varying (LPV) approach can be derived [70]. Through this process, the model's parameters vary for the different polyhedral regions of the state-space by using a scheduling variable [33]. In contrast to the LTV approach, the LPV method does not need successive online linearization. Therefore it can reduce computational burden and efficiently improve MPC.

At low speeds, applying real-time MPC becomes possible with accuracy due to the use of simplified models. However, with the increase in speed, the simplified model becomes uncertain and complex in the implementation in real-time [28]. Thus, more detailed models and the use of punctual simplifications have been proposed to enhance the MPC in terms of accuracy and computational efficiency. In [28], the authors used a Switched MPC in which vehicle models (kinematics, linear and nonlinear dynamics) are switched during the prediction according to error-based metrics and computational time. Reference [71] has considered an MPC control with 2 degrees of freedom (DoF) vehicle model with Pacejka tire modeling. The authors propose a steering angle envelope in the MPC function to improve lateral stability. Reference [72] has proposed a different approach to trajectory control by using a time-varying and non-uniformly spaced control horizon. The approach reduces the time interval for the near future and increases the time interval to extend the prediction horizon. Thus, the approach maintains the trajectory tracking performance and reduces the computational effort. In [73], the authors develop an NMPC controller with a single-track prediction model and Pacejka tire modeling to control the lateral position and yaw. The novelty is the imple-

PUC-Rio - Certificação Digital Nº 1912774/CA

mentation of an optimizer called continuation/generalized minimal residual. Moreover, different works have used dynamic and kinematic models with slip assumptions to design an MPC controller. In [74, 75], the authors used a kinematic model with slip angles to develop an MPC controller, which presented satisfactory results even though the slip estimation was considered a problem because of the absence of a more robust method in their modeling. Reference [14] presents an MPC controller architecture considering both the kinematic and the dynamic control in a cascade structure to perform path-tracking of an autonomous Baja vehicle. The results were considered satisfactory after practical experiments to validate the simulation data. Reference [76] applied an MPC-based path-tracking control in a ground vehicle over three types of roads: wet and dry asphalt pavement and ice-covered soil. In [77], the authors proposed a nonlinear model predictive controller for path-tracking of a ground vehicle considering the Pacejka Formula. Under some situations, high precision of path-tracking could result in loss of lateral stability. Thus, a direct yaw control with an MPC controller based on linear matrix inequality is proposed by [78], which achieves satisfactory results for a real vehicle. A trajectory tracking method with a time-varying model based on a simple dynamic model which considers both the linear and nonlinear MPC algorithms is proposed by [79]. The designed control architecture proposed by [80] is composed of an MPC controller based on a kinematic model, internal feedback control of yaw rate, and sideslip compensation to enhance path-tracking.

Machine Learning techniques can also be used to predict the future states of AGV during path-tracking control instead of using a physics-based model. In [18], an MPC controller is designed with learned vehicle dynamics employing experimental data. The metric results demonstrated that the proposed technique could successfully represent vehicle behavior and be suitable for real-time operation. In [81], the authors proposed a data-driven identification using neural networks to learn vehicle operation data to implement MPC control of a racing car. Reference [82] developed a data-driven identification of an AGV based on an LPV framework using machine learning techniques.

2.2 Physics-based and data-driven vehicle models

An essential aspect of studying autonomous vehicles is the development of representative mathematical models of vehicle dynamics. Vehicle modeling is an essential task in the design process since the final product can be built faster, with a satisfactory level of performance, while reducing financial costs. Moreover, control laws, in particular for path-tracking, are usually derived from

PUC-Rio - Certificação Digital Nº 1912774/CA

vehicle models [7]. Being so, several works have been developed considering three usual configurations consisting of a geometric, kinematic, and dynamic model.

Geometric models only consider the dimension and position of the vehicle during the motion and are commonly based on Ackermann steering [7, 83]. Different works have used geometric models [12, 13, 16, 84] as an initial step to develop some traditional and widely used geometric trajectory controllers, Pure Pursuit and Stanley controllers. However, geometric models are unsuitable for high-speed path-tracking due to their inability to include vehicle dynamics [4, 7].

On the other hand, kinematic vehicle models rely on the motion of the vehicle based on its position, velocity, and acceleration, usually lateral direction and yaw motion concerning fixed and global axes. Several works have extensively used these models due to their essential relationship to describe vehicle motion with simple equations. To mention a few, the authors [14] present a four-wheeled vehicle as a bicycle model where only one wheel per axle is considered. In [85], a steering controller based on a simple kinematic model was developed for autonomous vehicles in the Defense Advanced Research Projects Agency Grand Challenge. Other works have enhanced kinematic models by including both rear and front wheel slip angles to account for slippery terrains [74, 75, 80, 86].

Unlike geometric and kinematic models, dynamic models consider internal forces (e.g., the mass of the vehicle and tire-road interactions), momentum, or energy within the system [7]. Dynamic models are commonly related to the previous models. They derive from Newtonian equations of motion and may be commonly described as full-vehicle, or half-vehicle (bicycle model) models [83]. Moreover, tire forces derived from the interaction between tire and ground can be represented during maneuvering in both the lateral and longitudinal directions using different approaches, such as the Pacejka tire model applied in [7, 68, 77], and developed by [49], LuGre friction model designed by a collaboration between control groups in Lund and Grenoble [87], and the tire model based on Julien's Theory described by [21].

Models derived from physics combine Newton's second law and empirical observations under certain conditions, making them easy to understand. When nonlinear [88–90], these models maintain good interpretability but are complex and challenging to configure/adjust. Thus, when applied to control schemes, they become relatively complicated to implement in real-time. On the other hand, when linear [80, 91, 92], the models could be more simplified, taking into account only characteristics representative of the vehicle, such as tires, suspen-

PUC-Rio - Certificação Digital Nº 1912774/CA

sion, and powertrain. Thus, a predictive control based on a simplified model may perform well in situations where nonlinear aspects are not dominant. In addition, unmodeled dynamics introduce uncertainties, significantly reducing the performance of the MPC control.

The MPC considering a highly complex model, is not a good solution due to the computational cost of the optimization. We can see in [28] that although alternating the models reduces the computational time, situations in which dominant nonlinear control can increase solver time. In [71, 93], the authors did not use longitudinal dynamics modeling, excluding longitudinal tire slip effects and powertrain acceleration effects. Furthermore, in [72], the authors consider linear tire modeling and very low speeds. In summary, whether linear or nonlinear, models derived from physics suffer from different shortcomings, including prior knowledge of vehicle aspects and time-consuming efforts.

Alternatively, one potential solution is CI using machine learning (ML) and SYSID techniques, which ensures a given system can learn and adapt to different situations [41]. Although system identification and machine learning have been developing independently through the years, recently, a great effort has been made to establish a common ground for these approaches [38]. Moreover, no prior information about the system is required resulting in an essential advantage for model approximation [94]. In [45], longitudinal model identification and velocity control of an AGV vehicle are designed for low-speed applications. An adaptive ARX model as a function of the operating point is used to identify throttle level as the input signal and vehicle velocity as the output. The velocity control is designed using a PI (Proportional-Integral) controller. In [43], and [37], data-driven techniques are applied to derive longitudinal and the combination of lateral and longitudinal dynamics of a vehicle, respectively. In both works, linear system identification is compared to nonlinear physical modeling resulting in satisfactory results for normal driving conditions. References [48, 95] proposed identifying parameters related to the driver model incorporating human sensory dynamics. In [96], the authors proposed an identification approach for parameter identification and lateral vehicle dynamics state estimation based on a Linear Fractional Transform reformulation of the vehicle and tire models. The approach allowed the use of nonlinearities and proved to be suitable for real-time applications.

Machine learning methods are a potential solution since they have learning characteristics and adaptation to different complex problems with precision [3]. Reference [97] proposed a data-driven model based on deep neural networks to represent the longitudinal characteristics of a ground vehicle. The proposed approach predicts the distance and velocity of the

PUC-Rio - Certificação Digital Nº 1912774/CA

vehicle in real-time with accuracy. Longitudinal dynamics identification is also proposed by reference [44]. The authors compared four different structures of neural networks (NNs) (with and without pre-wired structure, non-recursive, and recursive), indicating the versatility of NNs. In order to improve racing performance and capture vehicle dynamics, references [98] and [3] use data-driven methods. The former applies an iterative learning control to improve lateral and longitudinal tracking over multiple laps. At the same time, a second algorithm alters the vehicle trajectory. The latter reference proposed a neural network architecture using past states and inputs from the physical model. The proposed method results achieved satisfactory performance on an experimental ground vehicle.

However, these approaches may face significant limitations and challenges, such as noise in the data, which may introduce overfitting. In addition, it is challenging to include aspects to satisfy physics constraints [58, 59]. Thus, to cope with these problems, hybrid approaches combining vehicle models with the modeling of the discrepancies can be proposed. This approach is chosen to improve the accuracy of vehicle modeling by proposing a discrepancy model to capture mismatches between vehicle models and measured data. Every model, to varying extents, has discrepancies, that is, the mismatches between measured real data and the predicted by the model [99]. Several approaches have been proposed to model discrepancies. In [99, 100], the authors proposed a discrepancy model based on the Bayesian hierarchical model in building structures. The Bayesian hierarchical approach is also applied to model discrepancy problems in [101]. In [58], the authors applied the learning discrepancy model to improve the physics-based model of a double pendulum on a cart. In [59], discrepancy models are used to improve dynamic models regarding the physics of falling objects. Reference [102] proposed a hybrid approach, including a physics-based model and gray-box for discrepancy models.

2.3 Original contributions

To the best of our knowledge, although researchers have proposed data-driven models to approximate vehicle dynamics, such approaches have not been applied so far, neither for tire modeling and multi-identified vehicle models for efficient predictive control nor hybrid approaches combining vehicle and discrepancy models.

The first contribution of this thesis (Chapter 6) relies on using ML architectures for tire learning to enhance model predictive control efficiently and contributes with the following:

PUC-Rio - Certificação Digital Nº 1912774/CA

PUC-Rio - Certificação Digital Nº 1912774/CA

- Artificial neural networks can be applied to approximate nonlinear tire models with arbitrary precision;

  RBF and MLP neural networks are designed to approximate Pacejka's tire model and experimental data. High computational efforts are commonly required to accurately predict tire curves from traditional tire models. Moreover, vehicle control depends on the tire-road interaction, demanding an accurate tire model. These issues can be relieved by using data-driven models. Therefore, in this thesis, data-driven neural tires are built using machine learning from simulated/measured data resulting in models with optimized architectures. It is worth noting that neural approaches to predict tire curves have been used before [52, 55, 103], but that techniques only consider lateral curves or parameter estimation of traditional tire modeling, unlike the approach adopted in this thesis;

- Neural tire models can be used effectively with MPC to provide nonlinear control laws;

  Predictive controllers based on AGVs with data-driven tire models (MPC-Neural) are designed to regulate virtual plants' torque and steering angle inputs considering simulated and experimental tire data. The present method considers a reduced dataset instead of numerous datasets present in system identification and data-driven control, which is a standard procedure used in the literature [18, 35, 37].

- MPC with neural tire model as prediction inference is computationally more efficient than traditional approaches.

  The predictive control results show that it is possible to improve computational time by 25% in some cases, which indicates that using a data-driven model motivates the application in real-time. The gain in computational efforts is relevant compared to recent references [18, 104] considering neural networks to predict control laws.

Next, the second contribution (Chapter 7) relies on the application of multi-ARX vehicle models to efficiently enhance model predictive control aiming:

- To demonstrate that simplified data-driven vehicle models can be used as reliable predictions for developing predictive control laws when no prior knowledge of the vehicle is available.
- To propose a novel Multi-ARX-MPC (MARX-MPC) control framework for adopting a unique cost function that considers multiple vehicle data-driven models. The controller's performance is evaluated based on

trajectory error and computational complexity compatible with recent works [18, 28, 72]).

- To demonstrate that data-driven models as MPC prediction perform better in computational terms compared to conventional approaches present in the literature due to low sensitivity to the action of lateral force (results point to a 63% reduction compared to LMPC and 88% to NMPC).
- To reduce computational effort by using data-driven models as MPC prediction, maintaining a satisfactory error for sudden trajectory change.

Lastly, the third contribution (Chapter 8) deals with the use of a hybrid approach that combines vehicle models approaches with black-box modeling of the discrepancies aiming:

- To obtain a more accurate vehicle model using a hybrid approach of vehicle models (data-driven and physics-based models) in which black-box models model discrepancies. The vehicle models are derived using experimental open data [105] collected across racing driving conditions, enabling reproduction by the interested reader. In this regard specifically, the proposed approach was able to improve other approaches by up to 28% in terms of RMSE reduction;
- To demonstrate that the proposed hybrid approach can accurately improve the vehicle modeling of racing vehicles in maneuvers at the limits of handling. Few existing works apply system identification to learn vehicle dynamics from sampled data [18, 37, 43, 44]. However, those studies only considered normal driving conditions with small steering angles due to gentle maneuvers. From the results, the proposed approach outperforms the state-of-the-art, which is relevant for model-based control applications and digital twins [106].
- To easily reformulated the proposed approach for different levels of model complexity. Therefore, one may select the model most suitable for each desired application;
- To modify the physics-based model to receive driver's commands directly instead of force variables (traction and braking forces). This procedure allows the hybrid approach to be used directly by higher-level control applications;
- To add a metric of discrepancy model contribution through simulation outputs. Thus, it is possible to quantify the black-box submodel enhancements. This effect is essential for decision-making, indicating which envelopes of the physical models have room for improvement.

PUC-Rio - Certificação Digital Nº 1912774/CA

In summary, the positioning of this thesis (Table 2.1) concerning some of the related works of literature regarding AGVs relies specifically on path-tracking using MPC, tire modeling, and vehicle models. Moreover, Fig. 2.1 presents an overview of the thesis position regarding path-tracking using MPC control.

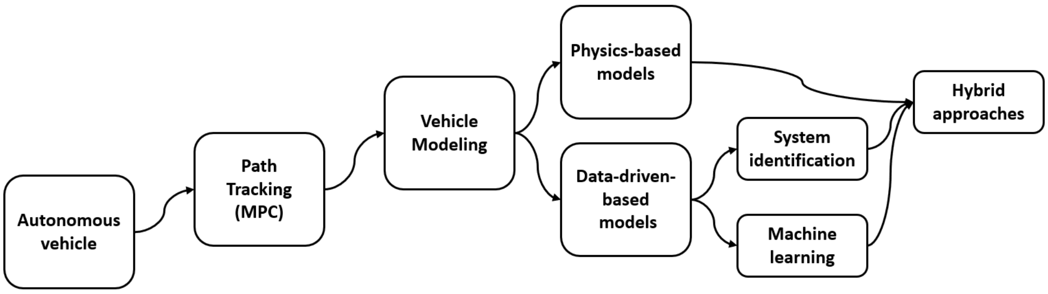

Figure 2.1: Thesis position regarding AGVs: the present thesis focuses on path-tracking using MPC control, vehicle models based on physics, and data-driven approaches.

PUC-Rio - Certificação Digital Nº 1912774/CA

2.4 Outline

The present thesis is organized into four main parts, being this introductory part (Part 1) the first. Part 2 presents a theoretical background regarding the subjects used in the scope of this thesis. Part 3 elucidates the original contributions and the proposed methodologies' results. Lastly, Part 4 brings conclusions, suggestions for future research publications, and the references used in this thesis. The parts and their respective chapters are divided as:

Part I: This part of the thesis is devoted to establishing a general introduction to the present work.

- Chapter 1: An introductory chapter is stated about the contextualization, motivation, and goals of this thesis;
- Chapter 2: This chapter details the literature review, together with the related original contributions, and the outline of this thesis;

Part II: This part of the thesis deals with concepts about vehicle system modeling, system identification, machine learning, and predictive control.

- Chapter 3: This chapter is devoted to exposing the basics of ground vehicle modeling;

Table 2.1: Summary of selected references describing techniques for vehicle models for efficient MPC.

	ROKONUZZAMAN et al. [28]	ROKONUZZAMAN et al. [18]	GUO et al. [73]	SPIELBERG et al. [3]	VICENTE et al. [37]	JAMES et al. [43]	DIAS et al. [45]	LIO et al. [89]	KAPANIA et al. [98]	PAN et al. [97]	This thesis
MPC											
Linear	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
Nonlinear	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓
Data-driven	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓
Tire models											
Linear	✓	✓	✗	✗	✓	✓	✗	✗	✗	✗	✓
Nonlinear	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓
ML models	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Vehicle models											
Linear	✓	✓	✗	✓	✓	✗	✓	✓	✓	✗	✓
Nonlinear	✓	✓	✓	✗	✓	✓	✗	✗	✗	✓	✓
Data-driven	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
Hybrid	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓

PUC-Rio - Certificação Digital Nº 1912774/CA

- Chapter 4: Fundamentals for system identification with black-box models and the mathematical formulation of the artificial neural networks are stated in the present chapter;
- Chapter 5: This chapter deals with the details regarding the mathematical formulation of the model predictive control;

Part III: The main contributions of the present thesis are stated in this part.

- Chapter 6: Machine learning architectures are applied for tire learning to enhance model predictive control efficiently;
- Chapter 7: This chapter is devoted to detail the results obtained by applying ARX models to enhance model predictive control;
- Chapter 8: The novel hybrid approach, combining data-driven approaches with black-box modeling of the discrepancies are given in this chapter to enhance vehicle modeling;

Part IV: Final comments, future research directions, and proposed work plan are given.

- Chapter 9: General conclusions obtained in the scope of the present work are given with a list of contributed publications. Moreover, future research directions are given at the end of this chapter;

Appendix: The appendix gives insight and description of different systems, making the document more easily readable.

- Appendix A: This appendix gives details about the case studies used in the scope of the present thesis, together with related references;

PUC-Rio - Certificação Digital Nº 1912774/CA

Part II

Theoretical Background

PUC-Rio - Certificação Digital Nº 1912774/CA

3
Physics-based vehicle models

The present chapter aims to state basic concepts regarding physics-based vehicle models. The first part is devoted to presenting the single-track model commonly used for control applications [107], which will be the focus of the first and second contributions. Secondly, other two approaches are presented, distinguishing the longitudinal [43] and lateral vehicle dynamics [72, 96, 108], which will be the focus of the third contribution.

3.1
Single-track model

The single-track model (Fig. 3.1) is a vehicle dynamic model widely used for control applications [107]. The vehicle is assumed to travel on a rigid path with suspension and aerodynamic aspects neglected. The vehicle motion can be described in the vehicle (x and y) and global frames (X and Y). Moreover, ψ is the vehicle's and global frames' yaw angle. Based on Newton's theorem, the governing equations of motion considering longitudinal and lateral dynamics can be expressed as

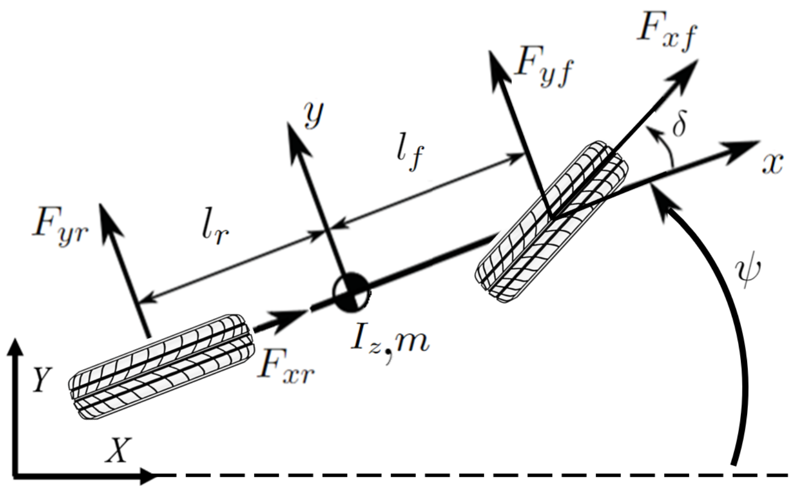

Figure 3.1: Single-track dynamic model.

PUC-Rio - Certificação Digital Nº 1912774/CA

$$m\ddot{x} = m\dot{y}\dot{\psi} + 2(F_{xf}\text{cos}\delta - F_{yf}\text{sin}\delta) + 2F_{xr} \, , \tag{3-1}$$

$$m\ddot{y} = -m\dot{x}\dot{\psi} + 2(F_{xf}\text{sin}\delta + F_{yf}\text{cos}\delta) + 2F_{yr} \, , \tag{3-2}$$

$$I_z\ddot{\psi} = 2l_f(F_{xf}\text{sin}\delta + F_{yf}\text{cos}\delta) - 2l_rF_{yr} \, , \tag{3-3}$$

$$I_{\omega f}\dot{\omega}_f = -2F_{xf}r_d + T_f \, , \tag{3-4}$$

$$I_{\omega r}\dot{\omega}_r = -2F_{xr}r_d + T_r \, , \tag{3-5}$$

where I_z and m are the moment of inertia about the yaw axis and vehicle mass, respectively; l_f and l_r are the front and rear axle distance from the center of gravity (CG). Besides, F_{xf}, F_{xr}, F_{yf}, and F_{yr} are the longitudinal and lateral forces applied to the front and rear axles, respectively. The parameters $I_{\omega f}$ and $I_{\omega r}$ are the mass moment of inertia of the wheels, r_d is wheel radius, T_f and T_r are the torque applied to the driven wheels when considering a front- and rear-wheel drive vehicle, respectively. The wheel rotational speeds for front and rear wheels are given by ω_f and ω_r, respectively. Finally, δ is the front wheel angle which can be defined as the relation between steering wheel angle δ_{sw} and steering transmission ratio i_{sw} [109], as follows

$$\delta = \frac{\delta_{sw}}{i_{sw}} \, . \tag{3-6}$$

The vehicle motion over the global frame can be derived from the kinematic model as

$$\dot{X} = \dot{x}\text{cos}\psi - \dot{y}\text{sin}\psi \, , \tag{3-7}$$

$$\dot{Y} = \dot{x}\text{sin}\psi + \dot{y}\text{cos}\psi \, . \tag{3-8}$$

PUC-Rio - Certificação Digital Nº 1912774/CA

3.1.1
Linear tire model

The tire-road interaction is one of the main aspects of the performance of ground vehicles since tires connect the vehicle to the environment [50]. Therefore, friction aspects under the tire (contact patch region) are essential for vehicle safety. Under situations with small slip angles, the lateral force and slip angles produce a linear relation expressed as [83]:

$$F_{yf} = C_f\alpha_f \, , \tag{3-9}$$

$$F_{yr} = C_r\alpha_r \, , \tag{3-10}$$

where the front and rear slip angles are given by α_f and α_r, respectively, and can be derived from Eqs. (3-11) and (3-12). Besides, the cornering stiffness can be established for the front (C_f) and rear (C_r) wheels.

$$\alpha_f = \delta - \frac{\dot{y} + l_f\dot{\psi}}{\dot{x}}\,, \tag{3-11}$$

$$\alpha_r = -\frac{\dot{y} - l_r\dot{\psi}}{\dot{x}}\,. \tag{3-12}$$

On the other hand, the longitudinal slip can be derived as

$$s = \frac{\dot{x} - \omega r_d}{\max[\dot{x}, \omega r_d]}\,. \tag{3-13}$$

The friction coefficients in longitudinal (μ_x) and lateral (μ_y) directions represent the relation between tire efforts on the tire as [110], in which F_z is the vertical effort on the tire.

$$\mu_x(s) = \frac{F_x}{F_z}\,, \tag{3-14}$$

$$\mu_y(\alpha) = \frac{F_y}{F_z}\,. \tag{3-15}$$

3.1.2 Nonlinear tire model

PUC-Rio - Certificação Digital Nº 1912774/CA

Unlike the linear tire model, nonlinear models for tires became vital because they can accurately predict at larger slip angles [28]. Thus, numerous studies are available regarding tire models, including the Brush model and Pacejka model [49], and the Tmeasy model [111]. The Pacejka tire model is also known as "Magic Formula" since there is no particular physical basis. Thus, the mathematical representation is based on experimental data, and curve fitting to approximate tire curves [83]. Here, a simplified version of the Pacejka formulation (Eqs. (3-16)-(3-17)) represents the tire-road interaction as follows

$$F = D\mathrm{sin}(C\mathrm{atan}(B\phi)) + S_v\,, \tag{3-16}$$

$$\phi = (1 - E)(\lambda + S_h) + (E/B)\mathrm{arctan}(B(\lambda + S_h)), \tag{3-17}$$

where λ parameter is replaced by longitudinal slip (s) or lateral slip angle (α) for longitudinal and lateral efforts on the tire (F), respectively. Moreover, the parameters B, C, D, and E are the stiffness and shape factors, peak value, and curvature factor, respectively. In addition, S_v and S_h are vertical and horizontal shift coefficients [49].

3.2
Another perspective of physics-based vehicle modeling

This section introduces other mathematical formulations of vehicle physics-based models distinguishing longitudinal and lateral dynamics.

3.2.1
Longitudinal dynamics

The longitudinal model described in [43] is used with a modification regarding the tractive and braking efforts. Here, a modification is implemented in the model to receive the driver's command, specifically the pedal's driver, in terms of throttle and braking efforts in percentage, which allows the modeling to be directly used for higher-level control approaches. Thus, based on Newton's theorem, the longitudinal motion can be expressed as

$$m\dot{v}_x = k_e P(t) - k_d v_x - mgk_r \,, \tag{3-18}$$

where m is the vehicle mass, v_x is the vehicle's longitudinal velocity, and g is the acceleration due to gravity. The force resultant (traction and braking) transmitted to the wheels is modeled by $k_e P$, where P approximates the driver's pedal signal. Here, the difference between the throttle and braking signals represents the driver's pedal signal. In addition, k_e groups the driveline gear ratio and the system's efficiency. The drag force is given by a linear function $k_d v_x$. Finally, the k_r is the rolling friction parameter.

PUC-Rio - Certificação Digital Nº 1912774/CA

3.2.2
Lateral dynamics

The bicycle model is a linear dynamic model commonly used for control solutions regarding lateral vehicle dynamics. The vehicle is assumed to travel on a rigid path with suspension and roll aspects neglected. Moreover, the lateral force remains in a linear function using a cornering stiffness coefficient [83]. This model is characterized by two degrees of freedom, composed of lateral velocity (v_y) and yaw rate ($\dot{\psi} = r$), and can be defined as

$$\dot{v}_y = -\left(\frac{C_f + C_r}{mv_x}\right) v_y - \left(\frac{C_f l_f - C_r l_r}{mv_x} + v_x\right) r + \frac{C_f \delta}{m} \,, \tag{3-19}$$

$$\dot{r} = -\left(\frac{C_f l_f - C_r l_r}{I_z v_x}\right) v_y - \left(\frac{C_f l_f^2 - C_r l_r^2}{I_z v_x} + v_x\right) r + \frac{C_f l_f \delta}{I_z} \,, \tag{3-20}$$

where I_z is the moment of inertia about the yaw axis, l_f and l_r are the front and rear axle distance from the center of gravity. Besides, the cornering stiffness can be established for the front (C_f) and rear (C_r) wheels, respectively. Finally, δ is the front wheel angle.

3.3 Summary

Due to computational development, many vehicle designers have chosen to perform computational tests instead of tests with physical prototypes. A large part of this is because tests with physical prototypes have a high cost of preparation and implementation, in addition to a great demand for time to be completed, and are currently used only as the last step for the final validation of the models and computational tests. When done well, computational tests have results that are very close to reality. Thus, several techniques have been used for modeling mechanical processes that allow more accurate evaluations of their behavior. Particularly for the AGVs field, developing representative models regarding vehicle dynamics is essential since they can be used to derive control laws. Thus, this chapter presented models derived from physics combining Newton's second law and empirical observations. Dynamic models based on physics laws and constraints make them easy to understand and interpret. Linear and nonlinear vehicle models were presented with tire modeling regarding longitudinal and lateral dynamics.

The physics-based models herein presented are used through all the original contributions (Sections 6-8) to be compared with the data-driven approaches. In the first contribution, dynamic models with the Pacejka tire model are compared with the proposed neural tires (neural architectures are the focus of Section 4.3). The second contribution relies on data-driven approaches to model a vehicle and perform path-tracking. The MPC control based on the data-driven approach is compared with the traditional one based on linear and nonlinear vehicle models. Lastly, the third original contribution deals with the hybrid approach that combines vehicle models with modeling the discrepancies between measured and predicted data from a vintage racing car. The main objective is to enhance the vehicle model outputs.

PUC-Rio - Certificação Digital Nº 1912774/CA

4
Machine learning and black-box system identification

The present chapter aims to state basic concepts regarding machine learning using ANNs and black-box system identification. The first section is devoted to presenting ARX models and validation metrics. Next, the nonlinear ARX models are described. Then, ANNs formulation, including RBF and MLP architectures, is given. Finally, the state-space approach is detailed when applied to system identification. It is interesting to point out that each section presents examples that the interested reader may reproduce.

4.1
ARX Models for system identification

A critical disadvantage of nonlinear dynamic models, consequently affecting the NMPC controller, is that knowing prior information about the dynamic system is necessary. In addition, the MPC procedure based on high-fidelity models is time-consuming [37]. On the other hand, linear models depend on their operating point [45]. Therefore, this work applies data-based system identification to overcome these difficulties. For this, an ARX model (Fig. 4.1) is applied, where $y(k)$, $u(k)$ are the model's output and input, respectively; $\xi(k)$ represents modeling error, measurement error, offset or noise for a sample k. In addition, $G(z^{-1})$ and $H(z^{-1})$ are the dynamic transfer function and the noise transfer function, respectively, and can be expressed as

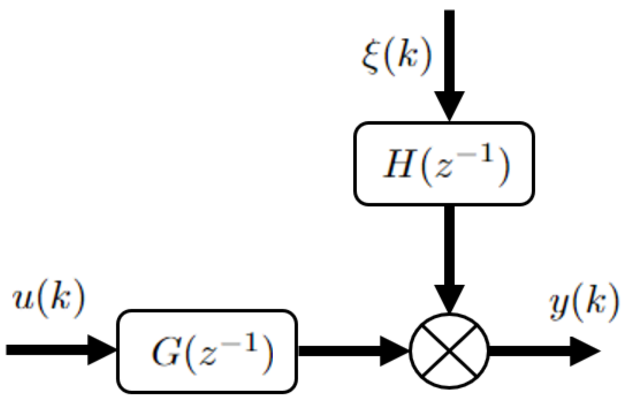

Figure 4.1: Structure of an ARX model.

PUC-Rio - Certificação Digital Nº 1912774/CA

$$\begin{cases} H(z^{-1}) = \dfrac{B(z^{-1})}{A(z^{-1})} \\ G(z^{-1}) = \dfrac{1}{A(z^{-1})} \end{cases}, \tag{4-1}$$

where

$$\begin{cases} A(z^{-1}) = 1 - a_1 z^{-1} - ... - a_{na} z^{-nb} \\ B(z^{-1}) = b_1 z^{-1} + ... + b_{nb} z^{-na} \end{cases}, \tag{4-2}$$

where a_1, a_2,..., a_{na} and b_1, b_2,..., b_{nb} are coefficients regarding delayed input and output samples, respectively. Moreover, z^{-1} is the delay coefficient.

From Fig. 4.1, it is possible to determine that:

$$y(k) = \frac{B(z^{-1})}{A(z^{-1})} u(k) + \frac{1}{A(z^{-1})} \xi(k)\,. \tag{4-3}$$

Finally, the ARX model can be parameterized as follows,

$$y(k) = (1 - A(z^{-1}))y(k) + B(z^{-1})u(k) + \xi(k)\,, \tag{4-4}$$

and therefore, the ARX model can be represented as

$$\begin{aligned} y(k) = -a_1 y(k-1) - a_2 y(k-2) - ... - a_{na} y(k-na) + \\ + b_1 u(k-1) + b_2 u(k-2) + ... + b_{nb} u(k-nb) + \xi(k)\,. \end{aligned} \tag{4-5}$$

The ARX model in Eq. (4-5) has $na + nb$ coefficients to estimate. Therefore, the batch least squares algorithm is used over input and output data pairs. The coefficients to be found have to minimize a cost function regarding the model's error for one-step-ahead prediction. Then, Eq. (4-5) can be rewritten using a linear regression model as

$$y(k) = \phi^T(k)\hat{\theta} + \xi(k)\,, \tag{4-6}$$

where $\phi^T(k)$ is a vector containing input and output data, $\hat{\theta}$ is the vector of coefficients to be estimated. Moreover, the residuals (ξ) is defined as

$$\xi(k) = y(k) - \hat{y}(k)\,, \tag{4-7}$$

where $y(k)$, and $\hat{y}(k)$ are the measured and predicted outputs, respectively. When enormous data are considered, Eq. (4-6) must be rewritten in matrices form as

$$Y_t = \Phi\hat{\theta} + \Xi\,, \tag{4-8}$$

where Y_t is the target measurement vector, Φ is the regressors matrix, and Ξ

PUC-Rio - Certificação Digital Nº 1912774/CA

is the residual vector. Finally, the regressors matrix $\hat{\theta}$ can be determined via the minimum norm procedure

$$\hat{\theta} = [\Phi^T \Phi]^{-1} \Phi^T Y_t \,. \tag{4-9}$$

4.1.1 Model validation and metrics

In order to evaluate an identified model, some procedures and metrics can be used, depending on the model's purpose. Then, two different methods are presented in this work to predict models. First, the One-Step-Ahead prediction uses past *measured values* to determine the output. In contrast, the Free-Run prediction uses past *predicted values* to calculate the future predictions [112]. Let $\hat{y}_s$ denote the FR prediction of $y(k)$ at a sample k. In addition, for brevity, it is possible to write an ARX model with the case $n_a = n_b$:

$$\begin{cases} \hat{y}_s(1) = y(1) \\ \hat{y}_s(2) = y(2) \\ \vdots \\ \hat{y}_s(na) = y(na) \\ \hat{y}_s(na+1) = [\hat{y}_s(na), \hat{y}_s(na-1), ..., \hat{y}_s(1), u(nb), u(nb-1), .., .u(1)] \\ \vdots \\ \hat{y}_s(k) = [\hat{y}_s(k-1), \hat{y}_s(k-2), ..., \hat{y}_s(k-na), u(k-1), u(k-2), ..., u(k-nb)] \\ \vdots \end{cases}$$

The error in the FR predictions are accumulated through the prediction and easily shows whether the model is valid. On the other hand, the OSA predictions present a reset at each interaction. Therefore, it is complicated to validate the model using only OSA predictions.

Through this thesis, the prediction models may be compared using different methods. First, error-based metrics include the Root Mean Squared Error (RMSE), the Multiple Correlation Coefficient (R^2) [113], and the normalized fit metric [114] between measured and predicted data as follows

$$\text{RMSE} = \sqrt{\frac{1}{N_l} \sum_{i=1}^{N_l} [y - \hat{y}]^2} \,, \tag{4-10}$$

$$R^2 = 1 - \frac{\sum_{t=1}^{N_l} [y - \hat{y}]^2}{\sum_{t=1}^{N_l} [y - \overline{y}]^2} \,, \tag{4-11}$$

$$F = 100 \left(1 - \frac{\| y - \hat{y} \|_2}{\| y - \overline{y} \|_2} \right) \,, \tag{4-12}$$

PUC-Rio - Certificação Digital Nº 1912774/CA

where y is the reference signal, $\hat{y}$ is the predicted data, $\overline{y}$ is the mean value of the reference signal, and N_l is the length of the data vector. Specifically for the fit metric, a perfect fit is indicated by $F = 100\%$, while a negative fit means a poor fit.

A correlation analysis between the measured and predicted data is also performed. The correlation approach can be determined by the variance-accounted-for metric (VAF), given by:

$$VAF = 100\left(1 - \frac{\mathrm{var}(y - \hat{y})}{\mathrm{var}(y)}\right). \tag{4-13}$$

A value of $VAF = 100\%$ ensures that the predicted model can identify the variance in the measured data. However, the model mismatches due to the noise in the data are responsible for that value not being expected.

Example 1: *The present example shows the importance of calculating the OSA and FR prediction errors to validate an ARX model.*

The flexible robotic arm detailed in Section A.1 is used for this example. During the study of the ARX model, different combinations of na and nb are defined in intervals of [1, 10]. The best four cases are selected for each type of simulation (FR and OSA). Finally, as model validation data, the final part was used as the estimation data (15% of the entire dataset). The results obtained are listed in Table 4.1, where the values of the RMSE and R^2 for the estimation and validation data consider both the OSA and FR simulations.

Table 4.1: Results in terms of R^2 and RMSE metrics, considering OSA and FR simulations for the robotic arm example.

	Case	na	nb	Estimation		Validation	
				RMSE $\times 10^{-4}$	R^2	RMSE $\times 10^{-4}$	R^2
OSA	38	4	8	5.44	0.999	3.77	0.999
	39	4	9	5.32	0.999	3.60	0.999
	48	5	8	5.33	0.999	3.83	0.999
	50	5	10	5.08	0.999	3.45	0.999
FR	38	4	8	145	0.997	105	0.997
	39	4	9	91.1	0.999	104	0.998
	48	5	8	179	0.996	109	0.997
	50	5	10	95.2	0.999	107	0.997

After checking the best cases among all tests, the combination $na = 4$ and $nb = 9$ presented the best results considering the FR simulation. Due to the characteristic of the step-ahead method, the errors are redefined at each iteration, resulting in similar results.

On the contrary, the free simulation accumulates the error at each iteration, making verifying discrepancies in each case easier. It is also noted

PUC-Rio - Certificação Digital Nº 1912774/CA

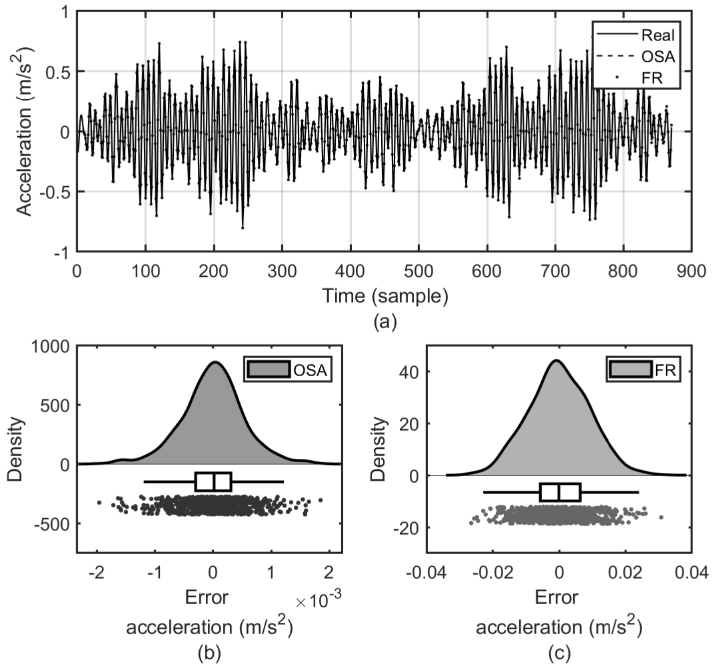

Figure 4.2: Example 1: a) OSA and FR predictions considering estimation phase. Raincloud considering (b) OSA and (c) FR model errors and the real data.

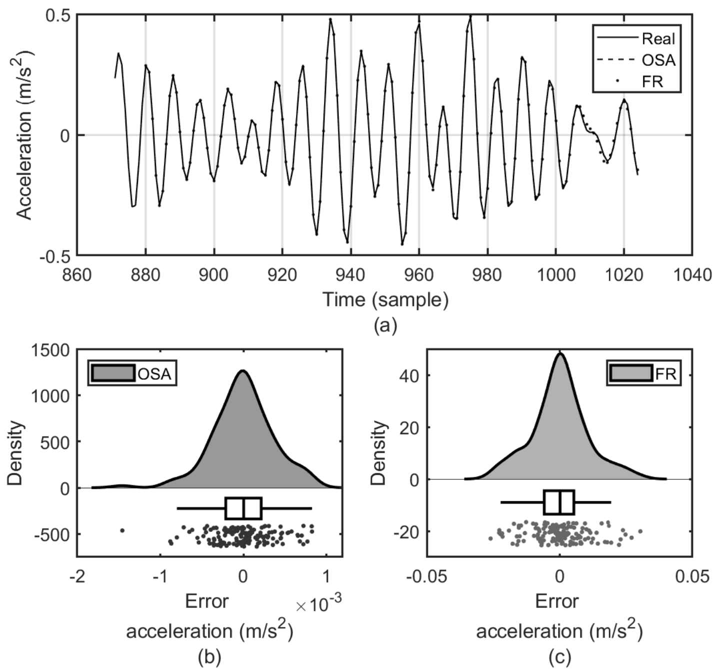

Figure 4.3: Example 1: a) OSA and FR predictions considering validation phase. Raincloud considering (b) OSA and (c) FR model errors and the real data.

PUC-Rio - Certificação Digital Nº 1912774/CA

that the increase in the parameters na and nb does not drastically influence the result. However, the greater the number of parameters more complex it tends to become visible the influence of each one of them in the system. This may hinder the implementation of advanced control systems. Figs. 4.2 and 4.3 show the comparison between the measured data and the data estimation and validation, considering the OSA and FR simulations.

4.2 NARX models for system identification

The Nonlinear AutoRegressive with eXogenous inputs (NARX) models is applied for a vehicle and discrepancies learning and can be defined as

$$\begin{aligned} y(k) = F[\phi(k)] = F[&y(k-1), ..., y(k-na), \\ &u(k-1), ..., u(k-nb) + \xi(k)] \, , \end{aligned} \tag{4-14}$$

where the difference between the linear approach aforementioned is the use of the function F[$\phi(k)$], which in this work is set as an artificial neural network (the focus of the Section 4.3) that can be defined as the concatenation of multiple hidden layers. In this sense, the ADAM learning algorithm is a first-order stochastic gradient descent that allows an individualized step adaptation of the free parameters in the search stage. This occurs due to the use of an individual learning rate [115].

Example 2: *The present example demonstrates the application of a NARX model.*

The simulated system detailed in Section A.2 is used for this example. The main objective of this example is to identify a simulated system using the NARX model. The true lags are known, and the application of the proposed method relies on test NARX models with neural networks varying the number of neurons with networks using Exponential Linear Unit (ELU) activation through hidden layers varying from 2 to 10. After some trial and error, the test was conducted using 50 neurons and three hidden layers using the ADAM algorithm with a learning rate set 10^{-3}.

Table 4.2: Results in terms of R^2 and RMSE metrics, considering OSA and FR predictions for Narendra and Parthasarathy's example.

	Estimation		Validation	
	RMSE	R^2	RMSE	R^2
OSA	0.0489	0.9765	0.0541	0.9648
FR	0.0513	0.9720	0.0481	0.9791

PUC-Rio - Certificação Digital Nº 1912774/CA

Table 4.2 shows the capability of the present approach to approximate the simulated system. The model's precision is also shown in Figs. 4.4 and 4.5, which depict the system outputs and errors through the rainclouds considering OSA and FR predictions, both in the estimation and validation phases.

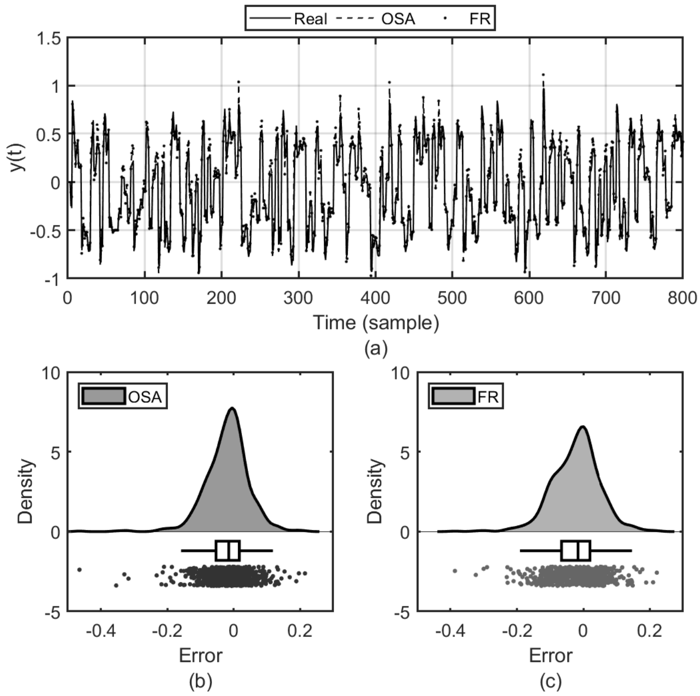

Figure 4.4: Example 2: a) OSA and FR predictions for Narendra and Parthasarathy's example considering the estimation phase. Raincloud considering (b) OSA and (c) FR model errors and the real data.

4.3 Artificial neural networks

ANNs comprise neurons that form a complex architecture considering inputs to produce approximations as a mapping [116]. The ANN architectures can adapt from data by adjusting their connections and coefficients [117]. The neural network as a topic rises from the work proposed by [118] dealing with problems regarding the nervous system. After that, some structures emerged in the NN topic as the perceptron [119] and the backpropagation algorithm [120].

Reference [121] presented the advantages and drawbacks of using ANNs in modeling systems. As advantages, ANNs are considered simple architectures, conceptually easy to comprehend, and can be defined with a couple of equations. However, drawbacks regarding NNs are that these models are often

PUC-Rio - Certificação Digital Nº 1912774/CA

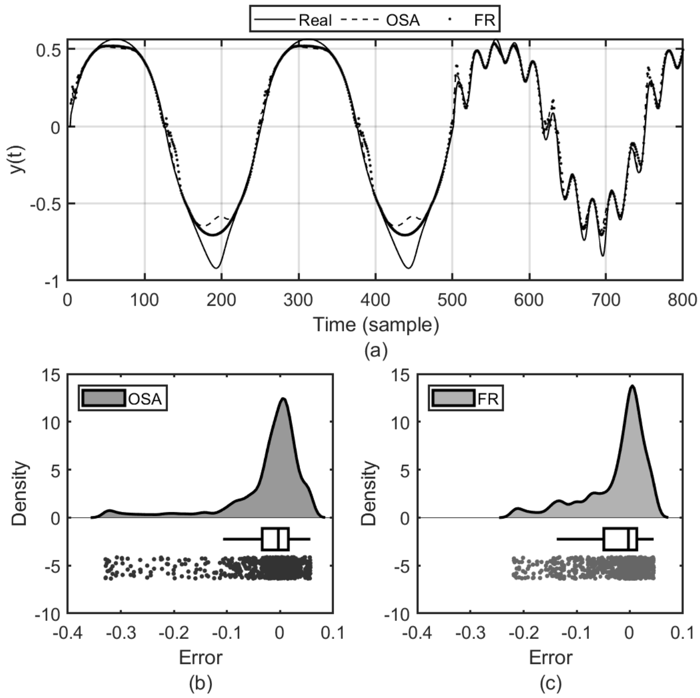

Figure 4.5: Example 2: (a) OSA and FR predictions for Narendra and Parthasarathy's example considering the validation phase. Raincloud considering (b) OSA and (c) FR model errors and the real data.

PUC-Rio - Certificação Digital Nº 1912774/CA

opaque, and it is challenging to capture dynamical system aspects and satisfy primary physics constraints.

This work briefly describes radial basis functions and multilayer perceptron networks (two of the most widely used ANNs) in the following.

4.3.1 Radial basis functions neural networks

The RBF neural network (RBFNN) is a network that uses radial basis functions as an activation function. RBF networks are built considering the input, hidden, and output layers (Fig. 4.6). In the input layer, input data are connected to source nodes. In contrast, in the hidden layer, different neurons of the activation functions have their output data weighted and then summed towards the output layer. The output of the neuron model is expressed as:

$$\hat{y}(t) = \sum_{i=1}^{M} W_i \phi(r(t), c_i, \sigma_i)\,, \tag{4-15}$$

where $\hat{y}(t)$ is the neuron output, M is the quantity of neurons inside the hidden layer, W_i is the output weights, $r(t)$ is the input vector, σ_i and c_i are the width and the center of the i-th hidden node, respectively. The function $\phi(\cdot)$ is known as the activation function, and the designers can choose it according to their

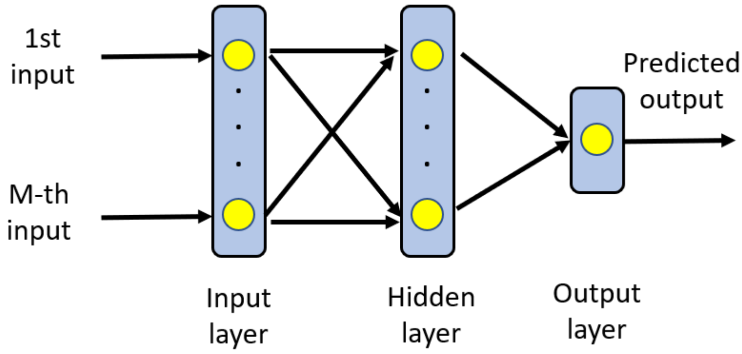

Figure 4.6: Scheme of an RBF Network with input, hidden, and output layers.

problem. Typical choices for radial basis functions networks are the following [112, 122]

– Gaussian:

$$\phi(l) = \exp\left(\frac{l^2}{\sigma^2}\right) , \tag{4-16}$$

– Cubic:

$$\phi(l) = l^3 , \tag{4-17}$$

– Thin-plate spline:

$$\phi(l) = \frac{l^2}{\sigma}\log\left(\frac{l}{\sigma}\right) , \tag{4-18}$$

– Multiquadratic:

$$\phi(l) = \sqrt{l^2 + \sigma^2} , \tag{4-19}$$

where l represent the norm between the ANN input r to a given center c, i.e. $l = \|r - c\|$. In this thesis, the multi-quadratic function is used as the activation function for RBF networks.

Some parameters of an RBFNN can be adjusted to obtain a better final result, among them the number of neurons in the hidden layer, the width and the position of the RBFs' centers, and the output weights. In order to achieve results with accuracy, the RBFNN parameters are defined using a supervised method. To this end, in this work, the minimization of the sum of squared errors is assumed and solved by an optimization problem to be discussed in Section 6.1.1.

PUC-Rio - Certificação Digital Nº 1912774/CA

4.3.2
Multilayer perceptron networks

MLP architectures were proposed to solve nonlinearly separable problems, including one or more hidden layers (Fig. 4.7). Mathematically, MLP networks are complex and became viable when researchers started to use the backpropagation algorithm [117]. The output of the MLP networks can be defined as

$$\hat{y}_j(t) = \phi\left(\sum_{i=1}^{Z} W_{ij}x_{ij} + b_j\right) , \tag{4-20}$$

where $\hat{y}_j(t)$ is the MLP output, Z is the number of hidden layers, W_{ij} are the weights between the i-th neuron, in the prior layer, and the j-th neuron in the actual layer, b_j is the bias weight, i is the number of neurons connected to the j-th neuron. Moreover, x_{ij} is the input data from the i-th neuron to the j-th neuron.

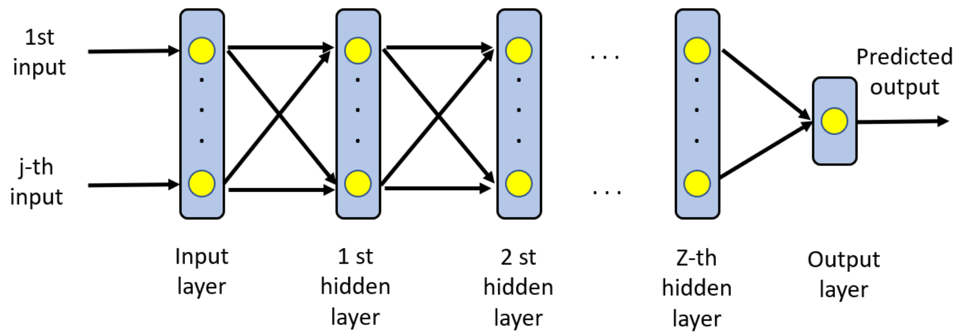

Figure 4.7: Scheme of an MLP Network with input, multiple hidden, and output layers.

In this work, the MLP networks with nodes connected by a feed-forward approach, activated by the sigmoid function that allows training through the backpropagation algorithm [123].

Example 3: *In the present example, the MLP and RBF architectures are applied to demonstrate the capacity of the ANNs.*

For this example, the cornering force is used as a function of the slip angle from real data detailed in Section A.3.

Considering the RBF and MLP neural networks, the limits of the search of the network coefficients are summarized in Table 4.3 after some trial and error. The RBF and MLP networks were also simulated considering multi-quadratic with three neurons and two neurons in two layers with sigmoid function, respectively. Finally, the optimization problems for RBF and MLP models rely on the error minimization between predicted and real data. The procedure is performed using the interior-point solver Ipopt [124] with the

PUC-Rio - Certificação Digital Nº 1912774/CA

open-source software - CasAdi [125]. CasADi is a software that uses a symbolic framework to enhance numeric optimization [125].

Table 4.3: Limits of search of the neural networks coefficients.

Network	Symbol	Definition	Lower limit	Upper limit
RBF	c	center	-1	1
	σ	width	0.01	7
	W	weights	-20	20
MLP	b	bias parameters	-2	2
	W	weights	-2	2

Fig. 4.8.a shows the predictions performed by the networks. It is possible to see a prediction accuracy with similar curve prediction from both results corroborated by the raincloud plots (Fig. 4.8.b-c). Table 4.4 shows the results for the RBF and MLP architectures regarding error-based metrics.

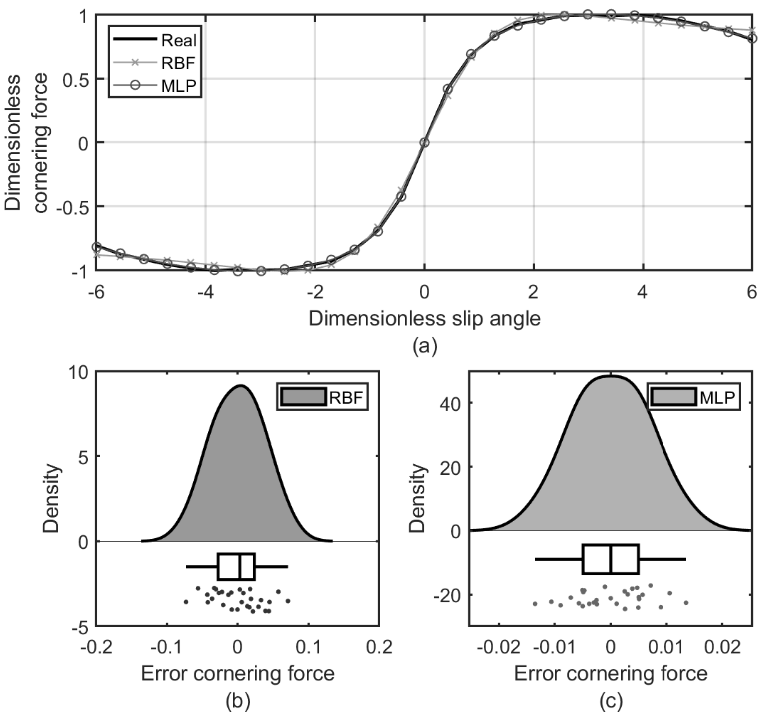

Figure 4.8: Example 3 (simulated system):(a) Dimensionless cornering force approximation using RBF and MLP networks. Raincloud considering (b) RBF and (c) MLP model errors and the real data.

4.4 State-space data-driven model

Another data-driven approach used for vehicle modeling is the subspace system identification approach, successfully implemented for vehicle modeling

PUC-Rio - Certificação Digital Nº 1912774/CA

Table 4.4: Results in terms of R^2 and RMSE metrics, and solver average computational time for all the architectures tested considering simulated tire data.

Model {Architecture}	R^2	RMSE
RBF {3}	0.9984	0.0347
MLP {2 2}	0.9995	0.0202

in recent literature [37, 43]. Thus, a linear time-invariant (LTI) model is required to represent the measurements best:

$$\dot{x}_d = A_d x_d + B_d u_d \,, \tag{4-21}$$

$$y_d = C_d x_d + w \,, \tag{4-22}$$

where w is the noise, u_d and y_d are the input and output, respectively. Considering the subspace methodology, the matrices A_d, B_d, and C_d are commonly defined based on one's knowledge of the system. On the other hand, the values inside the matrices are defined by the algorithm that aims to explain the output data based on the inputs applied.

Example 4: *The present example is given to demonstrate the application of the state-space data-driven model.*

The Electro-Mechanical Positioning System (EMPS) system detailed in Section A.4 is used. The main objective of this example is to identify a state-space data-driven model based on the measured data obtained from the EMPS experiment. Following the procedure adopted and successfully implemented in [43], a second-order ($n_d = 2$) state-space parameters were estimated using a prediction error method (PEM) [114], generally solved by employing a numerical search algorithm. In this example, the algorithm is initialized using a subspace state-space system identification (N4SID) algorithm. The System Identification toolbox of MATLAB®software is used to implement the estimation of the state-space parameters. Besides, the "*ssest*" function that combines the N4SID state-space initialization with the PEM estimation is also applied. Finally, the output given by the input data is the motor force, and the output data relies on the position and velocity data.

The data set was divided into training and test split of 60/40%, respectively. For brevity, the present example only focuses on the test data set. Thus, the results performed by the state-space data-driven method can be seen in Figs. 4.9 and 4.10, while Table 4.5 summarizes their performance metrics. In addition, the values of the optimized parameters for the state-space (with mean

PUC-Rio - Certificação Digital Nº 1912774/CA

of data samples removed and normalized data) can be found in Table 4.6.

Table 4.5: Performance metrics - EMPS model.

Position			Velocity		
$F\%$	$VAF\%$	RMSE	$F\%$	$VAF\%$	RMSE
90.60	99.16	0.009	82.15	96.83	0.012

Table 4.6: Parameters estimates for the state-space model.

$$A = \begin{bmatrix} 0.9973 & -0.0036 \\ -0.0014 & 0.9982 \end{bmatrix} \quad B = \begin{bmatrix} 0.0010 \\ -0.0014 \end{bmatrix}$$

$$C = \begin{bmatrix} 3.7916 & 2.6788 \\ -14.6146 & -19.2883 \end{bmatrix}$$

PUC-Rio - Certificação Digital Nº 1912774/CA

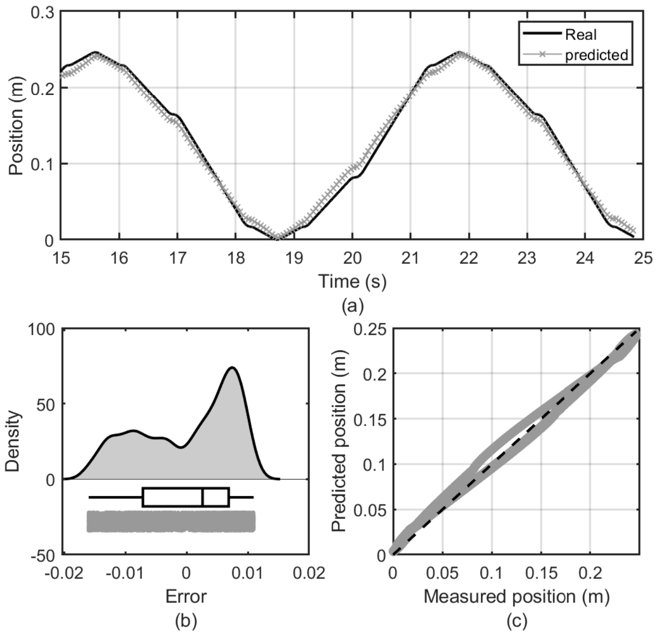

Figure 4.9: Example 4: (a) Comparison of measured and predicted position using the state-space approach considering test data. (b) Raincloud considering the error between measured and predicted data. (c) Correlation between the measured and the simulated position.

From the Figs. 4.9.b and 4.10.b, one can see raincloud plots considering the error between predicted and measured data. Although the position prediction is more accurate regarding the comparison metrics, its error distribution does not produce results with a mean next to zero. Besides, the correlation comparisons are given in Figs. 4.9.c and 4.10.c. The proposed approach can

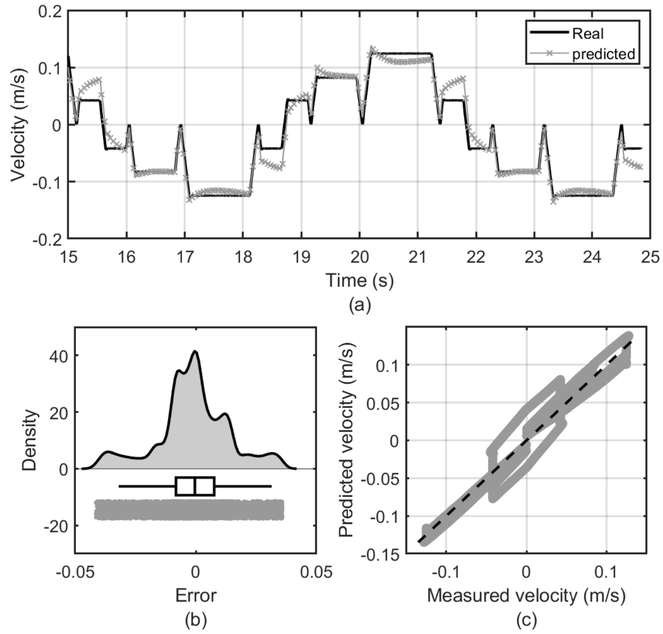

Figure 4.10: Example 4: (a) Comparison of measured and predicted velocity using the state-space approach considering test data. (b) Raincloud considering the error between measured and predicted data. (c) Correlation between the measured and the simulated velocity.

PUC-Rio - Certificação Digital Nº 1912774/CA

explain the variances in the measurements to a certain degree, with a percentage of variance accounted for 99.16% and 96.83% regarding the position and velocity data, respectively.

4.5 Summary

Potential solutions for modeling different processes are computational intelligence and system identification methods. They may be capable of accurately representing complex systems from measured data and considering the presence of disturbances. Moreover, with systems in which properties and particular characteristics are hard to be obtained, and no prior information about the process is available, the methodology arises as an alternative for modeling.

This chapter was devoted to presenting the (N)ARX models and their mathematical formulations. In addition, artificial neural networks were presented using RBFs and MLPs architectures. Finally, the state-space model in a data-driven way was presented and formulated. At the end of each section, examples were elaborated to make the present thesis more easily readable. The main topics presented in this chapter are inserted in this thesis by means of all the original contributions (Chapters 6-8). Thus, the first contribution deals with using machine learning for tire learning by using RBF and MLP architectures compared with the Pacejka tire model from the literature. Next, the second contribution deals with a combination of ARX models to model a vehicle plant accurately. Lastly, the third original contribution deals with data-driven approaches to modeling a vintage racing car, including NARX and state-space data-driven models. A hybrid approach is proposed combining the vehicle models with a discrepancy model given by a NARX model.

PUC-Rio - Certificação Digital Nº 1912774/CA

5
Model predictive control

The present chapter focuses on the mathematical formulation of the MPC controller, which will be used in this work's first and second original contributions. The contributions rely on path-tracking control using the MPC methodology. An MPC example is presented at the end of the present chapter and may be reproduced by the interested reader.

5.1
Formulation

Model predictive control is a control procedure used to predict a dynamic system's future states during a finite-time window (horizon) [30]. The basic part of the predictive controller is the prediction model. In this work, the prediction model is derived from the vehicle dynamic model with neural and Pacejka tires (Chapter 6) or the identified vehicle models (Chapter 7) in $t \in \mathbb{R}^+$ (continuous time), and it is abbreviated as

$$\dot{x}_t = f(x_t, u_t)\,, \tag{5-1}$$

where $x_t \in X \subseteq \mathbb{R}^n$ (n dimension) is the state vector, $u_t \in U \subseteq \mathbb{R}^m$ (m dimension) is the input vector, and $f : X \times U \mapsto X$ is the function that represents the dynamic model. Considering a reduced sampling interval T, the dynamic model in Eq. (5-1) can be discretized using Runge-Kutta 4th order or Taylor series. The latter approach is demonstrated as follows

$$x_{k+1} = x_k + Tf(x_k, u_k)\,, \tag{5-2}$$

where $k = 0, 1, 2, ...$ denotes the instants in discrete time such that $t = kT$. The main goal of the MPC controller, at each instant of time $k = 0, 1, 2, ...$, is to find a sequence of optimal control actions $u^*_{k,k+N-1}$ to solve the control problem that minimizes a cost function in a moving horizon window. One can design different cost functions that combine linear or nonlinear functions of the system states and inputs. Typical cost functions (such as Eq. (5-3)) aim to minimize the following:

1. the error between the state vector and its desired trajectory;

PUC-Rio - Certificação Digital Nº 1912774/CA

2. the amplitude of the control action; and

3. the variation of the control action.

$$J_N(u_{k,k+N-1}) = \sum_{i=k}^{k+N-1} \| x_i - q_i \|_Q^2 + \sum_{i=k}^{k+N-1} \| u_i - r_i \|_R^2 + \sum_{i=k+1}^{k+N-1} \| \Delta u_i \|_S^2 \,, \tag{5-3}$$

where q_k and r_k are the references to x_k and u_k, respectively; the matrices Q, R, and S, in general diagonals, give the importance to each term in the optimization procedure, and the variation of the control action is given by

$$\Delta u_k = u_k - u_{k-1} \,. \tag{5-4}$$

Each term in Eq. (5-3), reading from left to right, refers to each item 1-3 aforementioned. Note that $J_N(u_{k,k+N-1})$ is evaluated in a window of size N, and this notation shows that its argument is just $u_{k,k+N-1}$. However, for simplification the terms x_k (start condition of the moving window), $q_{k,k+N-1}$ (sequence of references to the state) and $r_{k,k+N-1}$ (references to the control action) are omitted but are needed to calculate the cost function. This work focuses only on the first (path-tracking) and third (variation of the control action) terms of the cost function, aiming mainly at comfort and accuracy during the path-tracking. Therefore, the matrices Q and S provide an essential effect on the controller performance in the present thesis. The matrix Q is responsible for giving importance to the states to be followed by the vehicle. In addition, the matrix S gives importance to the control actions to smooth vehicle states or control actions.

To apply the real-time MPC, it is necessary that the optimization problem in Eq. (5-3) be solved in a moving window of size N at every instant $k = 0, 1, 2, ...$ to get the optimal sequence of control actions as

$$\begin{aligned} \min \quad & J_N(u_{k,k+N-1}) \,, \\ \text{s.t.} \quad & u_i \in U \,, \\ & x_i \in X \,, \\ & u_{min} \leq u_i \leq u_{max} \,, \\ & x_{min} \leq x_i \leq x_{max} \,. \end{aligned} \tag{5-5}$$

where $u_{min,max}$ and $x_{min,max}$ are the constraints applied to the input and states. In addition, the predictions may be computed recursively as

$$x_{i+1} = f^d(x_i, u_i), i = k, k+1, ..., k+N-1 \,, \tag{5-6}$$

where f^d is the function that represents the dynamic model in discrete time.

PUC-Rio - Certificação Digital Nº 1912774/CA

The computational effort required is higher when compared to simple linear control laws. It is a consequence to obtain a gain in performance. An important detail regarding the MPC controller is the application of the control action. In a window $[k, k+N-1]$ of size N, the optimal control action u_k^* is applied for each instant $k = 0, 1, 2, \ldots$. The other control actions are used to initialize the algorithm optimization in the next iteration when the process is repeated. The general overview of the MPC scheme is presented in Fig. 5.1. In summary, the MPC algorithm can be computationally implemented as follows

1. To define the MPC parameters (horizon and matrices);
2. To initialize $k = 0$;
3. To read $x_k = 0$;
4. To obtain $u^*_{k,k+N-1}$ from the minimization of the cost function;
5. To apply the control action u_k^* in the system;
6. $k = k + 1$;
7. Continue with step 2;

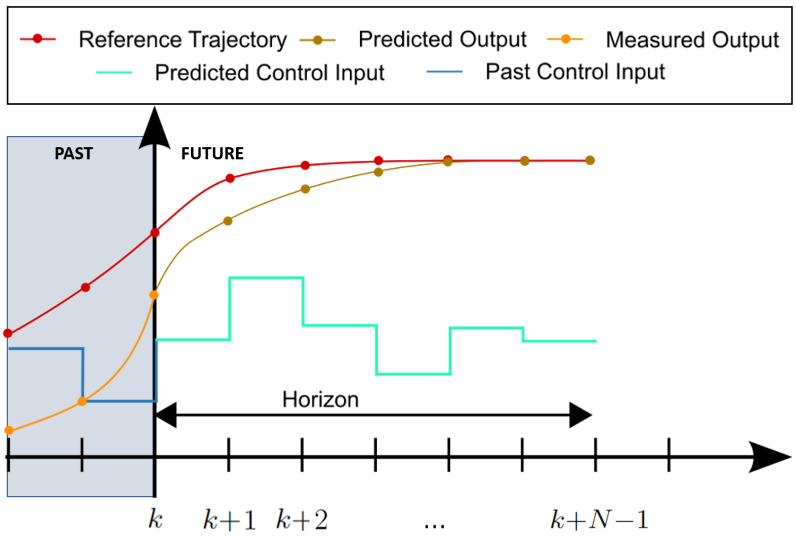

Figure 5.1: Overview of the MPC controller. At each instant, intending to find the best control sequence over a future horizon, one has to: 1) get new measurements to update the current state, 2) solve the optimization problem inside the prediction horizon, and 3) apply only the first optimal move u_k^*, discarding the remaining samples.

PUC-Rio - Certificação Digital Nº 1912774/CA

5.2 Multiple shooting

The problem defined in Eq. (5-3) involves a recursive procedure regarding the prediction model N times, aiming to calculate the cost function. This can lead to overly complex cost functions, which is undesirable from any point of view. In addition, the problem optimization problem is solved at each sampling interval, which can make the application of MPC impossible with large forecast horizons in fast dynamic systems.

One effective way to avoid recursion of the prediction model is to apply the Multiple Shooting method [126]. This consists of including the sequence of states inside the prediction window $x_{k+1,k+N-1}$ as decision variables of the procedure of optimization and add constraints that ensure that the dynamics in Eq. (5-7) are respected. The procedure can be seen as the search for the zeros of the equations that depend on the states in the prediction window that satisfy the dynamics equations. Thus, the cost function now includes the states of the system as a decision variable inside the prediction window, that is,

$$J_N(u_{k,k+N-1}, x_{k+1,k+N-1}) = \sum_{i=k}^{k+N-1} \| x_i - q_i \|_Q^2 + \sum_{i=k}^{k+N-1} \| u_i - r_i \|_R^2 + + \sum_{i=k+1}^{k+N-1} \| \Delta u_i \|_S^2 , \quad (5\text{-}7)$$

PUC-Rio - Certificação Digital Nº 1912774/CA

and the optimization problem is reformulated in order to guarantee the necessary recursion to the predictions through constraints that depend on Eq. (5-6). In terms of defining optimization problem, changing the function implies attracting constraints, such as

$$\begin{aligned} \min \quad & J_N(u_{k,k+N-1}, x_{k+1,k+N-1}) , \\ \text{s.t.} \quad & u_i \in U , \\ & x_i \in X , \\ & u_{min} \leq u_i \leq u_{max} , \\ & x_{min} \leq x_i \leq x_{max} , \\ & x_{i+1} = f^d(x_i, u_i) , \\ \text{for} \quad & i = k, k+1, ..., k+N-1 . \end{aligned} \quad (5\text{-}8)$$

Despite having more decision variables, the procedure is solved more efficiently because it avoids the composition of functions necessary to make the predictions. This presents advantages in cases where the prediction window is large, as optimizing $J_N(.)$ is significantly simpler despite the more significant number of decision variables that this implies. For path-tracking, the basic scheme regarding the MPC control can be seen in Fig. 5.2.

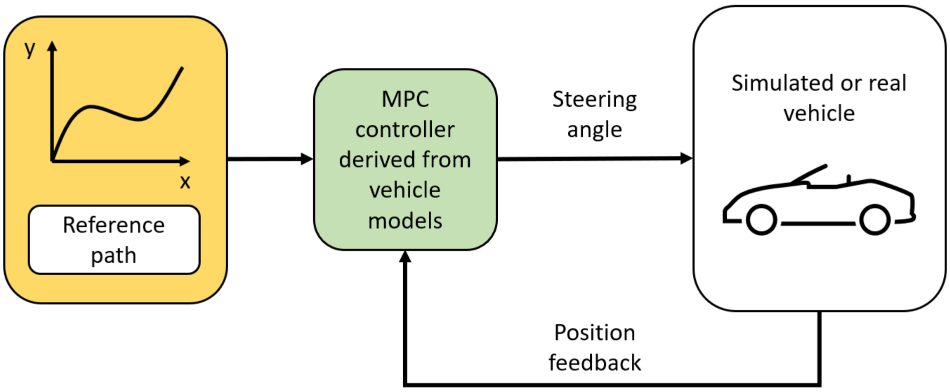

Figure 5.2: Schematic overview of MPC control applied to the path-tracking of an AGV. It is required a model of the process to predict its likely evolution and choose the “best” control action.

The resulting NLP is enabled to be solved by state-of-the-art numerical optimization algorithms, such as the Interior Point [124], and Sequential Quadratic Programming [63]. Through this thesis, a modification of the interior-point method (for details, see [124]) is used to solve NLPs together with the open-source application CasAdi [125] widely used for numerical optimization due to the facility to work with symbolic expressions.

Example 5: *The present example demonstrates the application of an MPC controller.*

The inverted pendulum problem detailed in Section A.5 is used to apply MPC formulations. The motion equations can be rewrite by defining $x =[\theta \ p \ \dot{\theta} \ \dot{p}]$ and $u = F$. Therefore, this example gives the optimization problem by Eq. (5-3) considering the states x and the control input u. The optimization problem is solved using the interior-point method by means of the Ipopt software package [124] with the open-source tool - CasAdi [125]. On the other hand, the ODE (Ordinary Differential Equation) system is solved with CVodes from the SUNDIALS Suite [127], which provides an accurate and fast solution for nonlinear systems. The procedure is implemented in MATLAB®under a laptop with Windows 10 OS endowed with an Intel i5-7300HQ CPU and 16 GB RAM.

For the predictive controller, the horizon N is set to 20, and the simulation sampling interval is set to 0.1 seconds. The weighting matrices are shown in Eq. (5-9), giving a more significant penalty for θ.

PUC-Rio - Certificação Digital Nº 1912774/CA

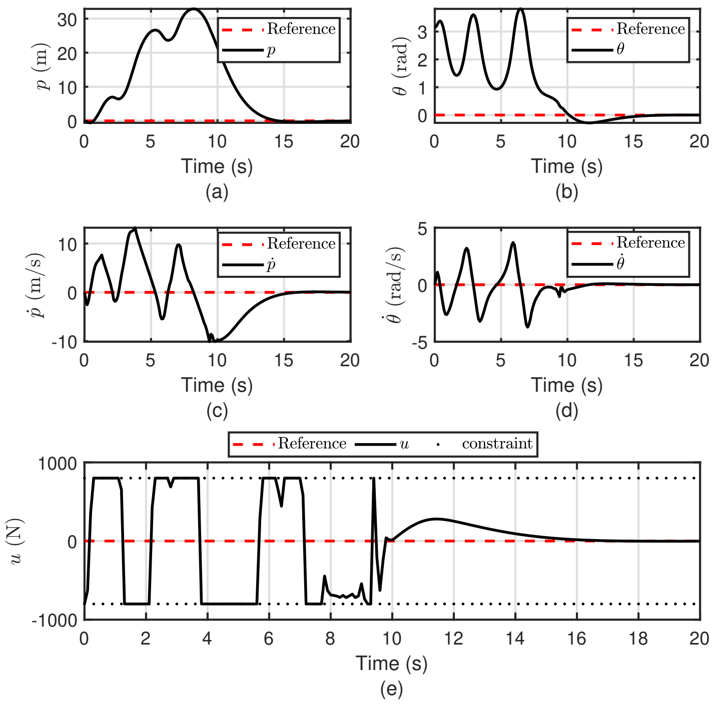

Figure 5.3: Example 5: Results for the MPC applied to the inverted pendulum. (a) Position. (b) Velocity. (c) Angle and (d) angular rate of the system. (e) Force applied to the system submitted to constraints.

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; S = \begin{bmatrix} 0 \end{bmatrix}. \tag{5-9}$$

The reference signals for all the states are the stabilization around the zero value. Finally, constraints of -800 <u< 800 [N] are applied to the input variable, and the initial conditions are set as x =[0 π 0 0].

Fig. 5.3 shows the MPC procedure applied to the inverted pendulum system. MPC results achieved good tracking performance with the reference signals followed through the simulation. In addition, the input variable (force) achieved the constraints satisfactorily, showing one of the main advantages of this type of control.

5.3 Summary

Although substantial work has been done developing control techniques for the path-tracking of AGVs, much of this effort has focused on MPC

PUC-Rio - Certificação Digital Nº 1912774/CA

controllers, which can be ascribed to numerous reasons. First, the MPC allows the implementation of constraints, leading to performance gains in closed-loop system operation. Next, the MPC tuning effort is low since it is relatively easy to customize the cost function and its weights. Then, the MPC is considered a robust controller since an optimal solution is found for each time step. Hence it may present satisfactory results in the presence of disturbances. Finally, the MPC controller is derived from a model of the process.

The present chapter dealt with the formulation of the MPC controller, procedures to solve the NLP, and the steps to implement the MPC computationally. Finally, an example of the methodology was presented to control an inverted pendulum. In this work, the focus of the first two original contributions (Chapters 6 and 7) relies on 1) the use of machine learning for tire learning to be used together with the nonlinear vehicle model and 2) the use of a combination of data-driven vehicle models, both to derive control laws to efficiently improve MPC approach in terms of accuracy and computational effort. The proposed approaches are designed to be compared to the LMPC and NMPC widely used in the literature showing the effectiveness, robustness, and adaptability of the MPC controller.

PUC-Rio - Certificação Digital Nº 1912774/CA

Part III

Contributions

PUC-Rio - Certificação Digital Nº 1912774/CA

6
Nonlinear tire model approximation using machine learning for efficient model predictive control

Model Predictive Controller is widely used as a technique for path-tracking control since it allows for dealing with system constraints and future forecasts. However, the performance of MPC is directly affected by the adopted model. A complex dynamic model can guarantee accuracy in path-tracking but may not be suitable in computational terms. On the other hand, a simplified model may not capture essential nonlinear aspects. Thus, to cope with these problems, the present contribution deals with data-driven tire modeling to improve vehicle path-tracking control. The main contribution is to show that neural tires can capture the nonlinearities in the interaction between lateral and longitudinal vehicle dynamics with a reduced computational cost for predictive controllers. Simulated and experimental tire data are approximated to design data-driven tire models using radial basis function and multilayer perceptron neural networks. Then, based on ground vehicles with neural tires, model predictive controllers are designed to regulate wheel torque and steering angle inputs. Tests were conducted to compare the proposed data-driven MPC approach with the classical nonlinear MPC controller. The results show that the neural tires approximate nonlinear tire models and experimental data with arbitrary precision in terms of accuracy and error-based metrics. The proposed methodology was successfully applied to perform trajectory and velocity tracking of ground vehicles. In addition, MPC with a neural tire model as prediction inference reduces the computational effort compared to traditional approaches.

PUC-Rio - Certificação Digital Nº 1912774/CA

6.1
Proposed approach

The focus of the present section is on stating the proposed approach based on data-driven models for tire modeling and MPC with ANN tire model to perform path-tracking of ground vehicles.

6.1.1
Data-driven tire modeling and optimization problem

The accuracy of the data-driven approach is relevant since the experimental data may be non-viable to measure. Moreover, experimental tests are costly to prepare and may demand work days. On the other hand, simulated tests do not require significant computational time and are adaptable for different case studies. In the present contribution, simulated and experimental data establish the relation between friction (output) and slip (input) in longitudinal and lateral directions. Then, the data-driven approximation problem is decomposed into parameter estimation to obtain data-driven models. Being so, after setting the number of neurons and hidden layers (when required), each architecture is designed to have a vector H representing MLP (weights and biases) and RBF (centers, widths, and weights) model parameters. These parameters are determined by the optimization algorithm presented in Eq. (6-1):

$$\min \quad J_p = \frac{1}{N_l} \textstyle\sum_{n=1}^{N_l} \xi_n \xi_n \,, \tag{6-1}$$

where $\xi(t) = y(t) - \hat{y}(t)$ i.e. the prediction error. Moreover, N_l indicates the amount of data used. Besides, one has to specify the limits of the search of the network parameter according to the problem.

Experimental data can also be used to determine the parameters present in the Pacejka equation. The procedure is based on the same optimization problem presented in Eq. (6-1), but with the view to obtain coefficients of the well-known equation.

The optimization problems for RBF and MLP models and the Pacejka curve with experimental data are solved using the Ipopt software package [124] with the open-source software - CasAdi.

First, data-driven tire models are designed based on the number of neurons and the dataset through an optimization problem (Fig. 6.1). Then, for path-tracking, at each time step, longitudinal and lateral slippage (input) are calculated using the dynamic model (Eqs. (3-1)-(3-8)) and sent to the ANN tire models to produce the value of friction coefficient (output).

6.1.2
MPC with ANN as tire model prediction

Predictive controllers are herein designed to perform path and velocity tracking. Dynamic vehicle models with data-driven tire modeling are used for MPC prediction of optimized control actions. The optimized control actions, steering, and torque on the driven wheels are first simulated on a virtual plant with simulated tire data (Section 6.2) and then on a virtual plant with

PUC-Rio - Certificação Digital Nº 1912774/CA

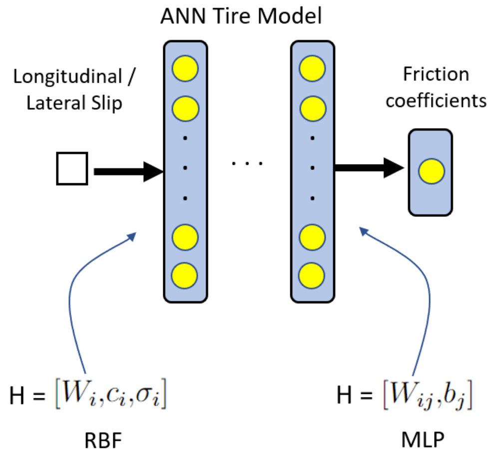

Figure 6.1: Neural networks architectures are designed to approximate friction coefficient curves using slip (input) and friction (output). Each architecture has a vector H representing model parameters to be determined by optimization problems. In particular, for MLP (weights and biases) and RBF (centers, widths, and weights) networks.

experimental tire data (Section 6.3). In the latter, look-up tables represent the experimental tire data.

The proposed control methodology aims to improve computational efficiency and verify the application of neural tire models with MPC to derive nonlinear control laws. The optimization problem is also solved using the Ipopt software package [124] with the open-source tool - CasAdi [125]. The ODE system is solved with CVodes from the SUNDIALS Suite [127]. The procedure with multiple shooting is applied considering the states $\mathbf{x} = [\dot{x}, \dot{y}, \dot{\psi}, \omega_f, \omega_r, X, Y, \psi]$, and inputs $\mathbf{u} = [\delta, T_f/T_r]$. However, it is important to point out that the MPC cost function (Eq. (6-2)) considers only the following states $\underline{\mathbf{x}} = [\dot{x}, X, Y]$.

$$J_N(\mathbf{u}_{k,k+N-1}, \underline{\mathbf{x}}_{k+1,k+N-1}) = \sum_{i=k}^{k+N-1} \parallel \hat{\underline{\mathbf{x}}}_i - \underline{\mathbf{x}}_i \parallel_Q^2 + \sum_{i=k+1}^{k+N-1} \parallel \Delta\mathbf{u}_i \parallel_S^2 \qquad (6\text{-}2)$$

where $\hat{\underline{\mathbf{x}}}$ represents the predicted states, $\underline{\mathbf{x}}$ represents the references given to the vehicle, Q is a weight that gives importance to the states to be followed by the AGV, and S is a weight that gives importance to the inputs (steering and torque). Moreover, the weights matrices are defined by trial and error. Finally, the proposed approach is summarized in Fig. 6.2.

In this contribution, the vehicle parameters are assumed to be known. However, the parameters needed to represent the vehicle for the proposed MPC can be measured or estimated with accuracy based on tests with simulated and real data. Recent works ([35, 37, 128]) have explored this topic by employing identification approaches to solve this issue in aspects of vehicle suspension

PUC-Rio - Certificação Digital Nº 1912774/CA

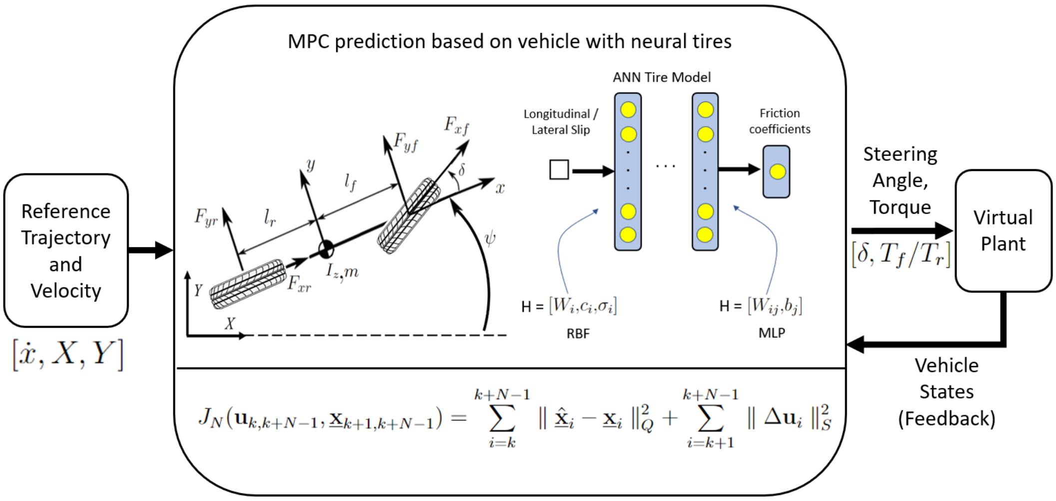

Figure 6.2: General overview of the proposed approach: Once the neural network architectures are obtained, the neural tire model is linked to the dynamic vehicle model to derive optimized control actions in the MPC controller. Finally, the control actions are sent to the virtual plant with simulated/experimental tires.

systems and lateral-longitudinal dynamics. Finally, neural tire model creation and path-tracking predictive control are implemented in MATLAB®under a laptop with Windows 10 OS endowed with an Intel i5-7300HQ CPU and 16 GB RAM.

PUC-Rio - Certificação Digital Nº 1912774/CA

6.2 Trajectory tracking results with simulated tire data

This section presents the data-driven modeling results from simulated tire data acquired from the Pacejka tire model. Moreover, an MPC controller is designed based on a vehicle with neural tires to control a virtual plant with Pacejka's tires. The results are presented in Subsections 6.2.1 and 6.2.2.

6.2.1 Neural tire model creation

In this section, neural tires are compared with the Pacejka friction curves. Simulated data from the Pacejka formula were developed with 200 data points considering a vehicle (with weight distribution 50/50) and tire parameters [129] listed in Tables 6.1 and 6.2, respectively. Besides, for RBF and MLP neural networks, the limits of search of the network coefficients are summarized in Table 6.3 after some trial and error.

Different RBF and MLP networks are simulated, modifying the number of neurons for both approaches and the number of hidden layers for the latter. In addition, this contribution considers two error-based metrics: Root Mean

Table 6.1: Vehicle parameters - simulated case.

Symbol	Definition	Value
m	Vehicle mass	2,500 kg
I_z	Inertia moment about yaw axis	2,200 kg.m^2
I_{ω_f/ω_r}	Inertial moment of the wheel	2.5 kg.m^2
l_t	Wheelbase	2.7 m
r_d	Radius of the tire	0.42 m
i_{sw}	Steering transmission ratio	30

Table 6.2: Coefficients from Pacejka formula - simulated case.

μ_x		μ_y	
Symbol	Value	Symbol	Value
B	0.208	B	0.154
C	1.650	C	1.300
D	6,213.4	D	5,367.9
E	0.604	E	-1.464
S_h	0.0	S_h	0.0
S_v	0.0	S_v	0.0

Table 6.3: Limits of search of the neural networks coefficients.

Network	Symbol	Definition	Lower limit	Upper limit
RBF	c	center	-1	1
	σ	width	0.01	7
	W	weights	-80	80
MLP	b	bias parameters	-4	4
	W	weights	-4	4

PUC-Rio - Certificação Digital Nº 1912774/CA

Squared Error - RMSE and the Multiple Correlation Coefficient (R^2), to compare the numerical results.

Table 6.4 depicts the metric results for longitudinal and lateral friction coefficients. The number of hidden layers inside the MLP network is set from 1 ({X}) to 3 ({X X X}), and the number of neurons inside these layers is set from 1 to 3 (X=1,2,...3). In turn, the number of neurons inside the RBF networks is set from 1 to 5 (X=1,2,...5). These quantities of the number of neurons and layers were selected aiming a reduced computational cost which leads to a less complex architecture to be implemented as a data-driven tire model.

The results achieved by the neural networks were satisfactory, specifically above one neuron, with R^2 metric ranging from 0.9587 to 1.000 and 0.9921 to 1.000 for longitudinal and lateral friction coefficients, respectively. On the other hand, the RMSE metric ranges from 0.1628 to 0.0004 and 0.0711 to 0.0022 for

Table 6.4: Results in terms of R^2 and RMSE metrics, and solver average computational time for all the architectures tested considering simulated tire data.

Model {Architecture}	μ_x R^2	μ_x RMSE	μ_y R^2	μ_y RMSE	Average time (s)
RBF {1}	0.2248	0.7055	0.2546	0.6922	0.0190
RBF {2}	0.9741	0.1289	0.9977	0.0382	0.1780
RBF {3}	0.9978	0.0374	1.000	0.0041	0.7405
RBF {4}	0.9998	0.0117	1.000	0.0022	0.6585
RBF {5}	0.9784	0.1179	0.9954	0.0687	0.8540
MLP {1 1}	0.9587	0.1628	0.9921	0.0711	0.0250
MLP {2 2}	0.9997	0.0155	1.000	0.0043	0.2530
MLP {3 3}	1.000	0.0011	1.000	0.0024	0.3305
MLP {1 1 1}	0.9832	0.1040	0.9988	0.0276	0.1655
MLP {2 2 2}	1.000	0.0004	1.000	0.0028	0.2865
MLP {3 3 3}	1.000	0.0004	1.000	0.0029	0.6260

longitudinal and lateral friction coefficients. Considering RBFNs, the RBF with four neurons (RBF {4}) presented the best result for both friction coefficients with an average simulation time (measured considering longitudinal and lateral friction components) of 0.6585 s. Considering the MLP, the MLP architecture composed of three hidden layers and three neurons in each layer (MLP {2 2 2}) presented the best results. However, this implies a more complex optimization calculation of network parameters, and therefore, the simulation time increases. Thus, considering both the simulation time and the accuracy of the error-based metrics, the MLP network with two hidden layers with two neurons in each layer (MLP {2 2}) presented the best result.

From the Figs. 6.3.a-b, one can see the longitudinal and lateral friction coefficient curves obtained from the best RBF and MLP neural network architectures. In particular, RBF {4} and MLP {2 2} neural networks. From the figures, it can be seen that the curves of MLP and RBF architectures are close to the reference tire model.

6.2.2 Path-tracking controller

This section compares the results of the MPC controller, based on a front-wheel-drive vehicle with neural tires, with those performed by an MPC controller designed based on a vehicle with Pacejka's tire model. Here, the vehicle model with neural tires (MLP {2 2}) is used to predict the future states of the vehicle over a finite horizon to perform path-tracking using optimized control actions. The effectiveness of the proposed control approach is verified

PUC-Rio - Certificação Digital Nº 1912774/CA

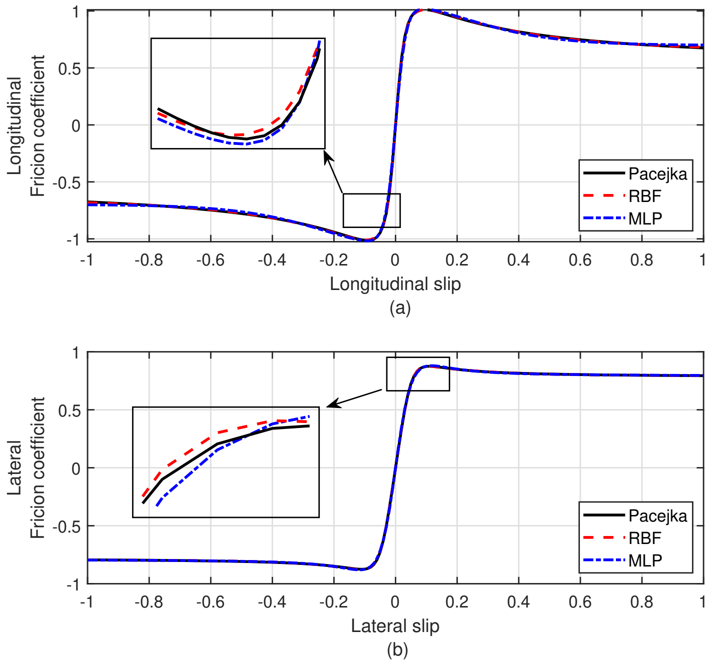

Figure 6.3: (a) Longitudinal friction approximation using MLP and RBF networks. (b) Lateral friction approximation using MLP and RBF networks.

PUC-Rio - Certificação Digital Nº 1912774/CA

by considering simulated tests on a plant with simulated tire data (Table 6.2) under double lane change and consecutive lane changes.

For the predictive controllers, the horizon N is set from 5 to 15, and the simulation sampling interval is set to 0.1 seconds. The weighting matrices are shown in Eq. (6-3). Both weights are set equal for both MPC approaches after some trial and error.

$$Q = \begin{bmatrix} 1,500 & 0 & 0 \\ 0 & 200 & 0 \\ 0 & 0 & 2,500 \end{bmatrix}; S = \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix} \times 10^3 \,. \tag{6-3}$$

Finally, constraints of -0.9 $< \delta <$ 0.9 [rad] and -2,500 $< T <$ 2,500 [N.m] are applied on the steering angle and the front-driven wheels representing typical vehicle operation zones. Moreover, an initial velocity of 19 m/s is given to the vehicle.

6.2.2.1
Double lane change

In the simulation test, the vehicle is motivated to perform a double lane change trajectory with a constant referenced speed of 20 m/s from an initial velocity of 19 m/s. The tracking performance and the computational effort can be seen in Table 6.5.

Table 6.5: RMS and maximum error under different lengths of horizon and control strategies for double lane change simulation.

MPC approach	Horizon length	RMSE $[Y]$ (m)	$\|Y_{max}\|$ (m)	Average solver time (s)
Pacejka	5	0.2029	0.3959	0.8437
	10	0.1989	0.3872	1.5745
	15	0.2019	0.3925	2.5170
Neural	5	0.2014	0.3938	0.8264
	10	0.1972	0.3850	1.6291
	15	0.1997	0.3894	2.5728

The results indicated that both MPC approaches achieved good path-tracking performance for all cases. However, the MPC-Neural case presented the smaller lateral offset and the smaller RMSE metric regarding the lateral position, respectively, 0.3850 and 0.1972 m. It is interesting to note that, as expected, the simulation time increases with the horizon. However, the lateral error remains similar throughout the simulations. Overall, the maximum computational effort go to MPC approaches with a horizon of 15. From this point of view, a horizon length of 5 has an advantage in terms of precision and simulation time.

In particular, from Fig. 6.4.a, one can see that both control methods (MPC-Neural and MPC-Pacejka with Horizon set to 5) conducted the vehicle to track the double lane change trajectory satisfactorily. Fig. 6.4.c demonstrates that the offset error in the lateral direction achieved a maximum of 0.396 m for both cases, approximately. From Fig. 6.4.b, the vehicle starts with an initial velocity of 19 m/s until it reaches the reference velocity of 20 m/s. Moreover, the velocity state remains close to the reference through the simulation, as shown in Fig. 6.4.d.

From Figs. 6.5.a-b, the MPC controllers provided similar responses, with smooth control inputs to the steering wheel and driven wheels, giving better control and stability on curves. Besides, Figs. 6.5.c-d demonstrate the evolution of the lateral and longitudinal friction coefficients, respectively. Both the controllers achieved low values for friction during the simulation in the straight direction. However, lateral slips increase during the maneuver. Table 6.5 also presents the computational effort for the double lane change maneuver. MPC-Neural performed similarly as the MPC-Pacejka in terms of accuracy and computational time.

PUC-Rio - Certificação Digital Nº 1912774/CA

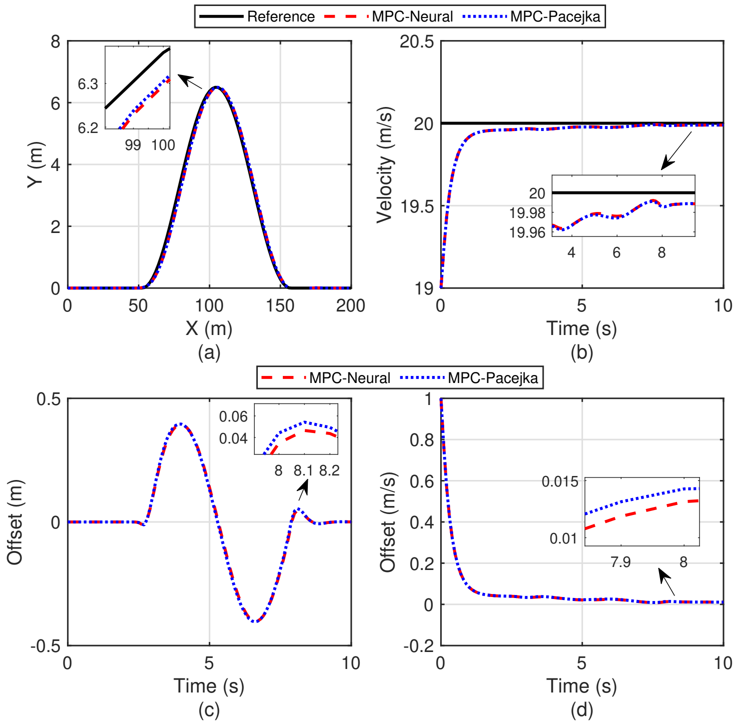

Figure 6.4: Results for double lane change simulation considering simulated data. (a) Trajectory. (b) Velocity. (c) Trajectory offset. (d) Velocity offset.

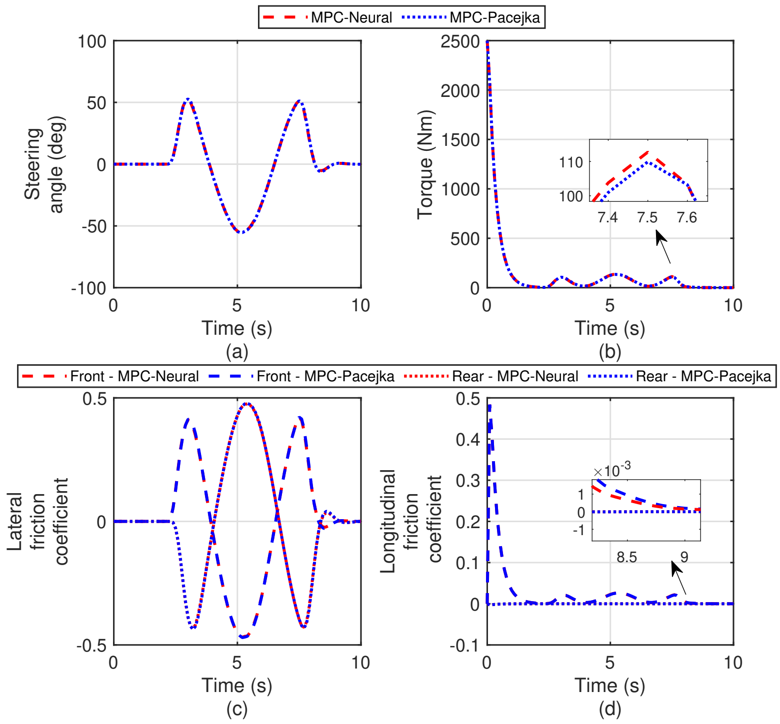

Figure 6.5: Results for double lane change simulation considering simulated data. (a) Steering wheel angle input. (b) Torque input. (c) Lateral friction coefficient. (d) Longitudinal friction coefficient.

PUC-Rio - Certificação Digital Nº 1912774/CA

6.2.2.2
Lane changes

Here, the vehicle is motivated to perform a sequence of lane changes with a constant and referenced speed of 20 m/s from an initial velocity of 19 m/s. The tracking performance and the computational efforts can be seen in Table 6.6.

Table 6.6: RMS and maximum error under different lengths of the horizon and control strategies for multiple lane changes simulation.

MPC approach	Horizon length	RMSE $[Y]$ (m)	$\lvert Y_{max}\rvert$ (m)	Average solver time (s)
	5	0.1728	0.3941	0.7887
Pacejka	10	0.1663	0.3850	1.5505
	15	0.1682	0.3910	2.3741
	5	0.1710	0.3910	0.8077
Neural	10	0.1647	0.3810	1.5303
	15	0.1661	0.3872	2.3223

PUC-Rio - Certificação Digital Nº 1912774/CA

MPC results from both approaches achieved good path-tracking with maximum error obtained during the double lane change maneuver. The RMSE decreased during the simulation, and a better convergence between reference and predicted data was performed. However, the MPC-Neural case presented a smaller lateral offset and smaller RMSE metric regarding the lateral position. Besides, the simulation time increased when the horizon was set to 15. On the other hand, when the predictive horizon is set to 5, both approaches converge to have precision and reduced computational effort.

Considering the application of the proposed MPC-Neural and the traditional MPC on the controllable plant, Figs. 6.6.a and 6.6.c show that both MPC-Neural and MPC-Pacejka achieved similar results, with an offset error of 0.395 m, approximately. From Figs. 6.6.b and 6.6.d, one can see that the vehicle velocity achieved the reference velocity with a minimum error.

The control inputs for the lane changes path can be seen in Figs. 6.7.a-b where both controllers achieved similar input curves. Moreover, Figs. 6.7.c-d show the lateral and longitudinal friction coefficients, respectively. High levels of longitudinal slip occurred on front-driven tires, while rear tires remained with low levels of slip. On the other hand, lateral slips increase under both tires during the maneuvers. Table 6.6 shows the average solver time considering lane changes maneuver. MPC-Neural also performed similarly to the traditional MPC-Pacejka in terms of computational time. This occurs mainly by the neural architecture (MLP {2 2} that requires an extra computational effort. Next, a more simplified architecture is used to represent real tire data.

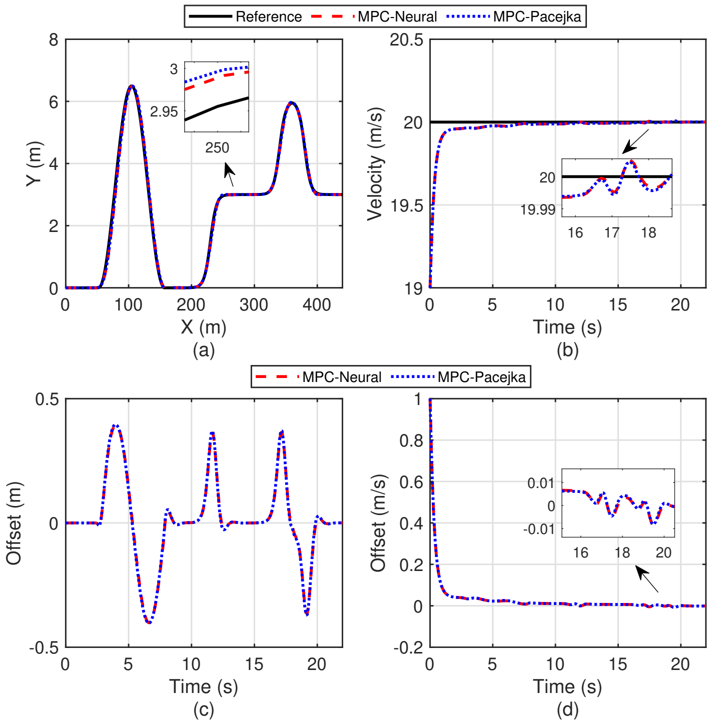

Figure 6.6: Results for lane changes simulation considering simulated data. (a) Trajectory. (b) Velocity. (c) Trajectory offset. (d) Velocity offset.

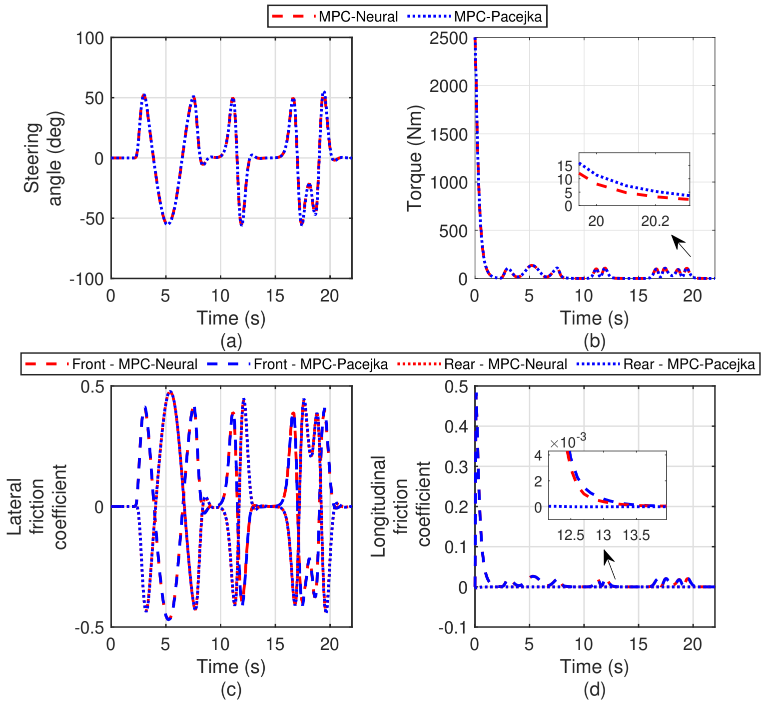

Figure 6.7: Results for lane changes simulation considering simulated data. (a) Steering input. (b) Torque input. (c) Lateral friction coefficient. (d) Longitudinal friction coefficient.

PUC-Rio - Certificação Digital Nº 1912774/CA

6.3 Trajectory tracking results with experimental tire data

This section presents the results obtained from the experimental data-based tire model. Besides, an MPC controller is designed based on a vehicle with neural tires to control a virtual plant with experimental tires. The results are presented in Subsections 6.3.1 and 6.3.2.

6.3.1 Neural tire model creation

Here, experimental data is used to derive a data-based tire model as an approximate model for tire modeling. To further validate the proposed Neural tire approach, experimental data [130] were obtained from tests performed by the Calspan Tire Research Facility as part of the Formula SAE Tire Test Consortium. The resulting test data are typical tire curves such as longitudinal and lateral forces versus slip/slip angle, respectively, and the aligning torque versus slip angle. For details regarding the tire tests and procedures, please refer to [131].

RBF, MLP architectures, and a fit by the Pacejka model are applied to approximate the experimental data. In particular, for longitudinal and lateral friction curves to be fit, the results consider approximately a vertical load of 660 N. The longitudinal and lateral friction curves contain 322 and 1268 data points, respectively.

For this case, the limits of the search of the network coefficients are the same presented in Table 6.3 for longitudinal and lateral friction curves. Besides, the same configuration regarding activation functions is used for the neural networks. For the Pacejka tire model, the limits of search of the coefficients (Table 6.7) are defined after some trial and error based on the reference [129]. Moreover, Table 6.8 indicates the optimized parameters to fit the experimental tire curves through the Pacejka tire model.

Table 6.7: Limits of search of the Magic formula coefficients.

Parameter	Definition	Lower limit	Upper limit
B	stiffness factor	-12	12
C	shape factor	-12	12
D	peak value	-12	12
E	curvature factor	-12	12
S_h	horizontal shift	-0.2	0.2
S_v	vertical shift	-1	1

PUC-Rio - Certificação Digital Nº 1912774/CA

Table 6.8: Coefficients from Pacejka formula - experimental case.

μ_x		μ_y	
Symbol	Value	Symbol	Value
B	10.3075	B	-10.8138
C	1.9157	C	-1.6192
D	2.6268	D	2.7166
E	0.5182	E	0.4118
S_h	0.0322	S_h	-0.0036
S_v	-0.2819	S_v	0.0694

The Neural and Pacejka tire model's friction curves are compared based on error metrics and average computational time. From the metric results presented in Table 6.9, one can see that the increase of neurons leads to more accurate results, demanding, however, an increase in computational effort. Considering both average computational time (to obtain longitudinal and lateral curves) and accuracy of the error-based metrics, MLP {1 1} and RBF {2} achieved the best results. However, MLP {1 1} is selected due to lower computational effort.

PUC-Rio - Certificação Digital Nº 1912774/CA

Table 6.9: Results in terms of R^2 and RMSE metrics, and solver average computational time for all the architectures tested considering experimental tire data.

Model {Architecture}	μ_x R^2	μ_x RMSE	μ_y R^2	μ_y RMSE	Average time (s)
RBF {1}	0.4403	1.6741	0.2978	1.9688	0.0255
RBF {2}	0.9953	0.1512	0.9988	0.0829	0.2310
RBF {3}	0.9968	0.1273	0.9992	0.0679	0.9600
RBF {4}	0.9968	0.1261	0.9992	0.0647	1.1300
RBF {5}	0.9919	0.2013	0.9992	0.0655	4.2050
MLP {1 1}	0.9957	0.1472	0.9991	0.0713	0.0415
MLP {2 2}	0.9968	0.1259	0.9993	0.0639	0.7920
MLP {3 3}	0.9969	0.1238	0.9993	0.0634	1.7550
MLP {1 1 1}	0.9957	0.1474	0.9991	0.0695	0.2670
MLP {2 2 2}	0.9968	0.1249	0.9993	0.0635	1.9750
MLP {3 3 3}	0.9970	0.1246	0.9993	0.0632	3.2150
Pacejka	0.9966	0.1313	0.9992	0.0671	0.2550

Fig. 6.8 demonstrates the longitudinal and lateral friction coefficient curves obtained from the best RBF and MLP architectures as well as the Pacejka model. One can see that the proposed method can capture the nonlinearities presented in tire data.

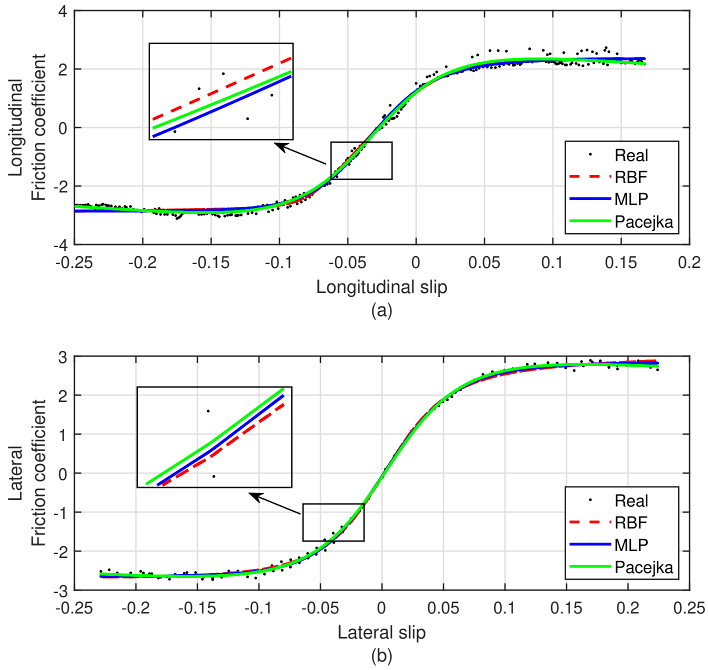

Figure 6.8: Comparison between experimental data and tire models considering (a) longitudinal and (b) lateral friction curves.

PUC-Rio - Certificação Digital Nº 1912774/CA

6.3.2 Path-tracking controller

In this part, the predictive control of a rear-wheel-drive vehicle with experimental data tire is implemented to corroborate the proposed neural tire approach's effectiveness. Two different trajectories are considered: double lane change and lane changes. The prediction is solved by the vehicle model with the neural tire, and the simulation is done considering the vehicle using experimental tire data. The latter uses look-up tables with slip and friction data as input and output.

For the predictive controllers, the horizon is set from 5 to 15, and the simulation sampling interval is set to 0.1 seconds. Also, the weighting matrices are set equal for both MPC approaches after some trial and error.

$$Q = \begin{bmatrix} 2,500 & 0 & 0 \\ 0 & 200 & 0 \\ 0 & 0 & 1,500 \end{bmatrix}; S = \begin{bmatrix} 1,000 & 0 \\ 0 & 1 \end{bmatrix} \times 10^3 . \qquad (6\text{-}4)$$

The vehicle (with weight distribution 50/50) parameters are listed in Table 6.10. In addition, constraints of -0.9 $< \delta <$ 0.9 [rad] and -1,200 $< T <$ 1,200 [N.m] are applied to the steering angle and the rear-driven wheels, respectively, representing typical Formula SAE vehicle operation zones. Finally, an initial velocity of 24 m/s is given to the vehicle.

Table 6.10: Vehicle parameters - experimental case.

Symbol	Definition	Value
m	Vehicle mass	270 kg
I_z	Inertia moment about yaw axis	120 kg.m^2
I_{ω_f/ω_r}	Inertial moment of the wheel	0.3 kg.m^2
l_t	Wheelbase	1.6 m
r_d	Radius of the tire	0.20 m
i_{sw}	Steering transmission ratio	10

6.3.2.1
Double lane change

The plant with experimental tire data is motivated to perform a double lane change maneuver with a constant and referenced speed of 25 m/s from an initial velocity of 24 m/s. The metric results in terms of lateral tracking error and computational efforts can be seen in Table 6.11.

Table 6.11: RMS and maximum error under different lengths of horizon and control strategies for double lane change considering experimental tire data.

MPC approach	Horizon length	RMSE $[Y]$ (m)	$\lvert Y_{max}\rvert$ (m)	Average solver time (s)
Pacejka	5	0.3157	0.6100	0.5588
	10	0.3119	0.5944	1.0134
	15	0.3125	0.5904	1.5668
Neural	5	0.3161	0.6139	0.4404
	10	0.3121	0.5986	0.8027
	15	0.3132	0.5955	1.1817

Metric results for a horizon of 5 presented the best computational performance compared to the horizon set to 10 and 15. It is interesting to note that MPC-Pacejka presented a similar performance in terms of error metrics compared to the MPC-Neural. However, in terms of computational time, MPC-Neural decreased the effort by 20% for horizons equal to 5 and 10, respectively, and 25% for a horizon equal to 15. The main reason is the simplified architecture of the MLP network (MLP {1 1}), which needs less computational effort.

From Fig. 6.9.a, one can see the desired trajectory and result of the proposed MPC-Neural. One also can note from Figs. 6.9.b-c, the followed

PUC-Rio - Certificação Digital Nº 1912774/CA

velocity and the lateral offset during the vehicle tracking. In particular, the maximum lateral offset achieved is 0.61 m.

Figs. 6.10.a-b illustrate the control inputs to the steering wheels and rear-driven wheels, respectively. The evolution of the lateral and longitudinal friction coefficients is shown in Figs. 6.10.c-d, respectively. The rear longitudinal friction is high initially due to the torque acting on the rear-driven wheels. However, as the vehicle moves, the longitudinal friction tends to decrease. On the other hand, lateral friction increases only when a steering angle is given to the wheels, as expected.

6.3.2.2
Lane changes

In this scenario, the vehicle is supposed to track a sequence of lane change maneuvers with a velocity of 25 m/s. The metric results in terms of lateral tracking error and computational efforts can be seen in Table 6.12.

Table 6.12: RMS and maximum error under different lengths of the horizon and control strategies for lane changes considering experimental tire data.

MPC approach	Horizon length	RMSE $[Y]$ (m)	$\lvert Y_{max} \rvert$ (m)	Average solver time (s)
Pacejka	5	0.2654	0.7317	0.5465
	10	0.2599	0.7052	0.9766
	15	0.2609	0.7085	1.6573
Neural	5	0.2661	0.7373	0.4382
	10	0.2603	0.7101	0.7757
	15	0.2612	0.7136	1.3855

The metric results show that RMSE and lateral offset are similar for both MPC approaches, approximately 0.71m. The main difference is that using MPC-Neural produced a computational effort reduction of 15% considering a horizon of 15 and 20% for horizon lengths of 5 and 10. Figs. 6.11 and 6.12 present the results for both approaches when the predictive horizon is set to 5.

Figs. 6.11.a-b show the trajectory and velocity tracking performance. The maximum lateral offset, approximately 0.71 m, can be seen in Fig. 6.11.c. Moreover, the reference velocity achieved a minimum offset in Fig. 6.11.d.

The control inputs are plotted in Figs. 6.12.a-b. It is observed that a high torque increases the rear longitudinal friction coefficient. However, as the simulation time elapses, the longitudinal friction tends to decrease, as seen in Fig. 6.12.d. Moreover, towed front wheels remain with a low level of longitudinal friction coefficient due to the absence of driven torque. On the

PUC-Rio - Certificação Digital Nº 1912774/CA

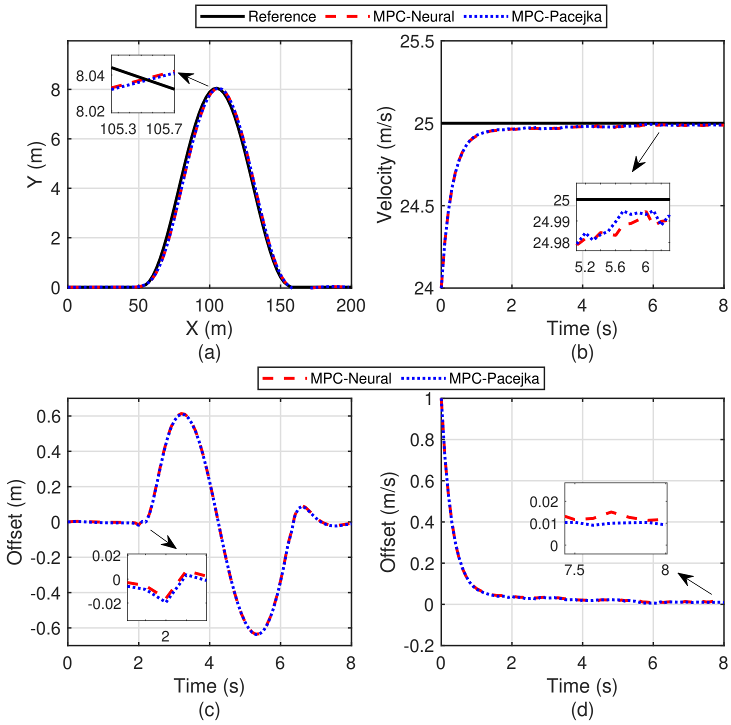

Figure 6.9: Results for double lane change simulation considering experimental tire data. (a) Trajectory. (b) Velocity. (c) Trajectory offset. (d) Velocity offset.

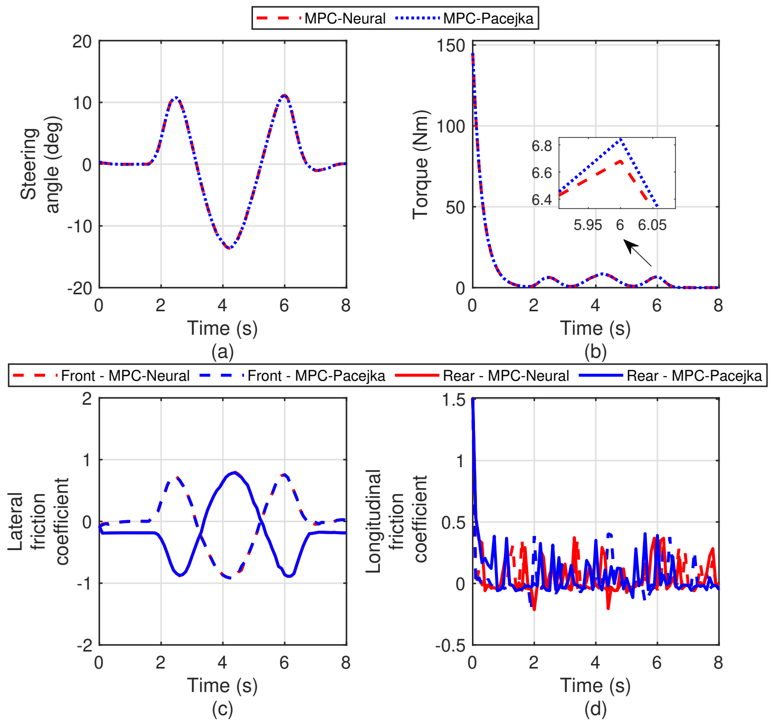

Figure 6.10: Results for double lane change simulation considering experimental tire data. (a) Steering input. (b) Torque input. (c) Lateral friction coefficient. (d) Longitudinal friction coefficient.

PUC-Rio - Certificação Digital Nº 1912774/CA

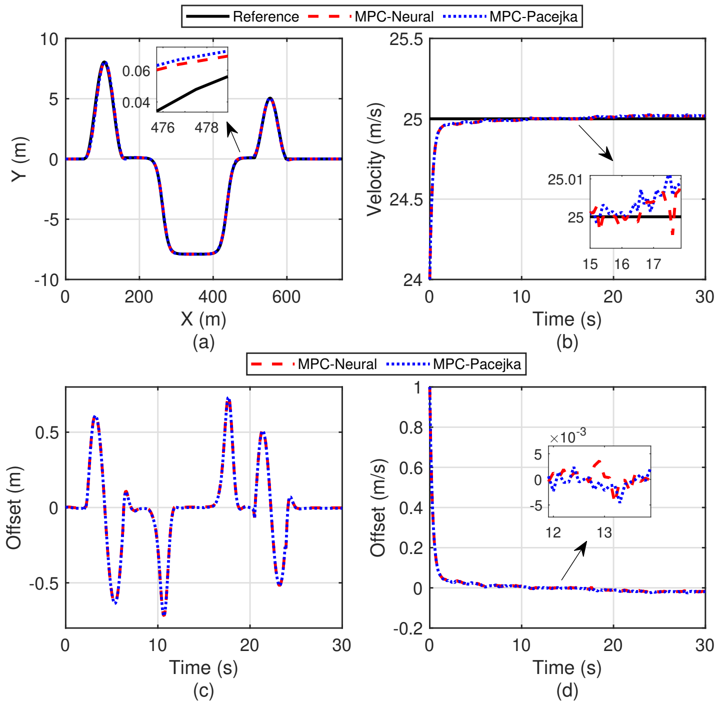

Figure 6.11: Results for lane changes simulation considering experimental tire data. (a) Trajectory. (b) Velocity. (c) Trajectory offset. (d) Velocity offset.

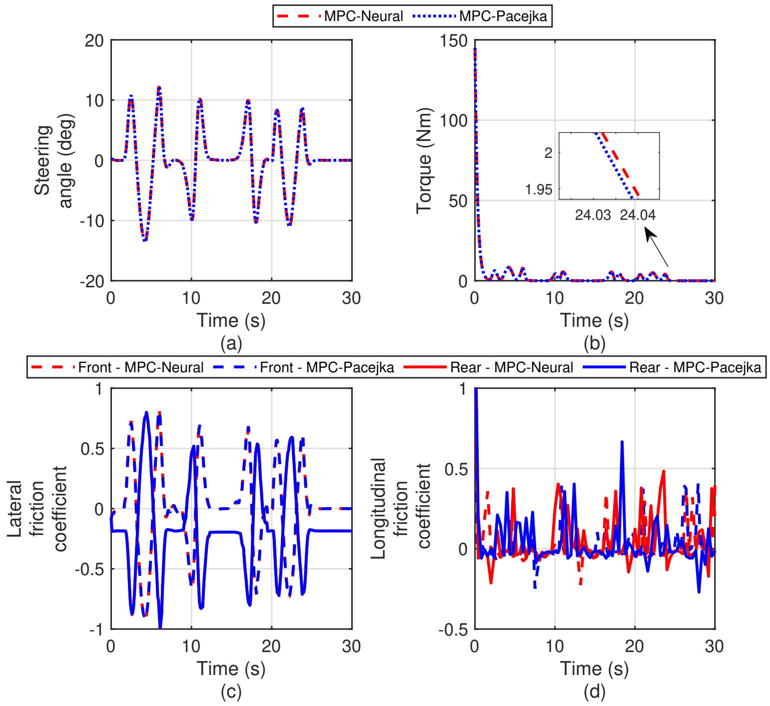

Figure 6.12: Results for lane changes simulation considering experimental tire data. (a) Steering input. (b) Torque input. (c) Lateral friction coefficient. (d) Longitudinal friction coefficient.

PUC-Rio - Certificação Digital Nº 1912774/CA

other hand, from Fig. 6.12.c, one can see that the lateral friction coefficient curve and the steering angle curve share the same trend, with the values increasing during the maneuver.

It can be seen from Fig. 6.13 that the total processing time for both maneuvers considering the horizon prediction set to 5. MPC-Neural performed even faster than the traditional MPC-Pacejka. The computational time decreases due to reducing the number of neurons in the data-driven tire model, in this case, MLP {1 1}.

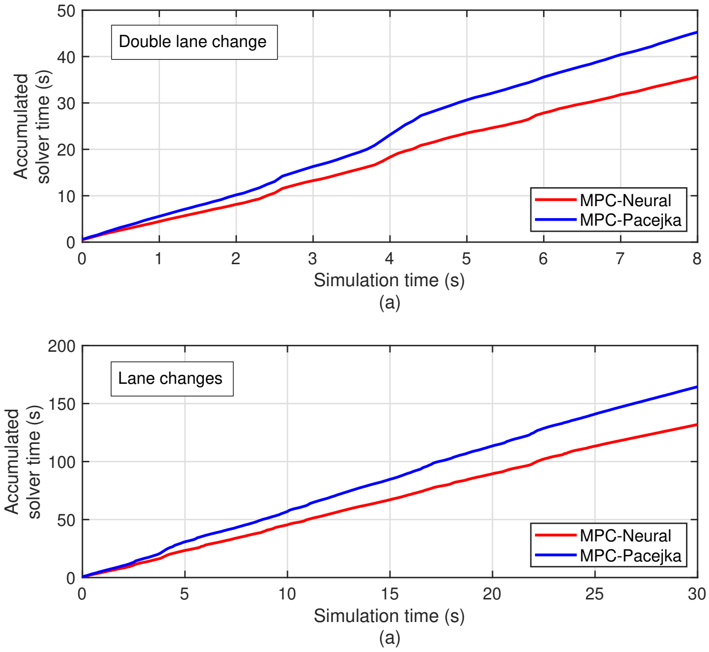

Figure 6.13: Total processing time through simulation with experimental tire data for (a) double lane-change and (b) lane changes.

PUC-Rio - Certificação Digital Nº 1912774/CA

6.4 Overall discussion and impacts

The simulated and experimental results presented in Section 6.2 and Section 6.3 give an insight into data-driven tire models applied to predictive control during trajectory tracking tasks. This approach allows for capturing nonlinear tire characteristics combined with a predictive control strategy during different maneuvers. As will be discussed next, the method presented herein gives advantages to current practice. It enables the construction of data-driven models for tires to arbitrary precision. Besides, the proposed methodology performs better computationally when used in prediction for MPC, which is essential in the scope of embedded solutions for control laws.

PUC-Rio - Certificação Digital Nº 1912774/CA

Simulated and experimental data were used to approximate tire curves, precisely longitudinal and lateral friction coefficients. The former data was obtained from the Pacejka model, and the latter was obtained from an experimental tire test. Data-driven models were derived using artificial neural networks, particularly MLP and RBF networks. In this case, Section 6.2.1 shows that the MLP {2 2} provided the best fit of data for both friction coefficients with reduced average computational time (considering both friction coefficients). Once the best architecture fits the simulated data, a predictive controller based on a vehicle with the selected neural tire is designed. A comparison with the exact vehicle using the Pacejka formula is shown in Section 6.2.2. The vehicle models with neural and Pacejka tires are used to predict the future states of the vehicle over the finite time window (MPC scheme). At the same time, the simulation occurs in a vehicle with simulated tire data. The results show that the predictions agree with trajectory, velocity, and control inputs (steering angle and torque). Besides, computational time reduction is observed using the predictive control with neural tire models. Thus, the data-driven models proposed herein are better for predictive models in MPC concerning computational use. Further improvements in computational time and trajectory tracking can be obtained using more complex models than the presented strategy. This approach is essential in embedded real-time applications and is crucial to optimize hardware use for MPC [132].

Regarding the results with experimental tire data, in Section 6.3.1, metric results and average simulation time to obtain friction coefficients indicate that the MLP {1 1} provided the best fit of experimental data. Then, an MPC controller based on a vehicle with neural tires is designed to control a vehicle with experimental tire data. One can see in Section 6.3.2 the good agreement between the reference trajectory and velocity throughout the simulations. The proposed methodology for neural-tire approximation with MPC herein presented gives good results even when real-world noise-corrupted data is used. The performance of the path-tracking controllers in terms of path-tracking is further verified by reported results in the literature (Table 6.13) regarding tracking control at high speed [27, 68, 133, 134].

Data-driven models give an alternative way to verify and analyze nonlinear aspects in tire modeling, which is lacking in existing tire models mainly because these models are well-suited for each application. Moreover, the results presented have implications considering data-driven modeling for MPC. The procedure grants computational time reduction, as observed in the tests performed. Furthermore, it showed that the proposed approach could provide good trajectory and velocity tracking under real-world scenarios when mea-

Table 6.13: RMS and maximum error under different control strategies and the proposed MPC approach.

Reference	Control strategy	RMSE $[Y]$ (m)	$\|Y_{max}\|$ (m)
[68]	MPC	-	0.40 - 0.50
[133]	MPC	0.20 - 0.60	0.76 - 1.76
[27]	Game theory-based	-	0.46 - 0.71
[134]	MPC	0.031 - 0.142	0.40 - 0.47
Proposed	MPC	0.15 - 0.32	0.36 - 0.73

sured data is applied. This approach indicates that, under high-speed driving situations, data-driven modeling for efficient predictive control can correctly replace physically-inspired derived models for tires. It is interesting to note, though, that the neural model is applied only to focus on the tire-road interaction, which is highly uncertain. Different models exist for each specific road condition [135, 136]. Such procedure is in line with most recent works for physically-inspired machine learning modeling [106].

Although the stability of the closed-loop system may be affected by the approximation error of the neural tire model, its behavior presents similarity when traditional approaches for tire curve approximation are also applied to closed-loop systems, as seen in [110]. Moreover, one of the shortcomings of the proposed approach is that the sudden change in the parameters may decrease the MPC performance. This problem could be solved by applying an observer/online training [18, 137]. However, the method presented herein builds on other physical models, increasing knowledge about the predictive modeling approach. In addition, this method helps to obtain a better overall data-driven model, as the machine learning models can focus on specific aspects of the system, that is, the tire-road interaction.

PUC-Rio - Certificação Digital Nº 1912774/CA

6.5 Summary

The trajectory tracking controller is an essential subsystem of autonomous vehicles that maintains the vehicle on an established trajectory. Several methodologies have been proposed throughout recent years. Predictive controller based on models has been widely used since it allows for dealing with constraints and uncertainties regarding the system. However, the effectiveness of the MPC approach relies on a reliable vehicle model used for future forecasts. Simplified dynamic models can ensure accuracy for specific situations with low computational cost; however, the MPC controller's performance can suffer due to nonlinear aspects. On the other hand, a nonlinear dynamic model may lead to an unsuitable real-time application of MPC. Thus, data-driven approaches are a potential solution with a satisfactory trade-off between accuracy and computational cost. Data-driven models are applied to approximate simulated and experimental tire data for efficient predictive control. The proposed approach has the potential to 1) approximate simulated and experimental tire data with arbitrary precision, reducing the effect of unmodelled aspects, 2) be used effectively with MPC to provide nonlinear control laws by providing accurate trajectory predictions, 3) be more efficient than traditional approaches in terms of computational time.

Here, Radial basis functions and Multilayer Perceptron architectures are compared by the different numbers of neurons, average simulation time, and error-based metrics. Results indicate that the MLP network performed better during the experiments than the RBF architecture for tire data prediction. In addition, the application of data-driven models with predictive control to provide nonlinear control laws was also established. Thus, MPC controllers based on dynamic vehicle models with neural tires are designed for path-tracking. Results indicate that using neural network architectures as prediction inference provided a computational time reduction of 10% to and 25% without accuracy losses. Results also indicate that the error regarding the MPC controllers in terms of lateral position is similar to that presented in recent literature with Y_{max} varying from 0.36 m to 0.73 m. It is noteworthy that the error is considered satisfactory, considering the vehicle's longitudinal velocity. Tests with a reduced velocity can decrease the trajectory error.

PUC-Rio - Certificação Digital Nº 1912774/CA

7
Lateral model identification using multi-ARX models for efficient model predictive control

Model Predictive Control proved effective for path-tracking control because it can use multiple constraints and variables with reasonable computational cost. The efficiency of this controller in terms of accuracy and computational efforts depends on the vehicle model applied to predict future states. This prediction model should be capable of capturing nonlinear aspects present in longitudinal and lateral vehicle dynamics without increasing computational costs. A potential solution is applying a data-driven approach that should represent an available dataset, not requiring high knowledge about the system. Thus, this contribution deals with data-driven vehicle modeling to efficiently enhance autonomous trajectory tracking control. A Multi-ARX-MPC (MARX-MPC) using different identified vehicle models is developed for this aim. A novel cost function weights the data-driven models to design the MPC control law. Comparative tests with the classical linear and nonlinear MPC controllers indicate that the data-driven approach can offer reliable results in terms of error-based metrics. Besides, the outcomes show that the proposed MARX-MPC can perform trajectory tracking on roads with continuous and noncontinuous reference positions for different speeds. Finally, compared with the conventional MPC controllers, the proposed architecture has decreased the computational effort by up to 88% with accuracy.

PUC-Rio - Certificação Digital Nº 1912774/CA

7.1
Proposed approach

This section focuses on the proposed MARX-MPC approach in which data-driven vehicle models are used to derive the MPC control law for the trajectory tracking of AGVs. First, the simulated testbed is presented, and then, the ARX approach to identify lateral vehicle dynamics is reported. Finally, the proposed MARX-MPC for trajectory tracking is discussed.

PUC-Rio - Certificação Digital Nº 1912774/CA

7.1.1
Simulated vehicle

A simulated testbed using MATLAB®/Simulink environment is developed to evaluate the performance of the proposed MARX-MPC controller. The simulated vehicle has 11 DoF, including three spatial DoF (longitudinal, lateral, and yaw) and four wheels with 2 DoF (longitudinal and lateral behavior of the wheels) each. Moreover, the vehicle model includes subsystems such as the steering wheel, engine, front-wheel driveline, suspension, and wheels with hydraulic brakes. The simulated vehicle is similar to that presented in [18]. The driver is modeled as a speed-tracking controller based on a Scheduled PI controller generating normalized acceleration and braking commands between 0 and 1. In addition, it is essential to point out that this virtual plant is sufficiently complex to be implemented with the MPC approach.

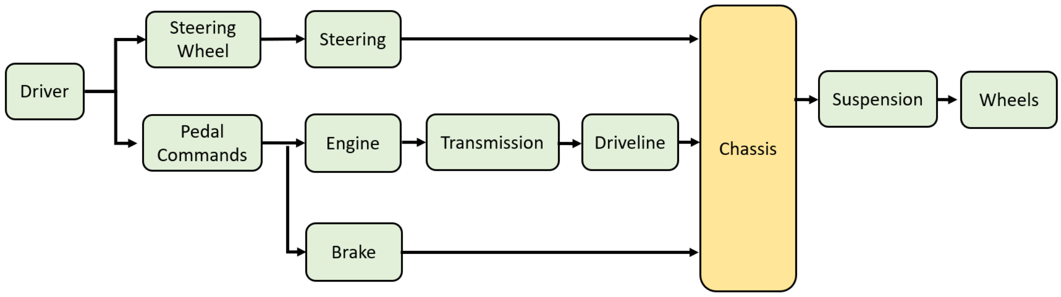

Figure 7.1: Complex AGV model used as controlled plant.

7.1.2
Dynamic vehicle model

Let some notation be recalled. Consider the single-track vehicle model presented by Eqs. (3-1)-(3-8) in Section 3.1. This contribution simplifies the vehicle model using only lateral dynamics with linear and nonlinear tire modeling using the Pacejka formulation. Thus, the modified vehicle model can be determined by the following

$$m\ddot{y} = 2F_{yf} + 2F_{yr} - mv_x\dot{\psi}\,, \tag{7-1a}$$

$$I_z\ddot{\psi} = 2l_fF_{yf} - 2l_rF_{yr}\,, \tag{7-1b}$$

$$\dot{X} = \dot{x}\cos\psi - \dot{y}\sin\psi\,, \tag{7-1c}$$

$$\dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi\,, \tag{7-1d}$$

where m and I_z are the mass and moment of inertia at the CG, respectively; l_f and l_r are the distance between the front and rear axles to CG. Moreover, F_{yf} and F_{yr} are the lateral forces on the front and rear wheels, respectively.

The forces can be modeled by the linear and nonlinear tire models presented in Sections 3.1.1 and 3.1.2. The vehicle motion can be described in the vehicle (x and y) and global frames (X and Y). In addition, ψ is the vehicle's and global frames' yaw angle.

7.1.3 Data-driven vehicle model

The accuracy of the identification procedure depends on the measured signals during data acquisition. Thus, a typical signal easily reproduced in a computer for lateral dynamics identification is the sinusoidal signal [138–140]. Sinusoidal signals are used as input signals to represent the vehicle's steering angle and lateral positions throughout time as output signals. Based on this, three different sinusoidal signals are selected considering different amplitudes, 0.15, 0.08, and 0.05 m, with the same frequency ($\pi/2$) according to the longitudinal vehicle velocity, 10, 15, and 20 m/s, respectively.

The simulated tests were performed using a sampling frequency of 100 Hz in a MATLAB®/Simulink environment using the complex model described in Section 7.1.1. The regressors matrix can be found using the approach presented by Eq. (4-9). The algorithm allowed the variation of na and nb parameters and the training and validation data percentage to find the best model based on error-based metrics (RMSE and R^2). For brevity, the focus is given only on the validation data set. In addition, the models are simulated in free-run simulation, in which actual input data is used to produce the predicted data. Thus, the measured data is used only in the initial condition. Hence, the FR simulation accumulates the prediction errors and shows the model validation more easily.

PUC-Rio - Certificação Digital Nº 1912774/CA

Table 7.1: Estimated coefficients and error-based metrics between actual and predicted signals, considering training and validation for FR for different longitudinal velocities.

	Coefficients				Error-based metrics			
Velocity	$\hat{a}_1$	$\hat{a}_2$	$\hat{b}_1$ ($\times 10^{-2}$)	$\hat{b}_2$ ($\times 10^{-2}$)	Training R^2	Training RMSE	Validation R^2	Validation RMSE
10	-2.0	1.0	-8.216	8.857	0.999	1.8532	0.9998	0.638
15	-2.0	1.0	-14.466	15.248	0.997	2.153	0.999	0.959
20	-2.0	1.0	-24.780	25.821	0.996	2.315	0.994	1.231

For the three identified models, the best results, in terms of error-based metrics for FR: RMSE and R^2 (Table 7.1), were detected when the training data corresponded to between 10-15% of the data set, and the validation set, in

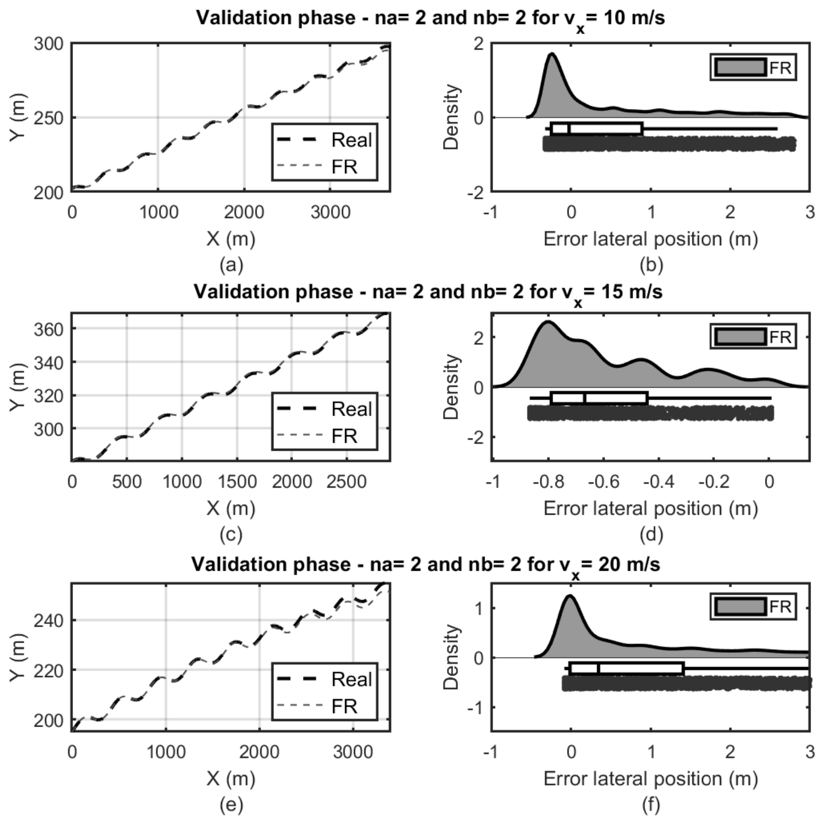

Figure 7.2: Validation results for each data-driven ARX model considering FR simulation for a) 10 m/s, b) 15 m/s, c) 20 m/s.

turn, corresponded to 85-90% of the entire set. These results justify high values for error metrics corresponding to training. Besides, for the three cases, the results indicate $na = nb = 2$. Table 7.1 also shows the estimated coefficients for the different sinusoidal signals used for identification.

Fig. 7.2 shows the performance comparison of these three vehicle models considering FR metrics for each longitudinal velocity (10, 15, and 20 m/s). It can be seen that the outputs from the identified models present similar behaviors to the measured lateral positions, considering the validation data. In addition, one can see the raincloud plots considering the error between predicted and real data. One can note that the identification presented by the models ARX-10 (ARX for the longitudinal velocity of 10 m/s) and ARX-20 can explain the data more accurately than ARX-15, as seen by the normal distribution with a mean of next to zero.

PUC-Rio - Certificação Digital Nº 1912774/CA

7.1.4
MARX-MPC for efficient trajectory tracking

Predictive controllers are herein designed to implement trajectory tracking. From the previous section, it can be seen that the estimated coefficients change as a function of the vehicle's velocity. However, a valid identified model for different velocities and vehicle behaviors is required. Therefore, it is proposed a weight cost function that weights each identified model based on the vehicle's velocity. The vehicle is driven at a speed between 10-20 m/s. Thus, to balance the results produced by identified models, it is also proposed a triangular supervisor function (shown in Fig. 7.3).

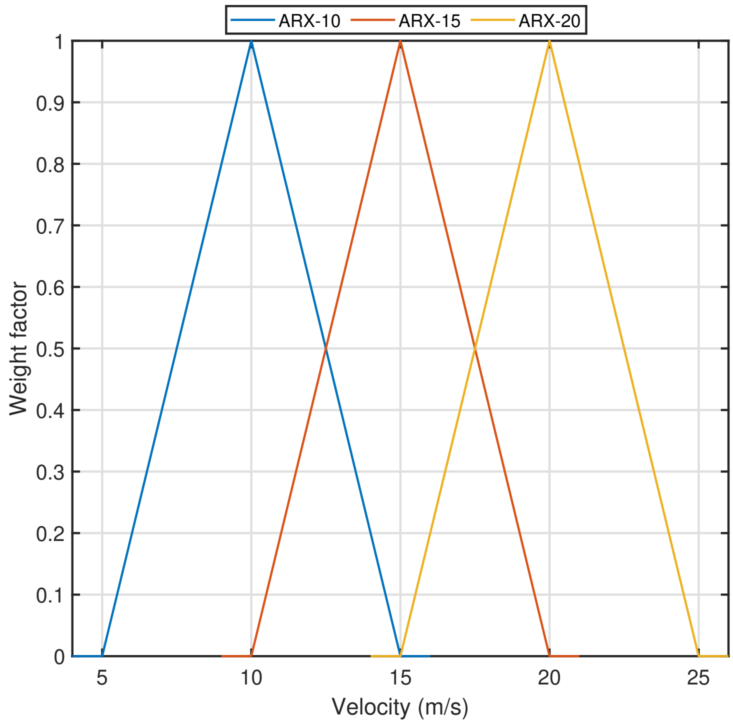

Figure 7.3: Weight factors according to longitudinal vehicle velocity.

Once the weight cost function is set, one may use it to derive the optimized control actions of the proposed MARX-MPC to control the complex model described in Section 7.1.1. The proposed MARX-MPC aims to enhance computational efficiency and verify the robustness of data-driven models to derive control laws. Only the lateral position and the steering angle are used from the controlled plant to derive the MPC cost function. Therefore, the cost function can be expressed as

$$J_N(\delta_{k,k+N-1}, y_{k+1,k+N-1}) = \sum_{i=k}^{k+N-1} \| \hat{y}_i - y_i \|_Q^2 + \sum_{i=k+1}^{k+N-1} \| \Delta\delta_i \|_S^2 \qquad (7\text{-}2)$$

PUC-Rio - Certificação Digital Nº 1912774/CA

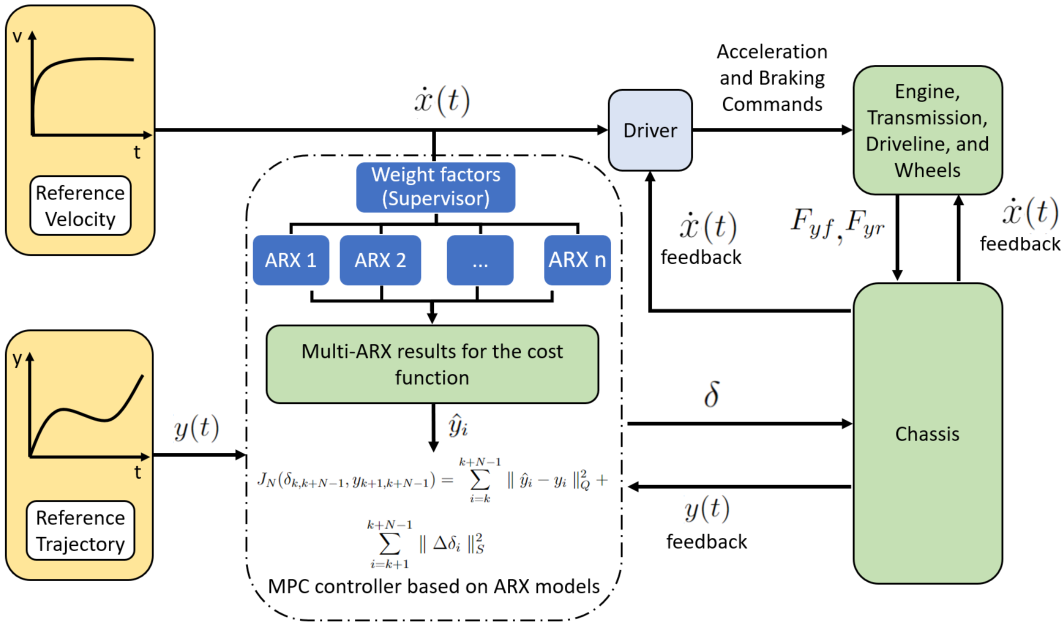

Figure 7.4: Proposed framework in which the MARX-MPC receives predicted and reference positions and vehicle's velocity to define the cost function. The steering angle is sent to the complex controlled plant, also composed of a velocity controller.

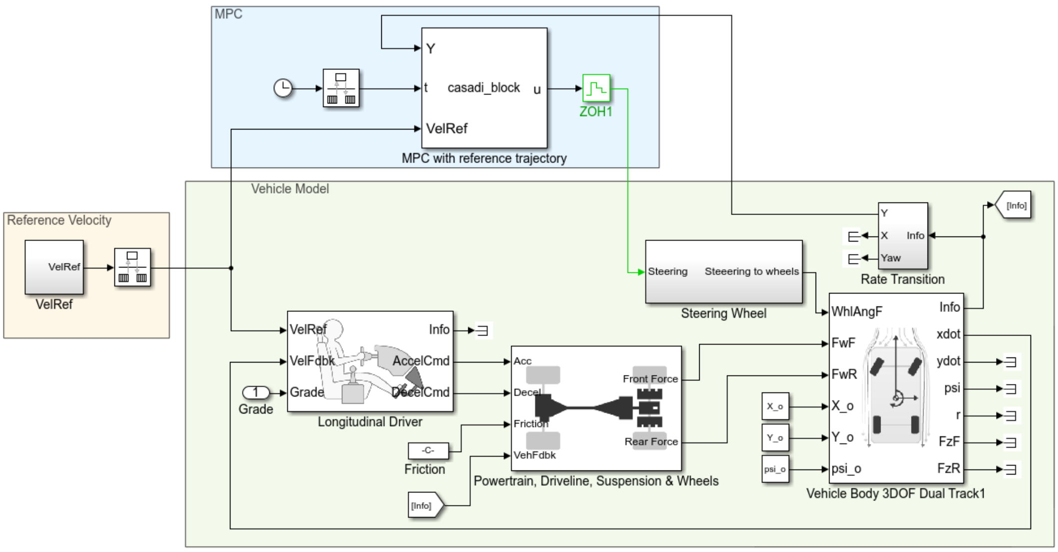

Figure 7.5: Proposed framework composed of vehicle model, MPC with reference trajectory, and reference velocity inside the Simulink platform.

where $\hat{y}$ is the predicted lateral position, y is the lateral reference position given to the vehicle, Q is a weight that gives the importance of lateral positions to be followed by the AGV, and S is a weight that gives importance for the input (steering).

Path-tracking control also needs to ensure comfort and safety handling for the vehicle's occupants. Then, in the second term of the cost function, a S parameter is applied regarding the steering angle. This parameter indicates

PUC-Rio - Certificação Digital Nº 1912774/CA

the importance of the control actions to ensure smooth vehicle handling. Specifically for the proposed MARX-MPC, the term that deals with the predicted lateral position in the cost function can be determined as follows

$$\hat{y}_i = w_1 v_x \hat{y}_i^{1'} + w_2 v_x \hat{y}_i^{2'} + \cdots + w_n v_x \hat{y}_i^{n'} \tag{7-3}$$

where the $\hat{y}_i$ is the global lateral position predicted at each interaction, $w_{1,2\cdots n}$ weighs the importance of the identified ARX models $\hat{y}^{1',2'\cdots n'}$ based on the vehicle's velocity. Although three ARX models are chosen, the proposed approach allows one to extend to a wider speed range if using additional models depending on the specific application. Finally, Fig. 7.4 shows a general overview of the proposed approach, and Fig. 7.5 presents the proposed approach inside the Simulink platform.

For the implementation of the MPC controllers, the prediction and control horizons are set with a fixed value of $N = 10$ since both aspects directly affect the computational methodology effort. Moreover, this allows a fair comparison between conventional MPCs and the proposed approach, excluding horizon and prediction length aspects. Tables 7.2 and 7.3 show the list of parameters regarding the simulated vehicle and the MPC controllers.

PUC-Rio - Certificação Digital Nº 1912774/CA

Table 7.2: Vehicle parameters.

Definition	Symbol	Value
Vehicle mass	m	2000 (kg)
Yaw moment of inertia	I_z	4000 (kg.m^2)
Distance from CG to the front axle	l_f	1.4 (m)
Distance from CG to the rear axle	l_r	1.6 (m)
Wheelbase	–	3.0 (m)
Track width	–	1.4 (m)

Table 7.3: Parameters of the MPC controllers.

Definition	Symbol	Value
Prediction horizon	N	10
Weight cost for lateral position	Q	100
Weight cost for steering angle	S	100
MPC step size	T_{MPC}	0.05 (s)
Simulation step size	T_s	0.001 (s)
Steering constraint	δ	[-20,20] (deg)

7.2 Results

The proposed approach is tested in a MATLAB®/Simulink simulation environment. The simulations are performed on a laptop with Windows 10

OS endowed with an Intel i5-7300HQ CPU and 16 GB RAM. Besides, the nonlinear programming (NLP) problems are solved using the Ipopt software package [124] with the open-source tool - CasAdi [125]. Linear and nonlinear systems are solved with CVodes from the SUNDIALS Suite [127].

7.2.1 ARX-MPC performance

The performance of the ARX–MPC based on each identified model according to the vehicle's velocity is initially evaluated. Different road profiles are considered since the trajectory has an essential effect on the vehicle's handling [141]. In this contribution, it is assumed that the trajectory planner provides a path with different curvatures to be traveled by the vehicle. Fig. 7.6 shows the vehicle's performance regarding trajectory tracking using the data-driven ARX-MPC for reference velocities of 10, 15, and 20 m/s. Moreover, the steering angle (control input) for each velocity is also shown in Fig. 7.6. One can see that the results are close to the reference trajectory, corroborated by the error metrics in Table 7.4. The upper and lower limits for the steering angle are restricted to [-20,20] deg to maintain comfort and safety. Then, as expected for predictive control, one can note that the restriction is respected during the trajectory tracking. Imposing actuation limits on the control is an advantage for model-based predictive control to maintain stability in previously defined satisfactory regions.

PUC-Rio - Certificação Digital Nº 1912774/CA

Table 7.4: Tracking distance errors for different ARX-MPC controllers.

	Controllers		
Metric error	ARX-MPC-10	ARX-MPC-15	ARX-MPC-20
R^2	0.9987	0.9985	0.9982
RMSE (m)	0.1560	0.1680	0.1795

Finally, a comparison of ARX-MPC solver time for the proposed approaches is shown in Fig. 7.7. Here, the solver time is computed at each time interval. The MPC solver time is similar among all three approaches, with a value close to 0.01 s in a normal distribution.

7.2.2 MARX-MPC performance

The performance of the proposed MARX–MPC approach is evaluated by driving the simulated vehicle to finish a segmented path. Therefore, a longer trajectory with different curvatures is designed to evaluate whether the vehicle can operate at low and high velocities with different magnitudes of steering

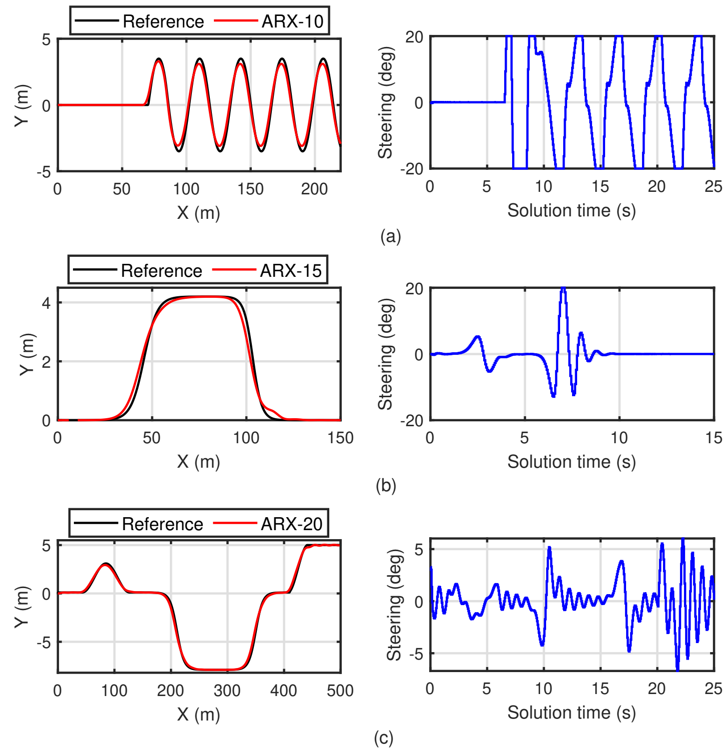

Figure 7.6: Trajectory tracking performance using ARX-MPC approach considering identified models concerning: a) 10 m/s, b) 15 m/s and c) 20 m/s.

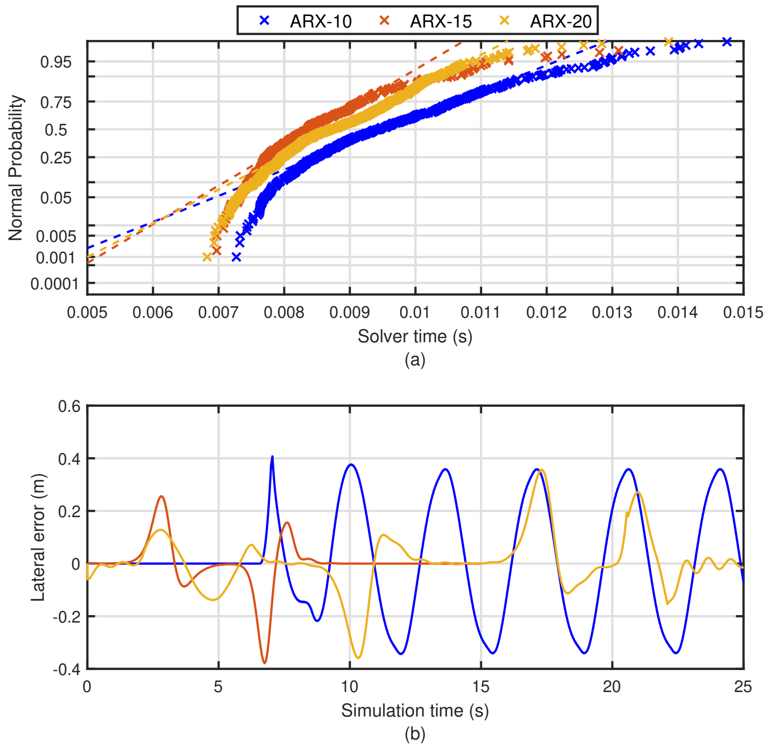

Figure 7.7: Solver time for each predictive control based on ARX models: ARX-MPC-10, ARX-MPC-15, and ARX-MPC-20.

PUC-Rio - Certificação Digital Nº 1912774/CA

angle. The chosen trajectory has three different segments, and each section is designed to have a specific curvature. Segment-1 has higher amplitudes at a lower frequency than segment-2. Segment-3, on the other hand, has the main characteristic of being a reference trajectory with abrupt trajectory change.

The proposed MARX-MPC approach is compared to the conventional MPCs based on dynamic models. Thus, two different predictive controllers are designed based on models presented in recent literature (see [28]). Here, predictive controllers based on a linear and nonlinear dynamic model with Pacejka tire modeling are referred to as "LMPC" and "NMPC", respectively. Fig. 7.8 shows the proposed architecture results compared to these conventional predictive controllers. In Fig.7.8.a, a comparison of the trajectory tracking performed by the conventional and proposed MPCs is presented.

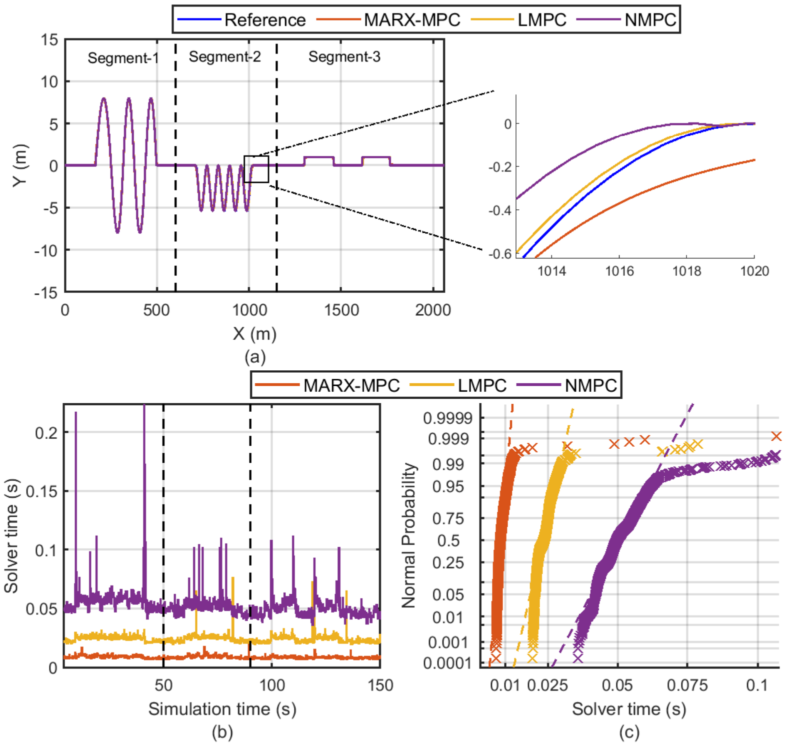

Figure 7.8: Comparisons of proposed MARX-MPC with conventional MPCs.(a) Trajectory produced by the predictive controllers. (b) Solver time for each MPC controller and (c) normal distribution of solver time.

Figs. 7.8.b-c show that the proposed architecture can reduce the computational time of the MPC control by up to 63% concerning the LMPC and up to 88% concerning the NMPC (see Tables 7.5 and 7.6). These results indicate that the proposed approach is promising for real-time applications with trajectory deviation consistent with current models. Fig. 7.8.c shows the normal distribution of solver time for the presented architectures and highlights the

PUC-Rio - Certificação Digital Nº 1912774/CA

computational advantage of the proposed architecture. Here, the solver time for each predictive controller is recorded at each time interval.

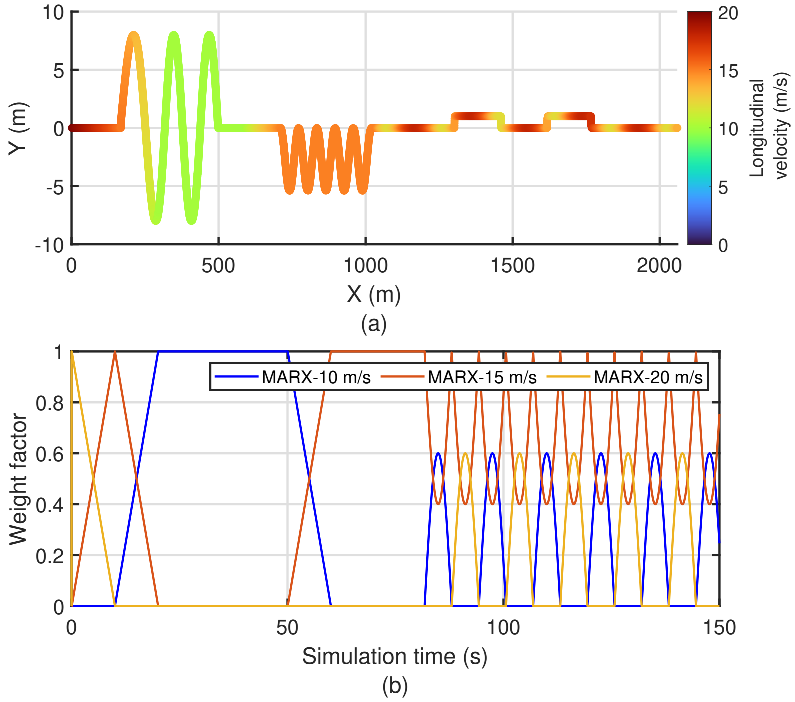

Figure 7.9: (a) Longitudinal velocity through the trajectory, and (b) Weight factors regarding MARX-MPC models – 10, 15, and 20 m/s.

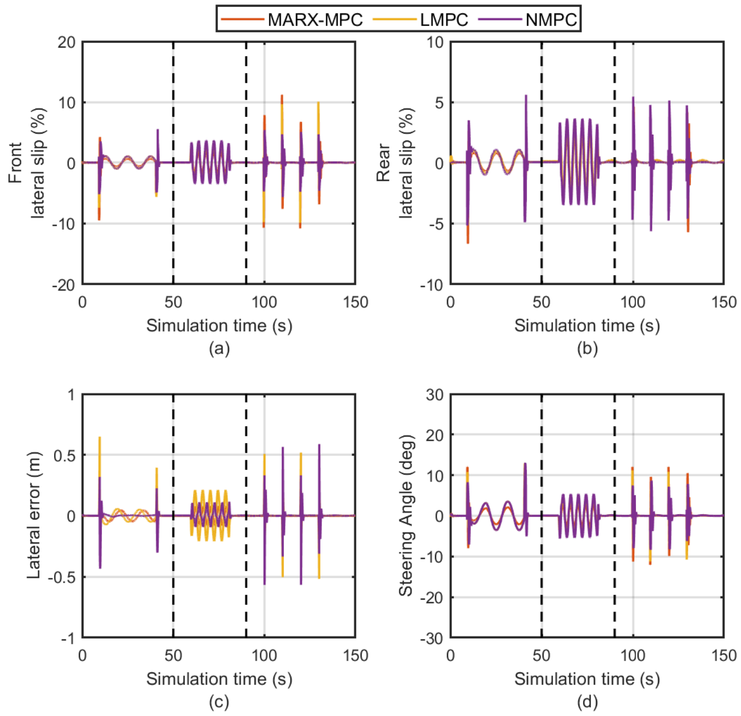

Figure 7.10: (a) Front lateral slip, and (b) rear lateral slip over time for each MPC architecture. (c) Lateral position error and (d) steering angle over time for each of the MPC architectures.

PUC-Rio - Certificação Digital Nº 1912774/CA

Fig. 7.8.b shows the solver time intervals for each architecture. The NMPC presented a high computational time consumption during vehicle maneuvers. Therefore, the solver's average time increases due to those instants. The interesting point is that the sudden change in trajectory (segment-3) increases computational performance in LMPC and NMPC controllers. However, for the proposed MARX-MPC, the computational time is reduced and almost insensitive, highlighting the computational efficiency of the proposed methodology. Fig. 7.9 shows the reference velocity through the trajectory and the weights through time associated with the ARX models of 10, 15, and 20 m/s. One can see the robustness of the proposal in the face of different speed profiles.

The computational cost affecting the performance of the NMPC controller is due to the increase of the front and rear lateral slip, as shown in Figs. 7.10.a-b. The plant controlled by the MARX-MPC also presented an accentuated level of slip. However, the computational impact is reduced. Due to the vehicle maneuvering movements lagging by the prediction distance, relevant lateral errors can be seen in segments 1 and 3 (Fig. 7.10.c). It is necessary to have a fair prediction horizon value for the required trajectories. A high horizon causes the vehicle to anticipate the trajectory to be followed. However, a significant lag may occur between the predicted and actual trajectory. On the other hand, a very short horizon makes it difficult for the vehicle to follow a sudden trajectory change, as seen in segment-3.

As seen, the lateral slip tends to increase at the maneuvering points. Therefore, a significant computational calculation is required for the nonlinear tire equation in the NMPC control. On the other hand, the proposed MARX-MPC model becomes insensitive to maneuvers of different complexities emphasizing its robustness. Tables 7.5 and 7.6 report the RMS errors, the average solver time, and the percentage reduction using the proposed approach for each one of the architectures during each segment, corroborating the graphical results presented. The NMPC has the slightest average error during the first and second trajectory segments. However, the proposed approach achieved the slightest error through the third segment. In addition, compared to the NMPC, the proposed approach could reduce the solver time by up to 88%.

The proposed approach could reduce lateral error by up to 41% and solver time by up to 75% compared to the LMPC approach. An important point to note is that the values presented for the MARX-MPC are compatible with recent works [18, 72] for a similar range of longitudinal velocity.

PUC-Rio - Certificação Digital Nº 1912774/CA

PUC-Rio - Certificação Digital Nº 1912774/CA

Table 7.5: RMS error over the entire trajectory and time averaging of the MPC solver for the predictive controls.

	Segment-1		Segment-2		Segment-3	
Controller	RMS error (m)	Average solver time (s)	RMS error (m)	Average solver time (s)	RMS error (m)	Average solver time (s)
MARX-MPC	0.0512	0.0094	0.0450	0.0086	0.0520	0.0080
LMPC	0.0615	0.0248	0.1090	0.0238	0.0776	0.0331
NMPC	0.0476	0.0562	0.0432	0.0503	0.0693	0.0685

Table 7.6: Error and average solver time reductions in percent using the proposed MARX-MPC compared to LMPC and NMPC.

	Segment-1		Segment-2		Segment-3	
Compared Controller	% RMS error reduction	% Average solver time reduction	% RMS error reduction	% Average solver time reduction	% RMS error reduction	% Average solver time reduction
LMPC	16.75	62.10	41.28	63.87	32.95	75.83
NMPC	-7.56	83.27	-4.17	82.90	24.96	88.32

PUC-Rio - Certificação Digital Nº 1912774/CA

7.3 Overall discussion and impacts

The results presented in the previous sections provide a broader view of identified vehicle models applied to predictive control during the trajectory tracking problem. The main objective of this contribution is to design a data-driven MPC framework that can effectively utilize a combination of simplified identified vehicle models to capture nonlinear effects that individual modeling cannot do satisfactorily. The proposed methodology also allows capturing nonlinear characteristics of the powertrain and tires to be combined with predictive control strategies during different maneuvers. Next, it is argued that the proposed method advantages current control practices. The method allows the construction of vehicle models based on data with satisfactory accuracy concerning the error-based metrics, with similar results in recent literature (0.05-0.50 m) [18, 26, 72]. In addition, the proposed methodology, when used to predict the MPC control, has better computational performance (reduction between 60% and 90%) compared to conventional models used in the literature (linear and nonlinear model) [108, 142, 143]. The proposed methodology still proved insensitive to the increase of lateral slip. This result shows the potential use of the methodology in embedded solutions for control laws.

Once the best architecture is found to fit the simulated data for each longitudinal speed (10, 15, and 20 m/s), a controller is designed based on a weight cost function from the identified model, which is a function of vehicle speed. The performance of the MARX-MPC control is compared with LMPC and NMPC literature models. Furthermore, a reduction in computational time is observed using the proposed control even with the increase of identified models in the cost function. Thus, the data-driven models proposed here are better for predictive models in MPC in terms of computational effort. Other computational time and trajectory tracking improvements can be obtained using more complex models than the presented strategy.

In general, data-driven vehicle models offer an alternative way to verify and analyze nonlinear aspects that conventional models cannot do properly. Furthermore, the results presented provide implications regarding data-based modeling for MPC. The procedure reduces computational time while maintaining satisfactory trajectory tracking, as observed in tests performed (Figs. 7.8 and 7.10). In addition, this approach indicates that data-based modeling for efficient predictive control can correctly replace physically inspired models for vehicles in medium and high-speed driving situations. Such a procedure agrees with more recent works that approach the identification of vehicle dynamics [3, 37, 48, 98]. The proposed data-driven approach is highly beneficial when formulating a complex mathematical model of the vehicle. Vehicles come in different shapes and sizes and have some degree of variation. Thus, designing an analytical model for each vehicle is difficult and time-consuming. In most cases, analytical simplification can introduce uncertainties due to unmodeled dynamics. On the other hand, the method presented herein builds on other physics-based models, enhancing the knowledge regarding predictive modeling methods.

7.4 Summary

Trajectory tracking is essential to autonomous ground vehicles since it reduces accidents and improves passenger comfort. Thus, Model predictive control (MPC) has proved to be an effective solution for path-tracking. The essential aspect of MPC control is the prediction model. A simplified dynamic model can achieve reduced computational cost; however, the trajectory error and uncertainties regarding the system affect its results. On the other hand, a nonlinear vehicle model can improve the trajectory error to the detriment of computational effort.

The present contribution proposes a framework for enhancing path-tracking in terms of computational and accuracy performances. It can be achieved by applying data-driven models to approximate lateral vehicle dynamics and using them as prediction models for efficient predictive control. Specifically, three different ARX models are identified, considering the vehicle's steering angle and lateral position as input and output, respectively. Then, a supervisor that weights the importance of the identified models is designed to guarantee an efficient predicted model. The cost function can be designed for the MARX-MPC controller using the result derived from the supervisor.

The performance of the proposed controller has been evaluated in a virtual environment with a trajectory segmented in different curvatures, considering a non-constant velocity profile. Moreover, the performance of the proposed approach has been compared to conventional approaches, specifically LMPC and NMPC controllers. Results indicate that using identified architectures as prediction inference provided a computational time reduction by up to 88%, with accuracy comparable to the conventional MPC approaches.

PUC-Rio - Certificação Digital Nº 1912774/CA

8
Hybrid physics-based and black-box data-driven nonlinear identification of vehicle dynamics

Due to the nonlinear interactions, vehicle modeling remains challenging, mainly under racing conditions. Accurate and feasible physics-based vehicle modeling requires previous knowledge about the vehicle. On the other hand, data-driven algorithms may be used to represent measured data. However, any model is subject to discrepancies due to simplifications or numerical approximations. Thus, to cope with these problems, this proposed contribution deals with a hybrid approach, combining data-driven approaches with black-box modeling of the discrepancies. This approach is chosen to improve the accuracy of vehicle modeling by proposing a discrepancy model to capture mismatches between vehicle models and measured data. The methodology is tested with a physics-based model often used for model-based control, a state-space model recently reported, and a purely black-box model when the proposed method showed significant improvement in predictive capability. The dataset comprises a driving section in racing conditions with the vehicle at handling limits. The results demonstrate that the hybrid approach improves vehicle modeling, reducing the model's mismatches by up to 28% in terms of RMSE. The proposed approach may lead to significant implications for control applications applied to autonomous vehicles, specifically through racing conditions where the vehicle's limits become critical. That is the condition that the traditional approaches fail and our method becomes more relevant.

PUC-Rio - Certificação Digital Nº 1912774/CA

8.1
Proposed approach

This section introduces the proposed approach regarding identifying physics-based, state-space, and black-box vehicle models. Moreover, the mathematical formulation of the proposed hybrid approach is also stated.

8.1.1
Identification of physics-based models

An optimization problem estimated the unknown parameters in the physics-based model through the least-squares approach. The procedure is given by the following:

– Estimate the parameters of the longitudinal dynamics k_e, k_d, and k_r;

– Estimate the parameters of the lateral dynamics C_f, C_r, and I_z.

The procedure adopted in this work is an optimization problem with the cost function defined as a normalized mean-square between measured data and predicted data:

$$J(\theta) = \sum_{i=1}^{N_t} (y(t_k) - \hat{y}(t_k, \theta))^T (y(t_k) - \hat{y}(t_k, \theta)) \ , \tag{8-1}$$

where $y(t_k)$, and $\hat{y}(t_k)$ are the output data and the predicted data at time t_k, given the vector of the parameters θ. For the longitudinal dynamics, $y = [v_x]^T$, and for the lateral dynamics, $y = [v_y, r]^T$. In addition, N_t is the length of the measurement data vector. Particularly for the lateral dynamics model (bicycle model), the longitudinal velocity present in Eqs. (3-19)-(3-20) is not considered a constant value but the measured longitudinal velocity through time. Therefore, the bicycle model is considered in a coupling way in which the longitudinal dynamics directly affect the lateral dynamics.

The identification is solved using the Ipopt software package [124] with the open-source software - CasAdi [125], which is a symbolic framework commonly used for optimization problems. Moreover, the ODE system is solved with CVodes from the SUNDIALS Suite [127], used for fast and precise numerical solutions.

PUC-Rio - Certificação Digital Nº 1912774/CA

8.1.2 Identification of state-space model

Considering vehicle identification in [43], the procedure was successfully implemented. The second-order ($n_d = 2$) state-space parameters were estimated using the PEM method [114], and the initialization occurred using the N4SID algorithm. In addition, the "*ssest*" function that combines the N4SID state-space initialization with the PEM estimation is also applied. Finally, in this contribution, the output is given by $y_d = [v_x \ r]^T$, and $u_d = [P \ \delta]$, to compare with the linear approaches.

8.1.3 Black-box vehicle modeling

Following the method described in Section 4.2, a NARX black-box model is implemented considering $y_d = [v_x \ r]^T$, and $u_d = [P \ \delta]$, in order to compare with the physics-based and state-space approaches. Besides, the maximum lags

(na and nb) and artificial neural network architecture were chosen after trial and error.

8.1.4 Hybrid model

The present study aims to use vehicle model approaches with discrepancy models using a hybrid approach to enhance accuracy. One of the main advantages is the possibility of using simplified model techniques to model the vehicle. The general idea is to combine engineer insights with aspects missed by vehicle modeling. Therefore, physics-based, state-space data-driven, and black-box vehicle models are used as a mean function. In addition, NARX models were used as discrepancy models by the following equation:

$$y_d(k) = \underbrace{y_m}_{\text{vehicle model}} + \underbrace{F[y_e(k-1), ..., y_e(k-ne), u(k-1), ..., u(k-nb)]}_{\text{discrepancy model}} \quad (8\text{-}2)$$

where y_d is the output of the proposed approach. The vehicle models are represented by the mean function (y_m), which is a function of the inputs $u(k)$ and states $x(k)$ for the physics-based model and state-space data-driven approaches. On the other, for black-box vehicle models, the mean function (y_m) is a function given by $F[y(k-1), ..., y(k-na), u(k-1), ..., u(k-nb)]$, i.e., the inputs $u(k)$ and outputs $y(k)$. The discrepancy term is a function of the system inputs $u(k)$ and itself $y_e(k)$. Here, the maximum lags (na, nb, and ne) and artificial neural network architectures are chosen after some trial and error. To facilitate the understanding and comparison between the methods, Table 8.1 shows the acronym used to distinguish the methods and their results. Finally, Fig. 8.1 shows a general overview of the proposed approach.

Table 8.1: Acronym description of the models and references used.

Acronym	Description	Reference
M1	Bicycle coupled model	[83]
M2	Longitudinal model	[43]
M3	State-space model	[37]
M4	Black-box model	proposed
M1-H	Hybrid bicycle coupled model	proposed
M2-H	Hybrid longitudinal model	proposed
M3-H	Hybrid state-space model	proposed
M4-H	Hybrid black-box model	proposed

PUC-Rio - Certificação Digital Nº 1912774/CA

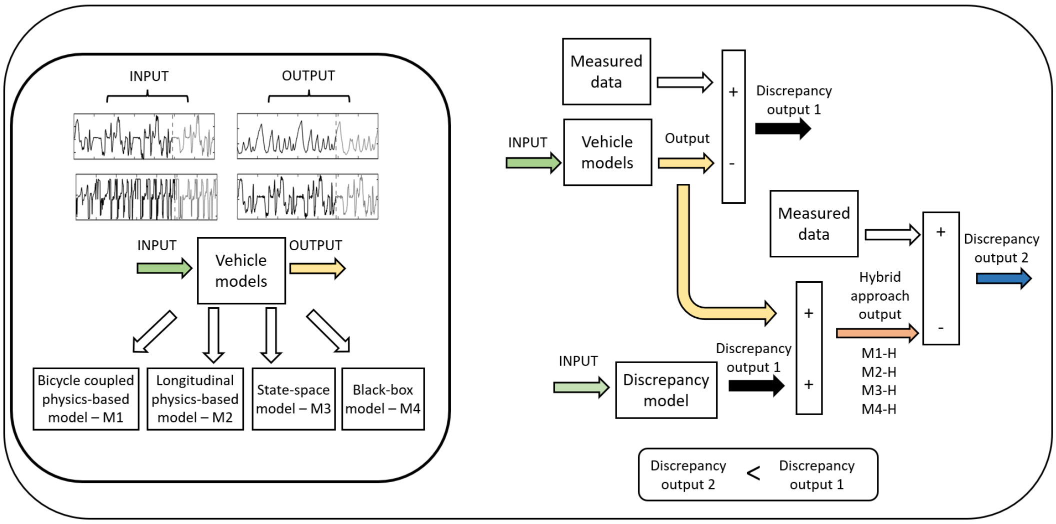

Figure 8.1: General overview of the proposed approach: Once the identified models (M1; M2; M3; M4) are obtained, they produce discrepancy data to be modeled by the NARX black-box approach. Next, the hybrid approach (M1-H; M2-H; M3-H; M4-H) comprises the output of the identified vehicle models summed with their respective identified mismatch models.

8.2 Experimental data

The experimental data were gathered in an instrumented Ferrari 250 LM Berlinetta GT during driving sections on The 2014 Targa Sixty-Six event that took place at the Palm Beach International Raceway in the US (Fig. 8.2). The road course at the Raceway was a 3.3 km-long track, 10-turn circuit (clockwise orientation) featuring a 1 km straight. The Revs Program at Stanford [144] made the data set publicly available.

The latitude/longitudinal data were acquired by a global navigation satellite system (GNSS)-aided inertial navigation system. The vehicle's position on a fixed set of coordinates is obtained by converting the longitudinal and latitude data using equirectangular approximation. The racing vehicle signals were sampled and recorded using MoTec hardware (for details, see [105]). Only a section of the available data was selected, considering longitudinal velocity greater than zero. The data set was divided into training and test split of 70/30%, respectively. The measured data comprises signals sampled at 100 Hz and 1000 Hz. However, the power spectrum of the relevant measurements showed that over 90% of the power was concentrated below 10 Hz. Then, in order to the view to reduce computational time, the data set was resampled at 20 Hz, similarly as in [37, 43]. Fig. 8.3 shows the measurements during the racing section.

PUC-Rio - Certificação Digital Nº 1912774/CA

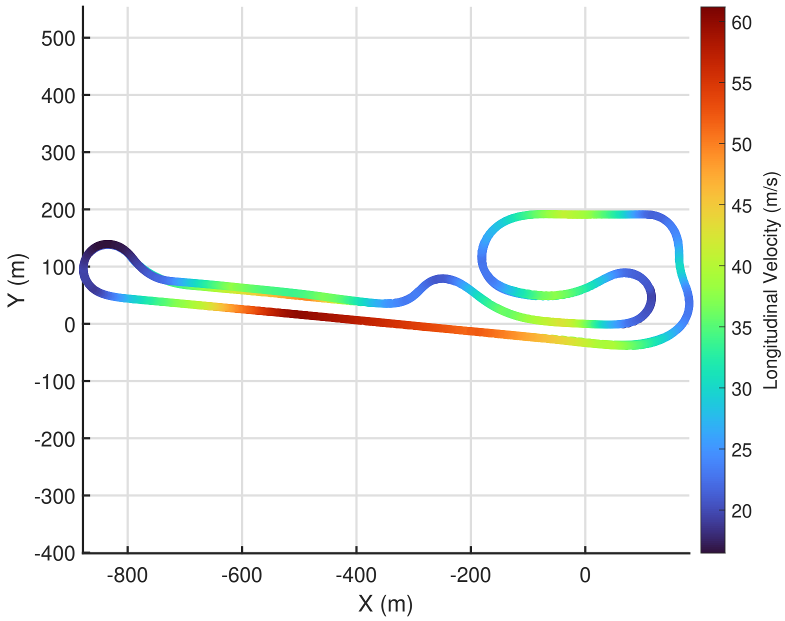

Figure 8.2: Driving route at the Palm Beach International Raceway.

PUC-Rio - Certificação Digital Nº 1912774/CA

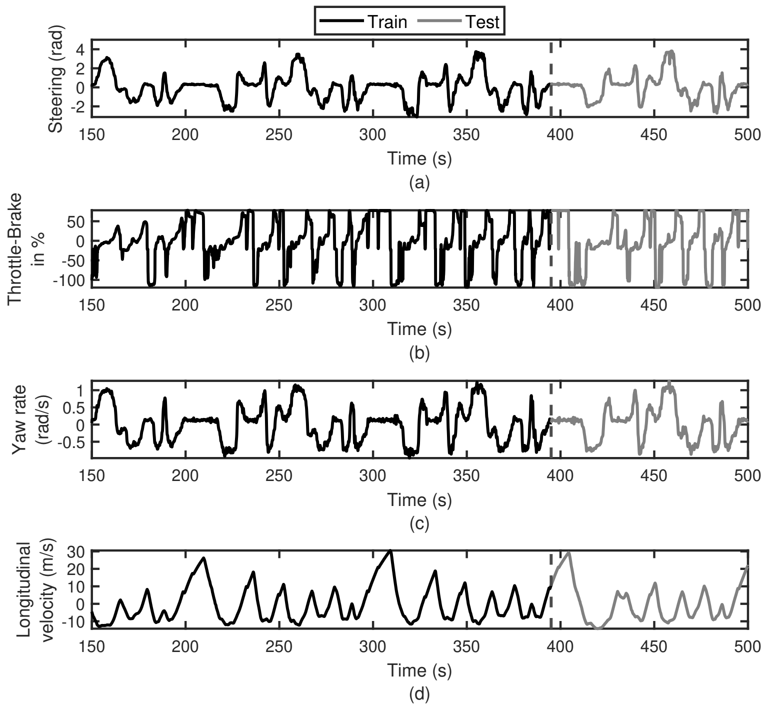

Figure 8.3: Measurements acquired during a racing section at the Palm Beach International Raceway. (a) Steering angle. (b) Difference between throttle and brake signals from the pedal in percent. (c) Yaw rate. (d) Longitudinal velocity.

PUC-Rio - Certificação Digital Nº 1912774/CA

8.3 Results

For brevity, the focus is given to the test data set. In addition, the models are derived from free-run simulation, in which actual input data is used to derive the predicted data. Thus, the measured data is used only in the initial condition. With this procedure, the predictions accumulate errors, making it easier to show whether the identified model is valid [94]. The NARX models with neural networks are tested, varying the order of the model (input and output lags). Namely, the procedure is performed by varying the number of lags from 5 to 10. The number of neurons was also tested by varying from 30 to 50 with Exponential Linear Unit (ELU) activation through hidden layers varying from 2 to 10. Sections 8.3.1 and 8.3.2 present the results for lateral and longitudinal dynamics, respectively. The values of the optimized parameters for the state-space (with mean of data samples removed and normalized data) and physically derived models can be found in Tables 8.4 and 8.5. Finally, the metrics to be compared are the Fit (F), RMSE, and variance-accounted-for metric (VAF).

8.3.1 Lateral dynamics modeling

The results performed by the models, including the proposed approach for lateral dynamics, can be seen from Fig. 8.4 to 8.6, while Table 8.2 summarizes their performance metrics.

Table 8.2: Performance metric summary - Lateral Model.

Model	Yaw rate			% Increase		% Reduction
	F%	VAF%	RMSE	F	VAF	RMSE
M1	72.87	93.21	0.054	16.85	5.90	25.97
M1-H	85.15	98.72	0.040			
M3	80.34	96.14	0.046	10.94	2.80	25.65
M3-H	89.14	98.83	0.034			
M4	88.53	98.68	0.351	5.42	0.88	23.58
M4-H	93.33	99.55	0.027			

Although the state-space model (M3) approach can accurately simulate the yaw rate trend, the proposed approach (M3-H) can outperform it with an RMSE reduction of 25.65%. The proposed hybrid (M1-H) approach also outperforms the bicycle model (M1) in terms of accuracy by 25.97% of RMSE reduction. In addition, considering the black-box approach (M4 and M4-H), a reduction of 23.58% in RMSE can be achieved.

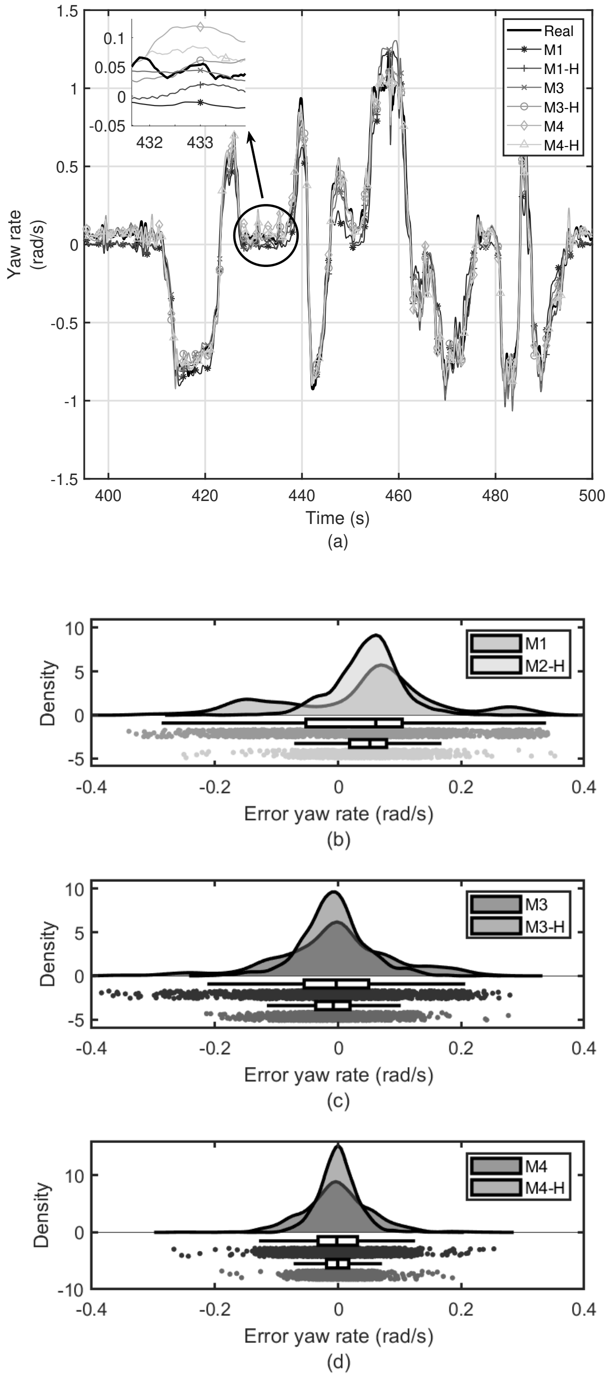

Figure 8.4: (a) Comparison of measured and predicted yaw rate using the proposed approach considering test data. (b) Raincloud considering the error between the state-space model and its hybrid approach. (c) Raincloud considering the error between the bicycle model and its hybrid approach. (d) Raincloud considering the error between the black-box model and its hybrid approach.

PUC-Rio - Certificação Digital Nº 1912774/CA

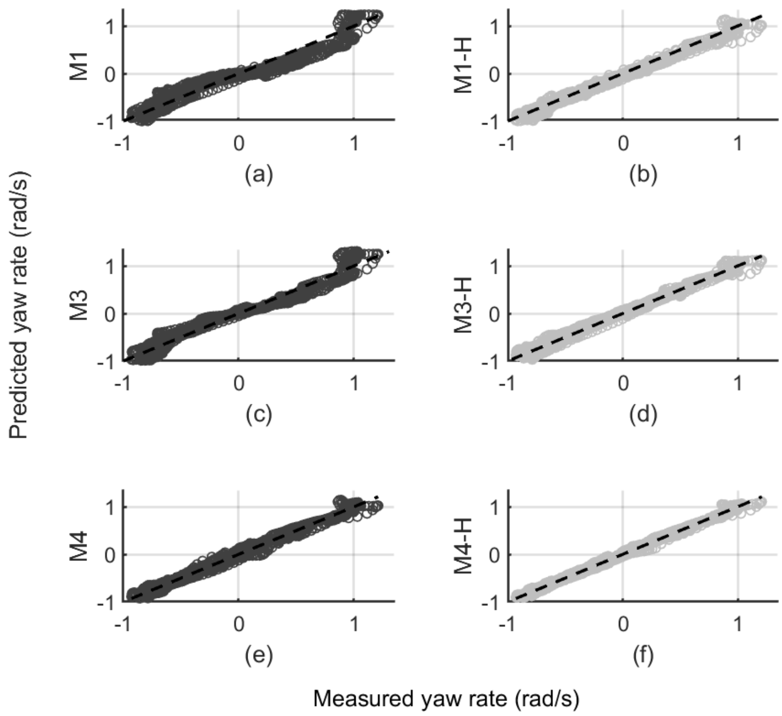

Figure 8.5: Correlation between the measured and the simulated yaw rate over the entire test data set. (a) Bicycle model and its (b) hybrid approach. (c) State-space model and its (d) hybrid approach. (e) Black-box model and its (f) hybrid approach.

PUC-Rio - Certificação Digital Nº 1912774/CA

Fig. 8.4 compares the test data and the prediction of the approaches, particularly in Figs. 8.4.b-d, one can see raincloud plots considering the error between predicted and measured data. The hybrid approach can enhance the vehicle model's accuracy, attenuating the error, as seen by the normal distribution with a mean of next to zero. Besides, the correlation comparisons are given in Fig. 8.5. The proposed approach can more accurately explain the variances in the measurements, with a percentage increase of variance accounted for 5.90%, 2.80%, and 0.88% regarding the bicycle, state-space, and black-box models, respectively. In addition, a model fit increase of 16.85%, 10.94%, and 5.42% can also be observed using the proposed approach, respectively. The black-box minor improvement results can be explained by the fact that the black-box approach (M4) can already represent most of the measured data.

Fig. 8.6 depicts the error between the measured and predicted yaw rate along the trajectory. It also confirms the same allusions made before that the hybrid approach considerably improves the results. Additionally, considering the bicycle, state-space, and black-box models (Figs. 8.6.a, 8.6.d, and 8.6.g), higher errors during entering and exiting in some curves can be seen. On the other hand, the proposed approach (Figs. 8.6.b, 8.6.e, and 8.6.h) can reduce

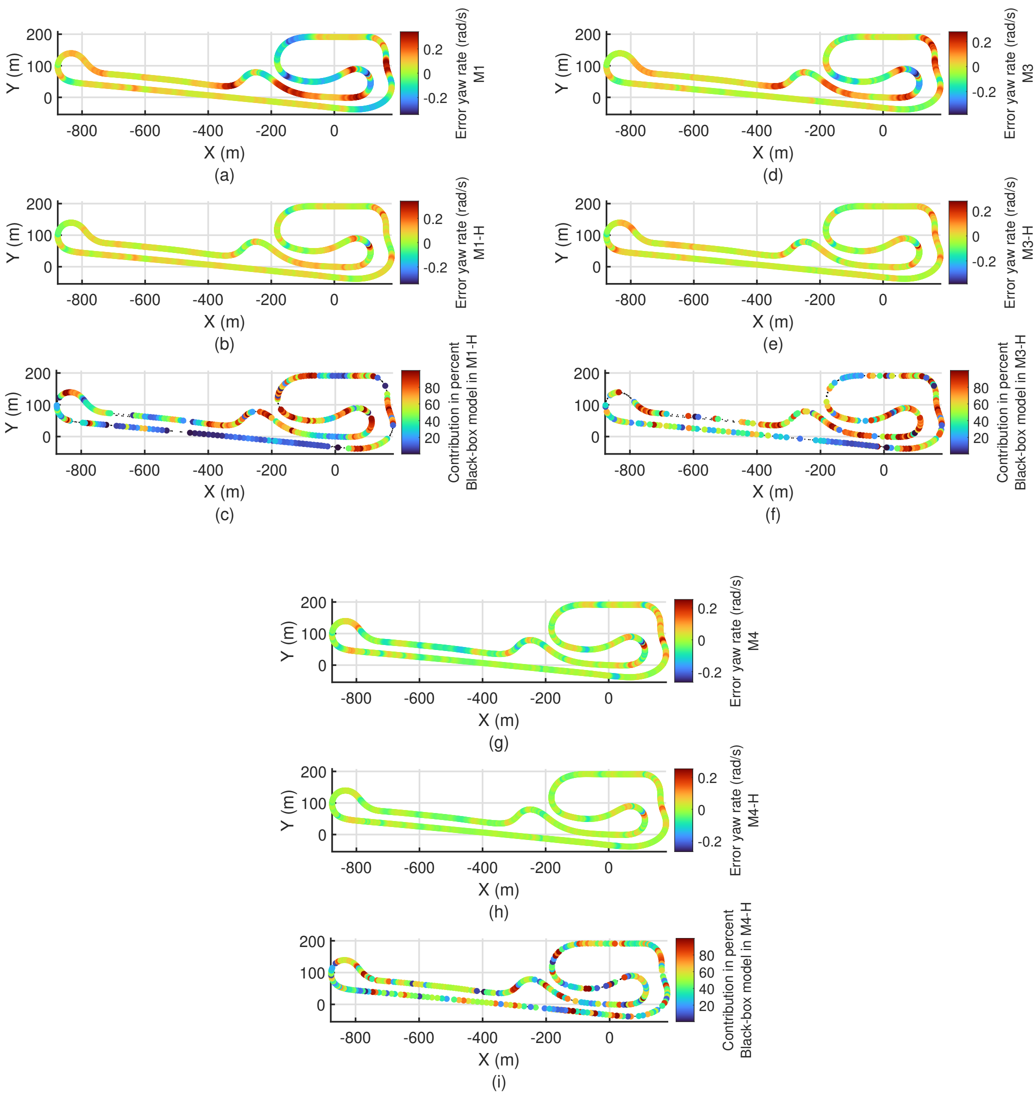

Figure 8.6: Error between the measured and the simulated yaw rate over the entire test data set. (a) Physics-based model and its (b) hybrid approach. (c) Discrepancy model contribution in percent - physics-based model. (d) State-space model and its (e) hybrid approach. (f) Discrepancy model contribution in the percent - state-space model. (g) Black-box model and its (h) hybrid approach. (i) Discrepancy model contribution in percent - black-box model.

PUC-Rio - Certificação Digital Nº 1912774/CA

the error along the entire trajectory. The points in which error reduction occurs are the points in which the discrepancy models have greater importance in the hybrid architectures, as seen in Figs. 8.6.c, 8.6.f, and 8.6.i.

8.3.2 Longitudinal dynamics modeling

The results for the longitudinal model and the hybrid approach can be seen from Figs. 8.7 to 8.9, while Table 8.3 summarizes their performance metrics.

Concerning the longitudinal dynamics, the proposed approach (M3-H) led to a reduction of 12.68% in the RMSE metric compared with the data-driven model. Besides, a reduction of 18.55% in the RMSE metric is obtained using the proposed physics-based model (M2-H) approach. On the other hand, a reduction of 28.57% can be achieved by the proposed approach considering a black-box vehicle model (M4 and M4-H). From Fig. 8.7, one can see the comparison between the measured and predicted longitudinal velocity data. Figs 8.7.b-d show the raincloud plots considering the error data. The proposed approach can also increase the vehicle model's approaches, reducing the error mainly in the extremes. The correlation comparisons are given in Fig. 8.8.

Table 8.3: Performance metric summary - Longitudinal Model.

Model	Longitudinal velocity			% Increase		% Reduction
	F%	VAF%	RMSE	F	VAF	RMSE
M2	65.49	90.01	0.286	17.44	5.70	18.55
M2-H	76.91	95.13	0.233			
M3	67.97	91.81	0.275	10.78	2.28	12.68
M3-H	75.30	93.91	0.240			
M4	75.66	95.72	0.240	15.67	3.31	28.57
M4-H	87.52	98.89	0.171			

PUC-Rio - Certificação Digital Nº 1912774/CA

The proposed approach can represent the measurement variances with higher accuracy. The VAF percentage increases by up to 5.70% using the proposed approach. Moreover, a percentage increase of 17.44% in model fit can be achieved using the proposed approach with the vehicle models. The lower accuracy compared to the lateral dynamic models is given by the transition between entering and exiting curves which could not be predicted accurately.

From Fig. 8.9, one can observe the error between measured and predicted longitudinal velocity along the trajectory considering the test data. Considering the physics-based, state-space, and black-box models (Figs. 8.9.a, 8.9.d, and 8.9.g), higher errors can be observed when entering and exiting the curves. Due to the maneuvers at the handling limits, the prediction cannot adequately capture the longitudinal velocity in zones with the transition between braking and velocity recovery. On the other hand, the hybrid approach (Figs. 8.9.b, 8.9.e, and 8.9.h) can reduce the discrepancies along the entire trajectory. It can be seen from Figs. 8.9.c, 8.9.f, and 8.9.i, the contribution given to the vehicle models using the discrepancy NARX models.

The main contributions occur in zones with a significant reduction of longitudinal velocity followed by a sudden velocity increase (before and after long straights). Contributions more outstanding than 90% in those situations

PUC-Rio - Certificação Digital Nº 1912774/CA

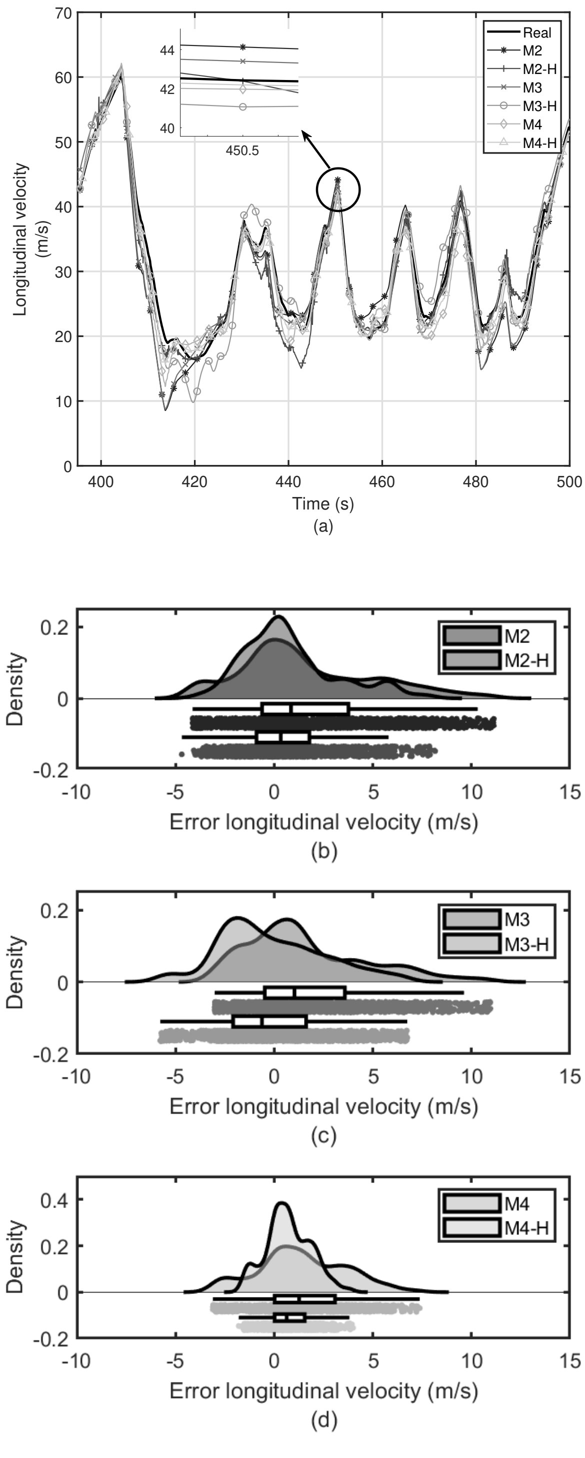

Figure 8.7: (a) Comparison of measured and predicted longitudinal velocity using the proposed approach considering test data. (b) Raincloud considering error between longitudinal physics-based model and its hybrid approach. (c) Raincloud considering the error between the state-space model and its hybrid approach. (d) Raincloud considering the error between the black-box model and its hybrid approach.

can be observed, and other improvements can also be noted along the entire racetrack. The simulation results demonstrated that including a discrepancy model attenuates the model discrepancies.

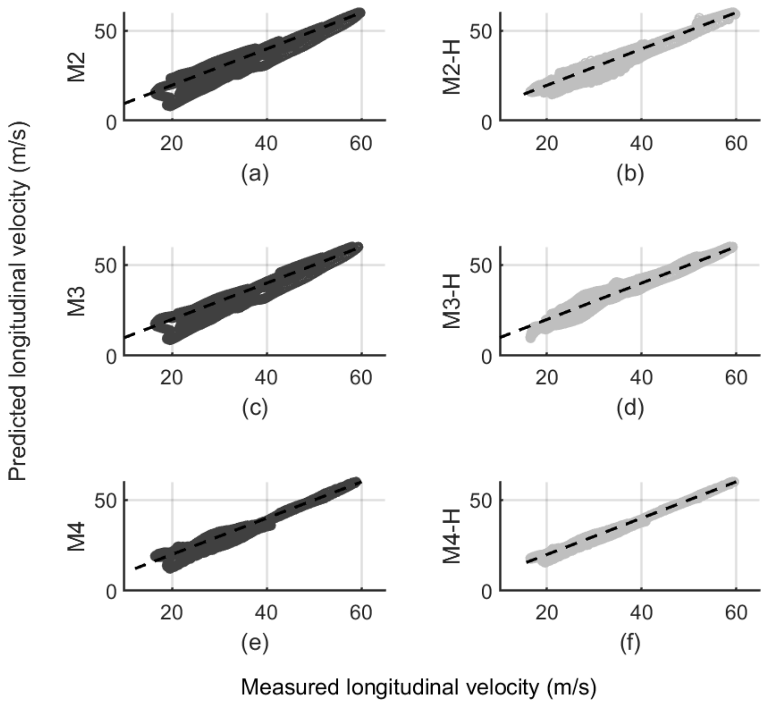

Figure 8.8: Correlation between the measured and the simulated longitudinal velocity over the entire test data set. (a) Longitudinal physics-based model and its (b) hybrid approach. (c) State-space model and its (d) hybrid approach. (e) Black-box model and its (f) hybrid approach.

PUC-Rio - Certificação Digital Nº 1912774/CA

Table 8.4: Parameters estimates for the second-order state-space model.

$$A_d = \begin{bmatrix} 0.282 & 0.132 \\ -0.003 & 0.995 \end{bmatrix} \; B_d = \begin{bmatrix} -1.355 & 0.041 \\ -0.005 & 9.5656 \times 10^{-4} \end{bmatrix}$$

$$C_d = \begin{bmatrix} -0.579 & 0.821 \\ -0.061 & 15.337 \end{bmatrix}$$

Table 8.5: Parameters estimates for the longitudinal and bicycle coupled dynamic models.

Longitudinal			Bicycle coupled		
k_e	k_d	k_r	Iz	C_f	C_r
61.647	115.872	-0.231	2558	3438	82424

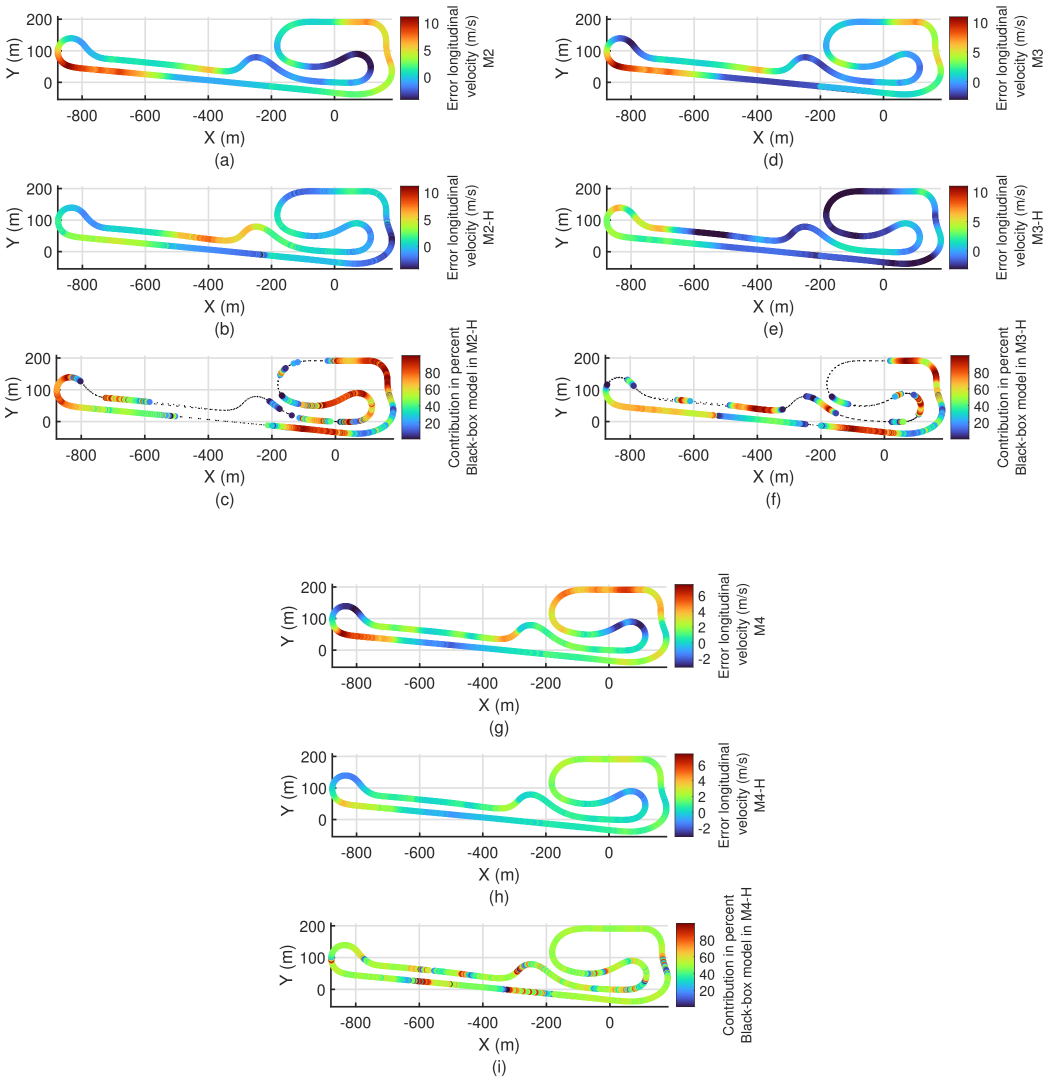

Figure 8.9: Error between the measured and the simulated longitudinal velocity over the entire test data set. (a) Physics-based model and its (b) hybrid approach. (c) Discrepancy model contribution in percent - physics-based model. (d) State-space model and its (e) hybrid approach. (f) Discrepancy model contribution in percent - state-space model. (g) Black-box model and its (h) hybrid approach. (i) Discrepancy model contribution in percent - black-box model.

8.4 Overall discussion and impacts

The results presented in Section 8.3.1 and Section 8.3.2 give an insight into learning model discrepancy applied to enhance vehicle modeling. As will be discussed next, the proposed approach contributes to the current state of the art and present an important contribution for model-based control [30], and digital twins applied to simulation and monitoring [106].

PUC-Rio - Certificação Digital Nº 1912774/CA

The simulation results demonstrated that the vehicle models could reproduce the real data (yaw rate) to a certain degree. However, including a residual learning model component enhances the accuracy by up to 25.97%. In addition, a percent increase by up to 16.85% of data fit is observed. Regarding the simulation results concerning longitudinal dynamics, one can observe a reduction of up to 28.57% in RMSE, considering the hybrid approach applied to the black-box model. Moreover, a percent increase by up to 17.44% of data fit can be seen in the results. As the black-box outperformed other approaches and was further improved by stacking another black-box in series, the results presented can be further improved. Exhaustive hyperparameter search, architecture search [145], or another available black-box approach can be applied to define the more appropriate application. These topics all fall out of the scope of the present contribution.

The hybrid model gives an alternative way to capture nonlinear aspects present in race conditions in which driving operation at the vehicle's limits becomes critical. In addition, the precision improvement inspires using the proposed model in control applications such as Model Predictive Control (MPC) which requires reliable and accurate models. The proposition of better models in terms of predictive capability is in line with recent works that aim to identify the vehicle dynamics [3, 37, 48, 98] in medium and high-speed driving situations. Specifically, the hybrid approach inspires its application in different dynamic situations such as rotating dynamic analysis [102], trajectory analysis [59], and structural health monitoring [146]. One of the areas for improvement of the proposed approach is that the sudden change in the vehicle parameters or track conditions may decrease the prediction performance. This problem could be solved by applying online training [18, 98] or considering multiple surface friction values during experimental tests [3, 81]. However, the proposed hybrid methodology compensates for unmodeled aspects commonly neglected by physics-based models, mainly in extreme handling conditions.

PUC-Rio - Certificação Digital Nº 1912774/CA

8.5 Summary

This contribution proposes a hybrid approach to identify a ground vehicle's dynamics. The dynamic system consists of an instrumented vintage racing car during driving sections on the Palm Beach racetrack [105]. First, the measured data estimates the unknown parameters of the longitudinal and lateral dynamic models considering driver inputs and vehicle outputs. Next, data-driven models are investigated to simulate the racing vehicle's longitudinal velocity and yaw rate. Finally, the mismatches between the real and predicted data are modeled using a black-box technique. Thus, the proposed hybrid approach adds the black-box component to different vehicle models.

The proposed approach applied to vehicle dynamics has the potential to 1) increase the accuracy of predicted data with arbitrary precision, reducing the effect of neglected modeling aspects, 2) be used effectively to improve the vehicle modeling through race conditions, 3) be easily reformulated considering different models complexity depending on the application, including driver inputs for higher-level control applications. Developments indicated that the identified models can reproduce the measured data and that adding the black-box component effectively reduces the model's mismatches by up to 28% in terms of RMSE. Moreover, the fit percentage increases by up to 17%. Therefore, the results demonstrated the precision improvement of the predictions.

PUC-Rio - Certificação Digital Nº 1912774/CA

Part IV

Final Remarks

PUC-Rio - Certificação Digital Nº 1912774/CA

9
Conclusion

The present work dealt with machine learning and system identification methods to enhance vehicle modeling and efficiently perform trajectory tracking using model predictive control. As mentioned throughout this thesis, the effectiveness of the MPC approach is directly affected by accurate models. Regarding vehicle dynamics, simplified dynamic models may lead to accurate results with low computational cost in conditions where nonlinear aspects are neglected. On the other hand, a nonlinear dynamic model may lead to an unsuitable application. Therefore, data-driven approaches used in this thesis rise as a potential solution guaranteeing a trade-off between precision and computational efforts. Explicitly speaking, three different approaches were proposed in this thesis: (i) ANNs can be applied to approximate nonlinear tire data with arbitrary precision, and (ii) a combination of ARX models with a new cost function, both to enhance trajectory tracking with efficiency; (iii) hybrid approach combining conventional vehicle models from literature with a discrepancy model to enhance the vehicle modeling.

The first formulation focuses on improving MPC using neural tires. RBF and MLP architectures are proposed and compared regarding tire curve approximations considering simulated and experimental data. In addition, the application of neural tires in predictive control is investigated to provide nonlinear control laws. Being so, the proposed MPC controllers are submitted to perform trajectory tracking in different scenarios. Results based on error-based metrics and computational efforts (reduction by up to 25%) ensure the advantage of the proposed approach compared to literature controllers.

Next, a combination of ARX models is implemented to enhance the MPC controller efficiently. The main objective of this contribution is to implement an MPC framework that allows a combination of simplified identified vehicles. Thus, it is possible to capture nonlinear aspects (including powertrain and tires) that individual modeling may not be able to do efficiently. The results show the potential of the proposed approach to provide nonlinear control laws effectively. It was possible to increase the accuracy (up to 40%) in terms of error-based metrics while having lesser computational effort (reduction by up to 88%) than conventional predictive controllers. Particularly, simulated and

PUC-Rio - Certificação Digital Nº 1912774/CA

experimental data were used to verify the potential of the proposed methods.

Lastly, regarding the hybrid approach, the results showed the potential to enhance the prediction of vehicle data reducing the effects of neglected modeling aspects (accuracy increased by up to 28%). Moreover, the proposed method can be easily adaptable and reformulated by modifying the vehicle and the black-box models according to the required application. Particularly in this case study, the proposed approach may lead to significant implications, specifically considering racing conditions where the vehicle's limits become critical.

Even though this work was focused on the case of RBF and MLP neural architectures for machine learning and ARX/NARX models for system identification approaches, due to their characteristics and properties, the methodologies proposed herein are flexible concerning the definition of the model. It is possible to change/tune either the ANN architectures or black-box models. This flexibility may result in better results for specific cases, depending on the characteristics of the required system. In addition, the predictive controllers can also be flexible and adjusted depending on the controllable variables and model configurations.

Overall, the proposed approaches regarding data-driven models applied to vehicle dynamics are an alternative way to ensure the capturing of nonlinear aspects in public road and race conditions. Moreover, the results presented herein indicate that the proposed methodologies inspire their use with MPC controllers, which require accurate and reliable models to ensure the compromise between path-tracking accuracy and computational costs.

9.1 Publications

During the development of the present thesis, the following papers were published in conference proceedings:

1. SOUSA, L. C.; AYALA, H. V. H. **Slip Estimation with Receding-horizon Strategy for Off-road Vehicles with Nonlinear Tire Interactions**. In: 28th Mediterranean Conference on Control and Automation, 2020.

2. SOUSA, L. C.; AYALA, H. V. H. **Nonlinear Model Approximation Methods for Off-road Vehicle Path Tracking with MPC**. In: XXIII Congresso Brasileiro de Automática, 2020.

PUC-Rio - Certificação Digital Nº 1912774/CA

3. SOUSA, L. C.; AYALA, H. V. H. **Nonlinear Tire Model Approximation Using Artificial Neural Networks**. In: XV Simpósio Brasileiro de Automação Inteligente, 2021.

4. LAGO, A. W. C.; SOUSA, L. C.; LOPES, F. R.; SOUSA, D. H. B.; AYALA, H. V. H., MEGGIOLARO, M. A. (2022). **Identificação usando método não linear de um sistema de posicionamento**. In: XXIV Congresso Brasileiro de Automática, 2022.

The following paper was published in the journal IEEE Access:

1. SOUSA, L. C.; AYALA, H. V. H. **Nonlinear Tire Model Approximation Using Machine Learning for Efficient Model Predictive Control**. IEEE Access, 10:107549-107562, 2022.

Moreover, the following two papers are in the final stages of production for future journal submission:

1. SOUSA, L. C.; AYALA, H. V. H. **Lateral model identification using multi-ARX models for efficient model predictive control**. Future submission to IEEE Transactions on Vehicular Technology.

2. SOUSA, L. C.; AYALA, H. V. H. **Hybrid physics-based and black-box data-driven nonlinear identification of vehicle dynamics**. Future submission to IEEE Transactions on Control Systems Technology.

PUC-Rio - Certificação Digital Nº 1912774/CA

9.2 Future works

The results of this thesis highlight several points for future improvement of this work to build more accurate and computationally efficient models. There is a need to build models to improve vehicle modeling and control. Thus, SYSID and ML techniques become important to capture nonlinear aspects present in tire efforts, public roads, and race conditions. In future works, the following topics will be pursued concerning the procedures for predictive model control and system identification suggested in this work:

- To consider more vehicle states and restrictions to ensure comfort to the passengers and the stability of the vehicle, e.g., the slip angle and yaw rate [31, 73];
- To implement different cost function architectures to enhance path-tracking control as the switched cost functions, in which a switching criterion is used to shift between different vehicle models [18, 28].

– To apply online estimation by means of recursive algorithms [147] and Neural Network-based learned dynamic model [18] to increase the reliability of the vehicle identified model;

– To apply online estimation by means of recursive algorithms [147] and Neural Network-based learned dynamic model [18] to increase the reliability of the vehicle identified model;

– To consider external disturbances affecting the system and uncertainties regarding the vehicle properties [148, 149] to ensure stability in control applications;

– To test the machine learning algorithms in the testbed platform in order to verify the implementation viability, effectiveness, and robustness in real-time;

– To use smarter initialization techniques and different solvers. The trajectory optimization problem is formulated using an NLP with discretized variables and constraints. It is an easy way to implement computationally, and several numerical solvers are available to solve the optimization problem. However, the NLP procedure suffers from issues related to gradient-based approaches, such as 1) the sensibility regarding the changes in initial guess and 2) highly prone to local minima;

The following can be cited in the scope of the hybrid approach using black-box identification. A more precise vehicle model that includes nonlinear aspects, including tire aspects [18, 68], and motor/brake characteristics [37, 133], can be established. In addition, accurate vehicle models derived from black-box architectures [43, 44] can also be used as the main function to be applied to Eq. (8-2). The latter is in line with the most recent research for physically-inspired machine learning modeling [106], which will significantly impact vehicle handling modeling for autonomous vehicle applications. Another possibility concerning vehicle models derived from NARX models is establishing a similar procedure adopted in Section 7.1.3 by varying the NARX parameters to find the best case for the combination of na, nb, and ne based on error-based metrics. The procedure should be tested using different data sets from other racetracks also available in reference [105] aiming for reliability and robustness. Considering control applications, adopting hybrid frameworks as an approximation from the vehicle models to derive efficient and accurate control laws could permit real-time implementation. Recent works use ANNs with MPC [3, 18, 81]. Additional topics for future consideration include applying the proposed approach with model predictive control considering the driver's commands input [150].

PUC-Rio - Certificação Digital Nº 1912774/CA

Bibliography

[1] MICHELI, F.; BERSANI, M.; ARRIGONI, S.; BRAGHIN, F. ; CHELI, F.. **Nmpc trajectory planner for urban autonomous driving**. Vehicle System Dynamics, p. 1–23, 2022.

[2] GREENBLATT, J. B.; SHAHEEN, S.. **Automated vehicles, on-demand mobility, and environmental impacts**. Current sustainable/renewable energy reports, 2(3):74–81, 2015.

[3] SPIELBERG, N. A.; BROWN, M.; KAPANIA, N. R.; KEGELMAN, J. C. ; GERDES, J. C.. **Neural network vehicle models for high-performance automated driving**. Science Robotics, 4(28), 2019.

[4] YAO, Q.; TIAN, Y.; WANG, Q. ; WANG, S.. **Control strategies on path tracking for autonomous vehicle: State of the art and future challenges**. IEEE Access, 8:161211–161222, 2020.

[5] FORTUNE BUSINESS INSIGHTS. **Autonomous cars market size, share & covid-19 impact analysis, by type (fully autonomous and semi-autonomous), by vehicle type (passenger cars and commercial vehicles), and regional forecasts, 2021-2028**. Technical Report FBI100141, Fortune Business Insights, 2021.

[6] SAE ON-ROAD AUTOMATED VEHICLES STANDARDS COMMITTEE AND OTHERS. **Sae j3016: Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems (revised)**, 2021.

[7] AMER, N. H.; ZAMZURI, H.; HUDHA, K. ; KADIR, Z. A.. **Modelling and control strategies in path tracking control for autonomous ground vehicles: A review of state of the art and challenges**. Journal of Intelligent and Robotic Systems, 86(2):225–254, 2017.

[8] PICCININI, M.; LARCHER, M.; PAGOT, E.; PISCINI, D.; PASQUATO, L. ; BIRAL, F.. **A predictive neural hierarchical framework for on-line time-optimal motion planning and control of black-box vehicle models**. Vehicle System Dynamics, p. 1–28, 2022.

PUC-Rio - Certificação Digital Nº 1912774/CA

[9] ZHANG, Z.; XIE, L.; LU, S.; WU, X. ; SU, H.. **Vehicle yaw stability control with a two-layered learning mpc**. Vehicle System Dynamics, p. 1–22, 2022.

[10] WISCHNEWSKI, A.; EULER, M.; GÜMÜS, S. ; LOHMANN, B.. **Tube model predictive control for an autonomous race car**. Vehicle System Dynamics, 60(9):3151–3173, 2022.

[11] RANKIN, A. L.; CRANE III, C. D.; ARMSTRONG II, D.; NEASE, A. D. ; BROWN, H. E.. **Autonomous path-planning navigation system for site characterization**. In: PROCEEDINGS OF THE SPIE, p. 176–186, 1996.

[12] WIT, J. S.. **Vector pursuit path tracking for autonomous ground vehicles**. PhD thesis, University of Florida, Florida, 2000.

[13] SNIDER, J. M.. **Automatic steering methods for autonomous automobile path tracking**. Technical Report CMU-RI-TR-09-08, 78p., Carnegie Mellon University, 2009.

[14] RAFFO, G. V.; GOMES, G. K.; NORMEY-RICO, J. E.; KELBER, C. R. ; BECKER, L. B.. **A predictive controller for autonomous vehicle path tracking**. IEEE Transactions on Intelligent Transportation Systems, 10:92–102, 2009.

[15] COULTER, R. C. **Implementation of the pure pursuit path tracking algorithm**. Technical Report CMU-RI-TR-92-01, 15p., Robotics Institute, Carnegie Mellon University, 1992.

[16] SHAN, Y.; YANG W.; CHEN, C.; ZHOU, J.; ZHENG, L. ; LI, B.. **Cf-pursuit: A pursuit method with a clothoid fitting and a fuzzy controller for autonomous vehicles**. International Journal of Advanced Robotic Systems, 12:225–254, 2015.

[17] TÖRO, O.; BÉCSI, T. ; ARADI, S.. **Design of lane keeping algorithm of autonomous vehicle**. Period. Polytech. Transp. Eng., 44:60–68, 2016.

[18] ROKONUZZAMAN, M.; MOHAJER, N.; NAHAVANDI, S. ; MOHAMED, S.. **Model predictive control with learned vehicle dynamics for autonomous vehicle path tracking**. IEEE Access, 9:128233–128249, 2021.

[19] RILL, G.; ARRIETA CASTRO, A.. **Road Vehicle Dynamics: Fundamentals and Modeling with MATLAB®**. CRC Press, 2 edition, 2020.

PUC-Rio - Certificação Digital Nº 1912774/CA

[20] GILLESPIE, T. D.. **Fundamentals of vehicle dynamics**. Society of Automotive Engineers, Warrendale, 1 edition, 1992.

[21] WONG, J. Y.. **Theory of ground vehicles**. John Wiley & Sons, Hoboken, 4 edition, 2008.

[22] HE, R.; SANDU, C. ; OSORIO, J. E.. **Systematic tests for study of tire tractive performance on soft soil: Part i – experimental data collection**. Journal of Terramechanics, 85:59–76, 2019.

[23] SABIHA, A. D.; KAMEL, M. A.; SAID, E. ; HUSSEIN, W. M.. **Ros-based trajectory tracking control for autonomous tracked vehicle using optimized backstepping and sliding mode control**. Robotics and Autonomous Systems, 152:104058, 2022.

[24] ZHANG, C.; HU, J.; QIU, J.; YANG, W.; SUN, H. ; CHEN, Q.. **A novel fuzzy observer-based steering control approach for path tracking in autonomous vehicles**. IEEE Transactions on Fuzzy Systems, 27(2):278–290, 2019.

[25] ZHOU, X.; WANG, Z. ; WANG, J.. **Popov-h∞ robust path-tracking control of autonomous ground vehicles with consideration of sector-bounded kinematic nonlinearity**. Journal of Dynamic Systems, Measurement, and Control, 143(11), 2021.

[26] CHOI, Y.-M.; PARK, J.-H.. **Game-based lateral and longitudinal coupling control for autonomous vehicle trajectory tracking**. IEEE Access, 10:31723–31731, 2022.

[27] AN, Q.; CHENG, S.; LI, C.; LI, L. ; PENG, H.. **Game theory-based control strategy for trajectory following of four-wheel independently actuated autonomous vehicles**. IEEE Transactions on Vehicular Technology, 70(3):2196–2208, 2021.

[28] ROKONUZZAMAN, M.; MOHAJER, N. ; NAHAVANDI, S.. **Effective adoption of vehicle models for autonomous vehicle path tracking: a switched mpc approach**. Vehicle System Dynamics, p. 1–24, 2022.

[29] DE BERNARDIS, M.; RINI, G.; BOTTIGLIONE, F.; HARTAVI, A. E. ; SORNIOTTI, A.. **On nonlinear model predictive direct yaw moment control for trailer sway mitigation**. Vehicle System Dynamics, p. 1–27, 2022.

PUC-Rio - Certificação Digital Nº 1912774/CA

[30] RAWLINGS, J. B.; MAYNE, D. Q. ; DIEHL, M.. **Model predictive control : theory, computation, and design**. Nob Hill, 2 edition, 2017.

[31] TAGHAVIFAR, H.. **Neural network autoregressive with exogenous input assisted multi-constraint nonlinear predictive control of autonomous vehicles**. IEEE Transactions on Vehicular Technology, 68(7):6293–6304, 2019.

[32] MOHAJER, N.; NAHAVANDI, S.; ABDI, H. ; NAJDOVSKI, Z.. **Enhancing passenger comfort in autonomous vehicles through vehicle handling analysis and optimization**. IEEE Intelligent Transportation Systems Magazine, 13(3):156–173, 2021.

[33] ROKONUZZAMAN, M.; MOHAJER, N.; NAHAVANDI, S. ; MOHAMED, S.. **Review and performance evaluation of path tracking controllers of autonomous vehicles**. IET Intelligent Transport Systems, 15(5):646–670, 2021.

[34] LIAN, Y. F.; ZHAO, Y.; HU, L. L. ; TIAN, Y. T.. **Cornering stiffness andsideslip angle estimation based on simplified lateral dynamic models for four-in-wheel-motor-driven electric vehicles with lateral tire force information**. Int. J. Automot. Technol., 16:669–683, 2020.

[35] JAMES, S.; ANDERSON, S. R.. **Linear system identification of longitudinal vehicle dynamics versus nonlinear physical modelling**. In: 2018 UKACC 12TH INTERNATIONAL CONFERENCE ON CONTROL (CONTROL), p. 146–151. IEEE, 2018.

[36] PYTKA, J.. **Lateral dynamics of a SUV on deformable surfaces by system identification. part II. models reconstruction**. IOP Conf. Ser.: Mater. Sci. Eng., 421, 2018.

[37] VICENTE, B. A. H.; JAMES, S. S. ; ANDERSON, S. R.. **Linear system identification versus physical modeling of lateral–longitudinal vehicle dynamics**. IEEE Transactions on Control Systems Technology, 29(3):1380–1387, 2021.

[38] LJUNG, L.; HJALMARSSON, H.; OHLSSON, H.. **Four encounters with system identification**. European Journal of Control, 17:449–471, 2011.

[39] PAPPALARDO, C. M.; GUIDA, D.. **A time-domain system identification numerical procedure for obtaining linear dynamical models**

PUC-Rio - Certificação Digital Nº 1912774/CA

of multibody mechanical systems. Arch. Appl. Mech, 88:1325–1347, 2018.

[40] KHAN, M. E.; KHAN, F. **A comparative study of white box, black box and grey box testing techniques**. International Journal of Advanced Computer Science and Applications, 3, 2012.

[41] AYALA, H. V. H.. **Computational intelligence methods applied to nonlinear black-box system identification**. PhD thesis, Pontifical Catholic University of Paraná, Curitiba, 2016.

[42] BABUŠKA, R.; VERBRUGGEN, H.. **Neuro-fuzzy methods for nonlinear system identification**. Annual Reviews in Control, 27:73–85, 2003.

[43] JAMES, S. S.; ANDERSON, S. R. ; DA LIO, M.. **Longitudinal vehicle dynamics: A comparison of physical and data-driven models under large-scale real-world driving conditions**. IEEE Access, 8:73714–73729, 2020.

[44] DA LIO, M.; BORTOLUZZI, D. ; ROSATI PAPINI, G. P.. **Modelling longitudinal vehicle dynamics with neural networks**. Vehicle System Dynamics, 58(11):1675–1693, 2020.

[45] DIAS, J. E. A.; PEREIRA, G. A. S. ; PALHARES, R. M.. **Longitudinal model identification and velocity control of an autonomous car**. IEEE Transactions on Intelligent Transportation Systems, 16(2):776–786, 2015.

[46] GASPAR, P.; SZABO, Z. ; BOKOR, J.. **A grey-box identification of an lpv vehicle model for observer-based side slip angle estimation**. In: 2007 AMERICAN CONTROL CONFERENCE, p. 2961–2966, 2007.

[47] HERRERA, D.; TOSETTI, S. ; CARELLI, R.. **Dynamic modeling and identification of an agriculture autonomous vehicle**. IEEE Latin America Transactions, 14:2631–2637, 2016.

[48] NASH, C. J.; COLE, D. J.. **Identification and validation of a driver steering control model incorporating human sensory dynamics**. Vehicle System Dynamics, 58(4):495–517, 2020.

[49] PACEJKA, H.. **Tire and Vehicle Dynamics**. Butterworth-Heinemann, Oxford, 3 edition, 2012.

PUC-Rio - Certificação Digital Nº 1912774/CA

[50] CABRERA, J.; CASTILLO, J.; PÉREZ, J.; VELASCO, J.; GUERRA, A. ; HERNÁNDEZ, P.. **A procedure for determining tire-road friction characteristics using a modification of the magic formula based on experimental results**. Sensors, 18, 2018.

[51] LÓPEZ, A.; VÉLEZ, P. ; MORIANO, C.. **Approximations to the magic formula**. Int.J Automot. Technol, 11:155–166, 2010.

[52] WANG, J.; LIU, Y.; DING, L.; LI, J.; GAO, H.; LIANG, Y. ; SUN, T.. **Neural network identification of a racing car tire model**. Journal of Engineering, 2018, 2018.

[53] SATZGER, C.; DE CASTRO, R. ; BÜNTE, T.. **A model predictive control allocation approach to hybrid braking of electric vehicles**. In: 2014 PROC. INTELL. VEH. SYMP., p. 286–292, 2014.

[54] LONG, C.; CHEN, H.. **Comparative study between the magic formula and the neural network tire model based on genetic algorithm**. In: 2010 THIRD INTERNATIONAL SYMPOSIUM ON INTELLIGENT INFORMATION TECHNOLOGY AND SECURITY INFORMATICS, p. 280–284, 2010.

[55] OLAZAGOITIA, J.; PEREZ, J. ; BADEA, F.. **Identification of tire model parameters with artificial neural networks**. Applied Sciences, 10, 2020.

[56] DAO, T.; CHEN, C.. **Path tracking control of a motorcycle based on system identification**. IEEE Transactions on Vehicular Technology, 60:2927–2935, 2011.

[57] NØRGAARD, M.; RVN, O.; POULSE, N. ; HANSEN, L.. **Neural networks for modelling and control of dynamic systems**. Springer-Verlag London Limited, 2001.

[58] KAHEMAN, K.; KAISER, E.; STROM, B.; KUTZ, J. N. ; BRUNTON, S. L.. **Learning discrepancy models from experimental data**. arXiv preprint arXiv:1909.08574, 2019.

[59] DE SILVA, B. M.; HIGDON, D. M.; BRUNTON, S. L. ; KUTZ, J. N.. **Discovery of physics from data: universal laws and discrepancies**. Frontiers in artificial intelligence, 3:25, 2020.

[60] FALCONE, P.; BORRELLI, F.; ASGARI, J.; TSENG, H. E. ; HROVAT, D.. **Predictive active steering control for autonomous vehicle**

PUC-Rio - Certificação Digital Nº 1912774/CA

systems. IEEE Transactions on control systems technology, 15(3):566–580, 2007.

[61] WU, D.-M.; LI, Y.; DU, C.-Q.; DING, H.-T.; LI, Y.; YANG, X.-B. ; LU, X.-Y.. **Fast velocity trajectory planning and control algorithm of intelligent 4wd electric vehicle for energy saving using time-based mpc**. IET Intelligent Transport Systems, 13(1):153–159, 2019.

[62] KUHNE, F.; LAGES, W. F. ; DA SILVA, J. G.. **Point stabilization of mobile robots with nonlinear model predictive control**. In: IEEE INTERNATIONAL CONFERENCE MECHATRONICS AND AUTOMATION, 2005, volumen 3, p. 1163–1168, 2005.

[63] BATKOVIC, I.; ZANON, M.; ALI, M. ; FALCONE, P.. **Real-time constrained trajectory planning and vehicle control for proactive autonomous driving with road users**. In: 2019 18TH EUROPEAN CONTROL CONFERENCE (ECC), p. 256–262, 2019.

[64] ROKONUZZAMAN, M.; MOHAJER, N. ; NAHAVANDI, S.. **Nmpc-based controller for autonomous vehicles considering handling performance**. In: 2019 7TH INTERNATIONAL CONFERENCE ON CONTROL, MECHATRONICS AND AUTOMATION (ICCMA), p. 266–270, 2019.

[65] YOON, Y.; SHIN, J.; KIM, H. J.; PARK, Y. ; SASTRY, S.. **Model-predictive active steering and obstacle avoidance for autonomous ground vehicles**. Control Engineering Practice, 17(7):741–750, 2009.

[66] YAKUB, F.; MORI, Y.. **Comparative study of autonomous path-following vehicle control via model predictive control and linear quadratic control**. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of automobile engineering, 229(12):1695–1714, 2015.

[67] GUO, J.; LUO, Y.; LI, K. ; DAI, Y.. **Coordinated path-following and direct yaw-moment control of autonomous electric vehicles with sideslip angle estimation**. Mechanical Systems and Signal Processing, 105:183–199, 2018.

[68] CAO, J.; SONG, C.; PENG, S.; SONG, S.; ZHANG, X. ; XIAO, F.. **Trajectory tracking control algorithm for autonomous vehicle**

PUC-Rio - Certificação Digital Nº 1912774/CA

considering cornering characteristics. IEEE Access, 8:59470–59484, 2020.

[69] WU, Y.; WANG, L.; ZHANG, J. ; LI, F.. **Path following control of autonomous ground vehicle based on nonsingular terminal sliding mode and active disturbance rejection control**. IEEE Transactions on Vehicular Technology, 68:6379–6390, 2019.

[70] BESSELMANN, T.; MORARI, M.. **Hybrid parameter-varying model predictive control for autonomous vehicle steering**. European Journal of Control, 14(5):418–431, 2008.

[71] CUI, Q.; DING, R.; WEI, C. ; ZHOU, B.. **Path-tracking and lateral stabilisation for autonomous vehicles by using the steering angle envelope**. Vehicle System Dynamics, 59(11):1672–1696, 2021.

[72] KIM, M.; LEE, D.; AHN, J.; KIM, M. ; PARK, J.. **Model predictive control method for autonomous vehicles using time-varying and non-uniformly spaced horizon**. IEEE Access, 9:86475–86487, 2021.

[73] GUO, N.; ZHANG, X.; ZOU, Y.; LENZO, B. ; ZHANG, T.. **A computationally efficient path-following control strategy of autonomous electric vehicles with yaw motion stabilization**. IEEE Transactions on Transportation Electrification, 6(2):728–739, 2020.

[74] LENAIN, R.; THUILOT, B. ; CARIOU, C.;AND MARTINET, P.. **Model predictive control for vehicle guidance in presence of sliding application to farm vehicles path tracking**. In: PROCEEDINGS OF THE 2005 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, p. 885–890, Barcelona, 2005.

[75] LENAIN, R.;AND THUILOT, B.; CARIOU, C. ; MARTINET, P.. **Adaptive and predictive path tracking control for off-road mobile robots**. European Journal of Control, 13:419–439, 2007.

[76] SHEN, C.; GUO, H.; LIU, F. ; CHEN, H.. **Mpc-based path tracking controller design for autonomous ground vehicles**. In: 2017 36TH CHINESE CONTROL CONFERENCE (CCC), p. 9584–9589, 2017.

[77] LI, S.;AND LI, Z.; ZHANG, B.; ZHENG, S.; LU, X. ; YU, Z.. **Path tracking for autonomous vehicles based on nonlinear model: Predictive control**. SAE Technical Paper, 2019.

PUC-Rio - Certificação Digital Nº 1912774/CA

[78] PENG, H.; WANG, W.; AN, Q.; XIANG, C. ; LI, L.. **Path tracking and direct yaw moment coordinated control based on robust mpc with the finite time horizon for autonomous independent-drive vehicles**. IEEE Transactions on Vehicular Technology, 69(6):6053–6066, 2020.

[79] SONG, X.; Y. SHAO, Y. ; QU, Z.. **A vehicle trajectory tracking method with a time-varying model based on the model predictive control**. IEEE Access, 8:16573–16583, 2020.

[80] TANG, L.; YAN, F.; ZOU, B.; WANG, K. ; LV, C.. **An improved kinematic model predictive control for high-speed path tracking of autonomous vehicles**. IEEE Access, 8:51400–51413, 2020.

[81] SPIELBERG, N. A.; BROWN, M. ; GERDES, J. C.. **Neural network model predictive motion control applied to automated driving with unknown friction**. IEEE Transactions on Control Systems Technology, 2021.

[82] FÉNYES, D.; NÉMETH, B. ; GÁSPÁR, P.. **A novel data-driven modeling and control design method for autonomous vehicles**. Energies, 14(2):517, 2021.

[83] JAZAR, R. N.. **Vehicle dynamics: Theory and Application**. Springer, New York, 3 edition, 2017.

[84] WIT, J.; CRANE III, C. D.; ARMSTRONG, D.. **Autonomous ground vehicle path tracking**. J. Robotic Syst., 21:439–449, 2004.

[85] HOFFMANN, G. M.; TOMLIN, C. J.; MONTEMERLO, M. ; THRUN, S.. **Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing**. In: 2007 AMERICAN CONTROL CONFERENCE, p. 2296–2301, New York City, 2007.

[86] LUCET, E.; LENAIN, R. ; GRAND, C.. **Dynamic path tracking control of a vehicle on slippery terrain**. Control Engineering Practice, 42:60–73, 2015.

[87] JOHANASTROM, K.; CANUDAS-DE-WIT, C.. **Revisiting the lugre friction model**. IEEE Control Systems Magazine, 28:101–114, 2008.

[88] BERNTORP, K.; QUIRYNEN, R.; UNO, T. ; CAIRANO, S. D.. **Trajectory tracking for autonomous vehicles on varying road surfaces by**

PUC-Rio - Certificação Digital Nº 1912774/CA

friction-adaptive nonlinear model predictive control. Vehicle System Dynamics, 58(5):705–725, 2020.

[89] LI, S. E.; CHEN, H.; LI, R.; LIU, Z.; WANG, Z. ; XIN, Z.. **Predictive lateral control to stabilise highly automated vehicles at tire-road friction limits.** Vehicle System Dynamics, 58(5):768–786, 2020.

[90] ARRIETA CASTRO, A.. **Development of a robust and fault tolerant integrated control system to improve the stability of road vehicles in critical driving scenarios.** PhD thesis, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, 2018.

[91] XU, T.; WANG, X.. **Roll stability and path tracking control strategy considering driver in the loop.** IEEE Access, 9:46210–46222, 2021.

[92] ATAEI, M.; KHAJEPOUR, A. ; JEON, S.. **Model predictive control for integrated lateral stability, traction/braking control, and rollover prevention of electric vehicles.** Vehicle system dynamics, 58(1):49–73, 2020.

[93] GUO, L.; GE, P.; YUE, M. ; LI, J.. **Trajectory tracking algorithm in a hierarchical strategy for electric vehicle driven by four independent in-wheel motors.** Journal of the Chinese Institute of Engineers, p. 1–12, 2020.

[94] LJUNG, L.. **Perspectives on system identification.** Annual Reviews in Control, 34:1–12, 2010.

[95] NASH, C.; COLE, D.. **Identification of a driver model incorporating sensory dynamics, with nonlinear vehicle dynamics and transient disturbances.** Vehicle System Dynamics, 60(8):2805–2824, 2022.

[96] BASCETTA, L.; FERRETTI, G.. **Lft-based identification of lateral vehicle dynamics.** IEEE Transactions on Vehicular Technology, 71(2):1349–1362, 2022.

[97] PAN, Y.; NIE, X.; LI, Z. ; GU, S.. **Data-driven vehicle modeling of longitudinal dynamics based on a multibody model and deep neural networks.** Measurement, 180:109541, 2021.

[98] KAPANIA, N. R.; GERDES, J. C.. **Learning at the racetrack: Data-driven methods to improve racing performance over multiple laps.** IEEE Transactions on Vehicular Technology, 69(8):8232–8242, 2020.

PUC-Rio - Certificação Digital Nº 1912774/CA

[99] GARDNER, P.; ROGERS, T.; LORD, C. ; BARTHORPE, R.. **Learning model discrepancy: A gaussian process and sampling-based approach**. Mechanical Systems and Signal Processing, 152:107381, 2021.

[100] GARDNER, P.; LORD, C. ; BARTHORPE, R. J.. **Bayesian history matching for structural dynamics applications**. Mechanical Systems and Signal Processing, 143:106828, 2020.

[101] ARENDT, P. D.; APLEY, D. W. ; CHEN, W.. **A preposterior analysis to predict identifiability in the experimental calibration of computer models**. IIE Transactions, 48(1):75–88, 2016.

[102] PIRES, I.; AYALA, H. V. H. ; WEBER, H. I.. **Nonlinear ensemble gray and black-box system identification of friction induced vibrations in slender rotating structures**. Mechanical Systems and Signal Processing, 186:109815, 2023.

[103] RIBEIRO, A. M.; MOUTINHO, A.; FIORAVANTI, A. R. ; DE PAIVA, E. C.. **Estimation of tire–road friction for road vehicles: a time delay neural network approach**. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 42(1):1–12, 2020.

[104] XIAO, Y.; ZHANG, X.; XU, X.; LIU, X. ; LIU, J.. **Deep neural networks with koopman operators for modeling and control of autonomous vehicles**. IEEE Transactions on Intelligent Vehicles, p. 1–12, 2022.

[105] KEGELMAN, J. C.; HARBOTT, L. K. ; GERDES, J. C.. **Insights into vehicle trajectories at the handling limits: analysing open data from race car drivers**. Vehicle system dynamics, 55(2):191–207, 2017.

[106] KARNIADAKIS, G. E.; LU, I. G. K. L.; PERDIKARIS, P.; WANG, S. ; YANG, L.. **Physics-informed machine learning**. Nature Reviews Physics, 3:422–440, 2021.

[107] WANG, H.; LIU, B.; PING, X. ; AN, Q.. **Path tracking control for autonomous vehicles based on an improved mpc**. IEEE Access, p. 161064–161073, 2019.

[108] WEI, L.; WANG, X.; LI, L.; FAN, Z.; DOU, R. ; LIN, J.. **T-s fuzzy model predictive control for vehicle yaw stability in nonlinear region**. IEEE Transactions on Vehicular Technology, 70(8):7536–7546, 2021.

PUC-Rio - Certificação Digital Nº 1912774/CA

[109] ZHAO, Y.; PI, W.; ZHANG, W.; WANG, Q.; FENG, S.; DENG, H. ; LIN, F.. **A vehicle handling inverse dynamics method for emergency avoidance path tracking based on adaptive inverse control**. IEEE Transactions on Vehicular Technology, 70(6):5470–5482, 2021.

[110] SAVARESI, S.; TANELLI, M.. **Active Braking Control Systems Design for Vehicles**. Springer, London, 1 edition, 2010.

[111] HIRSCHBERG, W.; RILL, G. ; WEINFURTER, H.. **Tire model tmeasy**. Vehicle System Dynamics, 45(sup1):101–119, 2007.

[112] AGUIRRE, L. A.. **Introdução à identificação de sistemas**. Editora UFMG, 2007.

[113] HABER, R.; UNBEHAUEN, H.. **Structure identification of nonlinear dynamic systems—a survey on input/output approaches**. Automatica, 26(4):651–677, 1990.

[114] LENNART, L.. **System identification: theory for the user**. Prentice Hall, 2 edition, 1999.

[115] DA COSTA, F. P.; DOMINGUES, P. H. L. S. P.; FREIRE, R. Z.; COELHO, L. S.; TAVAKOLPOUR-SALEH, A. ; AYALA, H. V. H.. **Genetic algorithm for topology optimization of an artificial neural network applied to aircraft turbojet engine identification**. In: 2019 IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (CEC), p. 1095–1101, 2019.

[116] MAREN, A. J.; HARSTON, C. T. ; PAP, R. M.. **Handbook of neural computing applications**. Academic Press, 2014.

[117] HAYKIN, S. S.. **Neural networks and learning machines**. Prentice Hall, 3 edition, 2009.

[118] MCCULLOCH, W. S.; PITTS, W.. **A logical calculus of the ideas immanent in nervous activity**. Bulletin of mathematical biophysics, 5(4):115–133, 1943.

[119] ROSENBLATT, F.. **The perceptron: a probabilistic model for information storage and organization in the brain**. Psychological review, 65(6):386, 1958.

[120] WERBOS, P.. **Beyond regression: new tools for prediction and analysis in the behavioral sciences**. PhD thesis, Harvard University, Cambridge, 1974.

PUC-Rio - Certificação Digital Nº 1912774/CA

[121] BILLINGS, S. A.. **Nonlinear system identification: NARMAX methods in the time, frequency, and spatio temporal domains**. John Wiley & Sons, 2013.

[122] HAYKIN, S. S.. **Adaptive filter theory**. Pearson Education, 2002.

[123] BISHOP, C. M.. **Pattern recognition and machine learning**. Springer, 1 edition, 2006.

[124] WÄCHTER, A.; BIEGLER, L.. **On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming**. Mathematical Programming, 106:25–57, 2006.

[125] ANDERSSON, J. A.; GILLIS, J.; HORN, G.; RAWLINGS, J. B. ; DIEHL, M.. **Casadi: a software framework for nonlinear optimization and optimal control**. Mathematical Programming, 11:1–36, 2006.

[126] BOCK, H. G.; PLITT, K.-J.. **A multiple shooting algorithm for direct solution of optimal control problems**. IFAC Proceedings Volumes, 17(2):1603–1608, 1984.

[127] HINDMARSH, A.; BROWN, P.; GRANT, K.; LEE, S.; SERBAN, R.; SHUMAKER, D. ; WOODWARD, C.. **SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers**. ACM Transactions on Mathematical Software, 31(3):363–396, 2005.

[128] KIM, G.; LEE, S. Y.; OH, J.-S. ; LEE, S.. **Deep learning-based estimation of the unknown road profile and state variables for the vehicle suspension system**. IEEE Access, 9:13878–13890, 2021.

[129] BAKKER, E.; NYBORG, L. ; PACEJKA, H. B.. **Tyre modelling for use in vehicle dynamics studies**. SAE Transactions, p. 190–204, 1987.

[130] MATHWORKS STUDENT COMPETITIONS TEAM. **Analyzing tire test data**. Available at `https://www.mathworks.com/matlabcentral/fileexchange/67987-analyzing-tire-test-data` (2021/09/24).

[131] KASPRZAK, E.; GENTZ, D.. **The formula sae tire test consortium-tire testing and data handling**. SAE Technical Paper, 1(2006-01-3606), 2006.

[132] VERSCHUEREN, R.; FRISON, G.; KOUZOUPIS, D.; VAN DUIJKEREN, N.; ZANELLI, A.; NOVOSELNIK, B.; ALBIN, T.; QUIRYNEN, R. ; DIEHL, M.. **acados—a modular open-source framework for fast embedded optimal control**. Math. Prog. Comp., 2021.

PUC-Rio - Certificação Digital Nº 1912774/CA

[133] WU, H.; SI, Z. ; LI, Z.. **Trajectory tracking control for four-wheel independent drive intelligent vehicle based on model predictive control**. IEEE Access, 8:73071–73081, 2020.

[134] CHEN, Y.; CHEN, S.; REN, H.; GAO, Z. ; LIU, Z.. **Path tracking and handling stability control strategy with collision avoidance for the autonomous vehicle under extreme conditions**. IEEE transactions on vehicular technology, 69(12):14602–14617, 2020.

[135] SHAJU, A.; PANDEY, A. K.. **Modelling transient response using pac 2002-based tyre model**. Vehicle System Dynamics, p. 1–27, 2020.

[136] MARQUES, F.; WOLIŃSKI, L.; WOJTYRA, M.; FLORES, P. ; LANKARANI, H. M.. **An investigation of a novel lugre-based friction force model**. Mechanism and Machine Theory, 166, 2021.

[137] ZHANG, Z.; ZHENG, L.; WU, H.; ZHANG, Z.; LI, Y. ; LIANG, Y.. **An estimation scheme of road friction coefficient based on novel tyre model and improved sckf**. Vehicle system dynamics, 60(8):2775–2804, 2022.

[138] PYTKA, J. A.; TARKOWSKI, P.; FIJAŁKOWSKI, S.; BUDZYŃSKI, P.; DĄBROWSKI, J.; KUPICZ, W. ; PYTKA, P.. **An instrumented vehicle for offroad dynamics testing**. Journal of Terramechanics, 48(5):384–395, 2011.

[139] LIU, N.; ALLEYNE, A. G.. **Iterative learning identification applied to automated off-highway vehicle**. IEEE Transactions on Control Systems Technology, 22(1):331–337, 2014.

[140] RIBEIRO, A.; KOYAMA, M.; MOUTINHO, A.; DE PAIVA, E. ; FIORAVANTI, A.. **A comprehensive experimental validation of a scaled car-like vehicle: Lateral dynamics identification, stability analysis, and control application**. Control Engineering Practice, 116:104924, 2021.

[141] MOHAJER, N.; ROKONUZZAMAN, M.; NAHAVANDI, D.; SALAKEN, S. M.; NAJDOVSKI, Z. ; NAHAVANDI, S.. **Effects of road path profiles on autonomous vehicles' handling behaviour**. In: 2020 IEEE INTERNATIONAL SYSTEMS CONFERENCE (SYSCON), p. 1–6, 2020.

[142] TIAN, Y.; YAO, Q.; HANG, P. ; WANG, S.. **A gain-scheduled robust controller for autonomous vehicles path tracking based on lpv**

PUC-Rio - Certificação Digital Nº 1912774/CA

system with mpc and h∞. IEEE Transactions on Vehicular Technology, 2022.

[143] ZHANG, W.; WANG, Z.; DRUGGE, L. ; NYBACKA, M.. **Evaluating model predictive path following and yaw stability controllers for over-actuated autonomous electric vehicles**. IEEE Transactions on Vehicular Technology, 69(11):12807–12821, 2020.

[144] KEGELMAN, J. C.; HARBOTT, L. K. ; GERDES, J. C.. **2014 targa sixty-six. stanford digital repository**. `http://purl.stanford.edu/hd122pw0365`, 2016. Accessed: 2022-08-18.

[145] REN, P.; XIAO, Y.; CHANG, X.; HUANG, P.-Y.; LI, Z.; CHEN, X. ; WANG, X.. **A comprehensive survey of neural architecture search: Challenges and solutions**. ACM Computing Surveys (CSUR), 54(4):1–34, 2021.

[146] CANDON, M.; LEVINSKI, O.; OGAWA, H.; CARRESE, R. ; MARZOCCA, P.. **A nonlinear signal processing framework for rapid identification and diagnosis of structural freeplay**. Mechanical Systems and Signal Processing, 163:107999, 2022.

[147] CHEN, G.-Y.; GAN, M.; CHEN, J. ; CHEN, L.. **Embedded point iteration based recursive algorithm for online identification of nonlinear regression models**. IEEE Transactions on Automatic Control, p. 1–8, 2022.

[148] AKERMI, K.; CHOURAQUI, S. ; BOUDAA, B.. **Novel smc control design for path following of autonomous vehicles with uncertainties and mismatched disturbances**. International Journal of Dynamics and Control, 8(1):254–268, 2020.

[149] YAN, Z.; GONG, P.; ZHANG, W. ; WU, W.. **Model predictive control of autonomous underwater vehicles for trajectory tracking with external disturbances**. Ocean Engineering, 217:107884, 2020.

[150] SHI, Q.; HE, Z.; WEI, Y.; WANG, M.; ZHENG, X. ; HE, L.. **Single pedal control of battery electric vehicle by pedal torque demand with dynamic zero position**. IEEE Transactions on Intelligent Transportation Systems, 23(11):21608–21619, 2022.

[151] TAVARES, M. F.. **Utilização dos modelos ARX e ARMAX em plantas industriais ruidosas**. PhD thesis, Universidade de São Paulo, 2017.

PUC-Rio - Certificação Digital Nº 1912774/CA

[152] DE MOOR, B. L. R.. **Daisy: Database for the identification of systems.** Available at `https://homes.esat.kuleuven.be/~smc/daisy/daisydata.html` (2022/05/14) [Used dataset: Data from a flexible robot arm, Mechanical Systems, code number: 96-009.].

[153] KUMPATI, S. N.; KANNAN, P. ; OTHERS. **Identification and control of dynamical systems using neural networks.** IEEE Transactions on neural networks, 1(1):4–27, 1990.

[154] COOPER TIRES. **Indy lights downloads.** Available at `https://www.coopertire.co.uk/motorsport-tires/technical-data-resources/indy-lights/` (2022/07/20) [Used dataset: 2017 Indy Lights Road Course Force-Moment Data.].

[155] JANOT, A.; GAUTIER, M. ; BRUNOT, M.. **Data set and reference models of emps.** In: NONLINEAR SYSTEM IDENTIFICATION BENCHMARKS, p. 1–7, 2019.

[156] ÅSTRÖM, K. J.; MURRAY, R. M.. **Feedback systems: an introduction for scientists and engineers.** Princeton university press, 2 edition, 2021.

PUC-Rio - Certificação Digital Nº 1912774/CA

A
Case studies

The present appendix is devoted to introducing the case studies used in the context of system identification. As the present thesis is concerned with the overall procedure in the scope of system identification, the description of the case studies, and the exposition of the results are separated to make the text more fluid.

A.1
A Flexible robotic arm example

The system identification methodology has been acquiring great importance in the robotics, since the parameters involved in robot dynamics depend of the environment in which they are located, such as traffic on rigid or deformable soils, gravity, ambient temperature, among others [151].

PUC-Rio - Certificação Digital Nº 1912774/CA

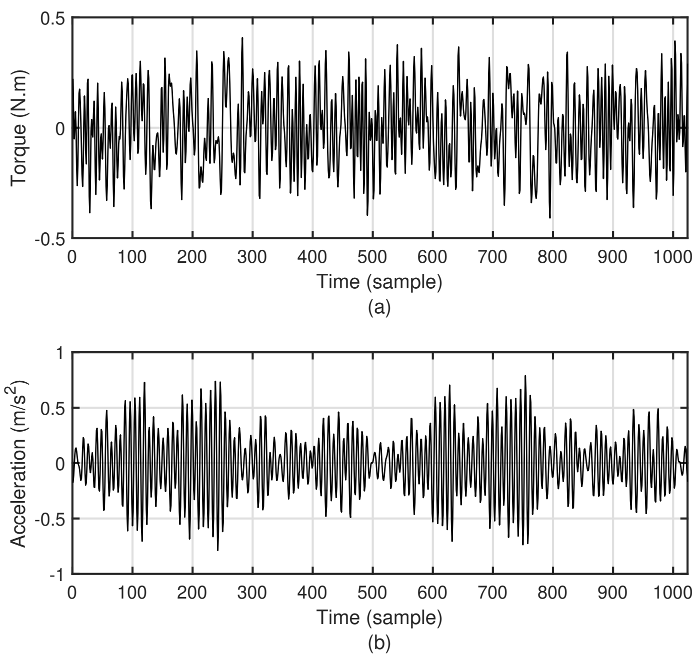

Figure A.1: Input (upper) and output (lower) for the flexible robotic arm example.

Thus, a reliable model of the various components of the dynamics of a robot must be defined. The present study case is related to a robotic arm installed in an electric motor, having the torque applied to the structure as the input of the system and the acceleration in the flexible arm as the output data (Fig. A.1). The data was obtained from the DaISy database [152] with real measurements carried out by the university of KU Leuven in Belgium.

A.2
Narendra and Pathasarathy's example

One of the first papers presented with the application of ANNs to system identification was the one proposed by [153]. The authors proposed the identification of the following system

$$y(t) = F[y(t-1), y(t-2), y(t-3), u(t-1), u(t-2)]\,. \tag{A-1}$$

with

$$F(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_2^2 + x_3^2}\,. \tag{A-2}$$

Considering the estimation phase, similar to that presented by [41], the PRBS input signal comprises 800 samples between the range [−1,1] with a length of 4 samples. In addition, each window is multiplied by uniformly distributed random values in the range [0, 1]. The estimation data is shown in Fig. A.2

PUC-Rio - Certificação Digital Nº 1912774/CA

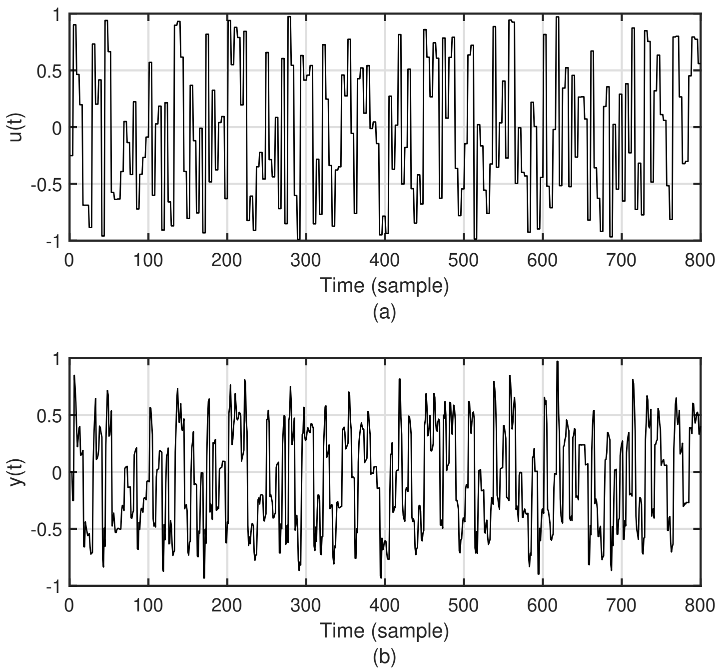

Figure A.2: Input (upper) and output (lower) for the Narendra and Parthasarathy's example (estimation phase).

Considering the validation data (Fig. A.3), the input and output signals, with 800 samples, are obtained from Eq. A-3 [153] as follows

$$u(t) = \begin{cases} \sin\left(\dfrac{2\pi t}{250}\right), & \text{if } t \leq 500 \\ 0.8 \cdot \sin\left(\dfrac{2\pi t}{250}\right) + 0.2 \cdot \sin\left(\dfrac{2\pi t}{25}\right), & \text{if } t > 500 \end{cases} \tag{A-3}$$

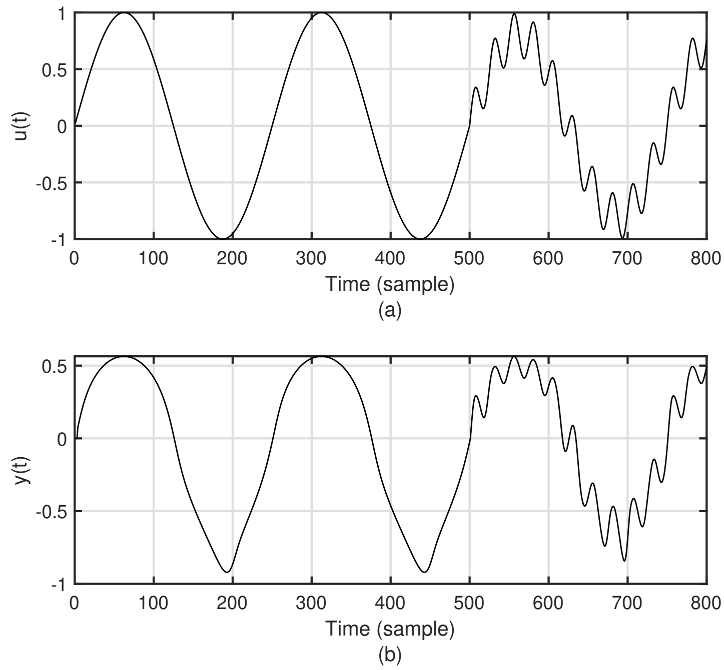

Figure A.3: Input (upper) and output (lower) for the Narendra and Parthasarathy's example (validation phase).

PUC-Rio - Certificação Digital Nº 1912774/CA

A.3 Cornering force example

The present case study comprises a dataset obtained from the Cooper Tires technical data resources, particularly the 2017 INDY Lights force-moment data [154]. Here, data regarding self-aligning torque and cornering force as a function of the slip angle are established.

The data used specifically for this case study relies on the cornering force considering that a load over the tire equals 1,471 N and a camber angle equals 0 deg. In addition, the data were dimensionless to facilitate the procedure. Fig. A.4 presents the dimensionless data.

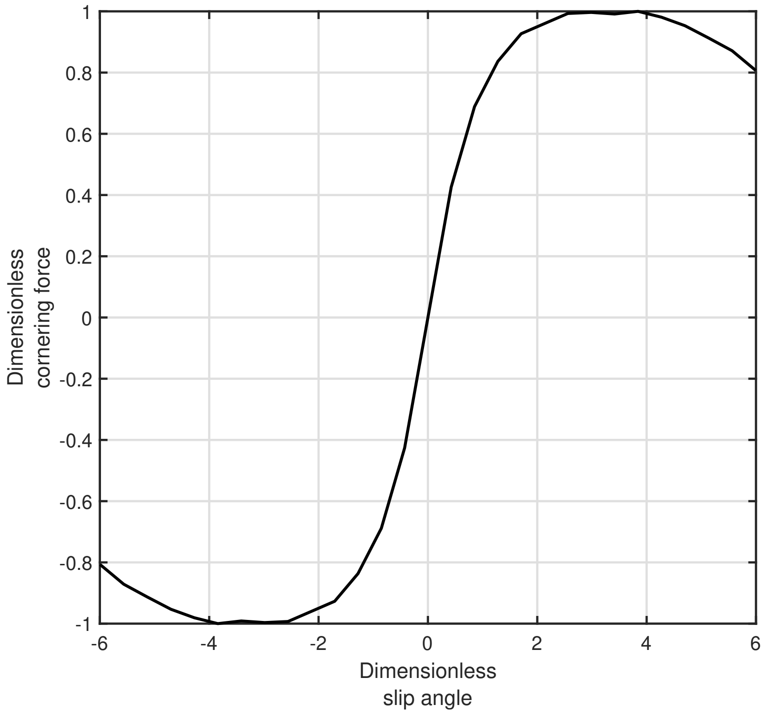

Figure A.4: Dimensionless cornering force data.

PUC-Rio - Certificação Digital Nº 1912774/CA

A.4 Electro-mechanical positioning system (EMPS) example

The Electro-Mechanical Positioning System (EMPS) is a standard drive system configuration for the prismatic joint of robots or machine tools. The present case study comprises a dataset obtained from the study developed by the reference [155].

The EMPS system is driven by a proportional-derivative (PD) controller. A dSPACE digital control system records all measurements with a sampling frequency of 1kHz for approximately 25 s. On the other hand, the joint position is recorded by another encoder with a resolution of 12,500 counts per revolution. Since the encoder works in quadrature count mode, its resolution is about 50,000 counts per revolution and ensures a high quality of the measured data.

From Newton's law, the EMPS system can be modeled as follows

$$\tau_{idm}(t) = M\ddot{q}(t) + F_f\dot{q}(t) + F_c sign(\dot{q}(t) + offset\,, \qquad \text{(A-4)}$$

where q, $\dot{q}$, and $\ddot{q}$ are the joint position, velocity and acceleration, respectively; τ_{idm} is the joint force/torque; M is the inertia of the arm; F_c and F_v are respectively the Coulomb and viscous friction; $offset$ is an offset of measurements regarding τ_{idm}. The available data are the motor's position and

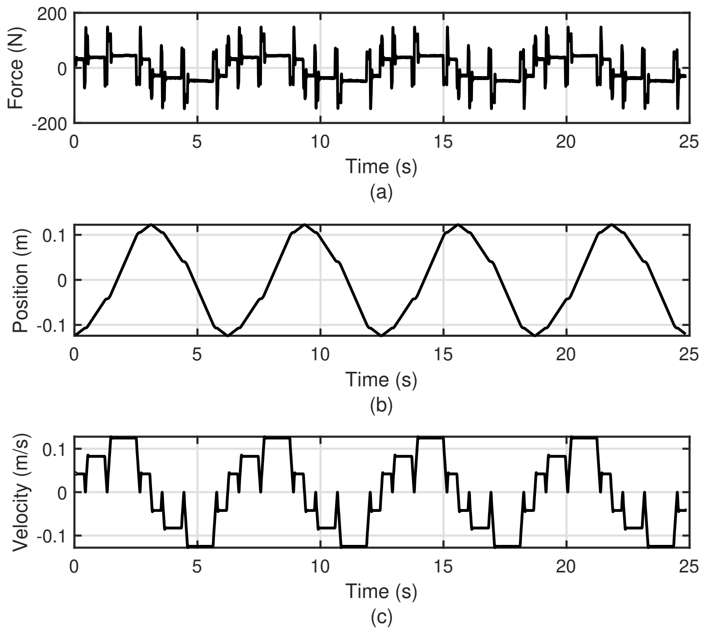

Figure A.5: Measurements acquired during EMPS experiments. (a) Force, (b) position, and (c) velocity.

PUC-Rio - Certificação Digital Nº 1912774/CA

voltage, reference position, and time. Particularly, for this case study, from the position data is possible to derive velocity and acceleration data. On the other hand, the motor voltage (ν) can be linked to τ_{idm} by the following equation [155]

$$\tau_{idm}(t) = g_{\tau}\nu\,, \tag{A-5}$$

where g_{τ} is the drive gain of the EMPS. Fig. A.5 shows the data used in this case study.

A.5 Inverted pendulum example

The present case study relies on the classic swing-up case study of an inverted pendulum. An inverted pendulum is a pendulum type that has its CG above its pivot point. It is a dynamic system considered unstable, which will fall over without support.

Fig. A.6 presents a simplified scheme for a balance system consisting of an inverted pendulum on a cart. In order to model this dynamic system, state variables are set representing the position and velocity of the base, p and $\dot{p}$, as well as the angle and angular rate of the system over the base, θ and $\dot{\theta}$. Moreover, u represents the horizontal force (F) applied at the base. Finally, the motion equations regarding the dynamic pendulum system is given by the following [156]

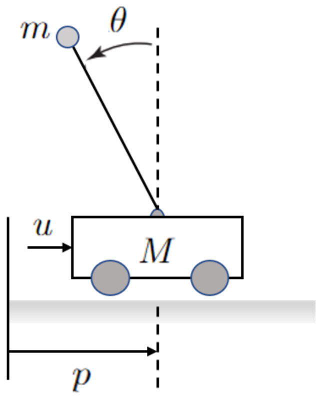

Figure A.6: Inverted pendulum example.

$$(M+m)\ddot{p} - mlcos\theta\ddot{\theta} + c\dot{p} + ml\sin\theta\dot{\theta}^2 = u\,, \tag{A-6}$$

$$-mlcos\theta\ddot{p} + (J+ml^2)\ddot{\theta} + \gamma\dot{\theta} - mgl\sin\theta = 0\,, \tag{A-7}$$

where M is the mass of the base, m and J are the mass and moment of inertia of the system to be balanced, respectively; l is the distance from the base to the CG of the system, c and γ are coefficients regarding viscous friction, and g is the acceleration due to gravity.

PUC-Rio - Certificação Digital Nº 1912774/CA